

[www.fast-sales.ru](http://www.fast-sales.ru)



## Руководство разработчика **Fast-Sales Pro**

(для версии 2.4.4)

## Оглавление

<b>О руководстве .....</b>	<b>39</b>
<b>1. Архитектура системы и принципы ее функционирования .....</b>	<b>40</b>
1.1 Общая архитектура .....	40
1.2 Описание директорий и файлов .....	41
Директория admin .....	42
Директория backups .....	43
Директория images .....	43
Директория includes .....	45
Директория logs .....	46
Директория modules .....	46
Директория payments .....	46
Директория shippings .....	47
Директория templates .....	47
Директория uploads .....	47
Файл .htaccess .....	48
Файл bug_report.php .....	48
Файл captcha.php .....	48
Файл check_requirements.php .....	48
Файл crc32.dat .....	49
Файл index.php .....	49
Файл messages.php .....	49
Файл mind_report.php .....	49
Файл robots.php .....	49
Файл robots.txt .....	49
Файл setup.php .....	50
Файл thumb.php .....	50
1.3 Работа модулей .....	50
<b>2. Создание модуля .....</b>	<b>53</b>
2.1 Программный каркас модуля .....	53
2.2 Состав функции модуля .....	57
2.2.1 Типы функций модуля .....	57
2.2.2 Сохраняющая/изменяющая данные в БД функция .....	57
2.2.3 Выводящая данные из БД функция .....	59
2.3 Страницы модуля в админке .....	60
2.4 Страницы модуля на витрине .....	60
2.5 Многоязычные особенности .....	62

2.5.1	Текстовые метки.....	62
2.5.2	Тексты, формируемые модулем .....	63
2.6	Распределение прав администраторов .....	64
2.7	Вспомогательные AJAX-модули .....	65
2.8	Написание тестового модуля «Hello, world!» .....	69
2.8.1	Постановка задачи .....	69
2.8.2	Создание таблицы модуля в БД .....	70
2.8.3	Создание класса модуля для взаимодействия с БД.....	71
2.8.4	Программирование модуля в админке .....	73
2.8.5	Программирование модуля на витрине .....	78
2.9	Создание модуля на примере модуля «Распродажа» .....	81
<b>3.</b>	<b>Взаимодействие с базой данных .....</b>	<b>82</b>
3.1	Написание классов.....	82
3.2	Правильное удаление информации из БД.....	83
<b>4.</b>	<b>Управление кэшированием.....</b>	<b>85</b>
<b>5.</b>	<b>Встраивание платежных систем.....</b>	<b>86</b>
<b>6.</b>	<b>Смена/модификация дизайна и его интеграция .....</b>	<b>87</b>
6.1	Введение.....	87
6.2	Структура директории скина витрины Fast-Sales .....	88
6.3	Структура дизайна витрины.....	90
6.4	Принцип подключения шаблонов модулей Fast-Sales .....	91
6.5	Процесс добавления или смены дизайна витрины .....	91
6.6	Глобальные переменные CMS в шаблонах дизайна .....	94
6.7	Примеры кода и работы шаблонов Fast-Sales.....	94
6.8	Нюансы и полезные советы.....	94
<b>7.</b>	<b>Требования .....</b>	<b>96</b>
7.1	Требования к именованию элементов кода .....	96
7.1.1	Имена глобальных переменных .....	96
7.1.2	Имена локальных переменных .....	96
7.1.3	Имена констант.....	96
7.1.4	Имена функций.....	96
7.1.5	Имена классов и их методов .....	97
7.1.6	Имена таблиц и полей БД .....	97
7.1.7	Имена текстовых меток интерфейса .....	98
7.2	Требования к именованию объектов файловой системы .....	98
7.3	Требования к оформлению PHP-кода и кода шаблонов дизайна .....	99
7.3.1	PHP-код .....	99

7.3.2 HTML и Smarty-код .....	102
7.3.3 JavaScript-код.....	108
7.4 Прочие требования .....	108
<b>8. Справочник.....</b>	<b>110</b>
8.1 Глобальные переменные в коде PHP .....	110
\$_SESSION['admin_info'] .....	110
\$_SESSION['customer_info'] .....	110
\$_SESSION['visitor_id'].....	110
\$BASE_CURRENCY .....	111
\$CACHE .....	111
\$CBLOCKS .....	111
\$CONFIG .....	112
\$CURRENCIES.....	112
\$DB.....	112
\$LABEL .....	113
\$LANGUAGES .....	113
\$LANGUAGES_COUNT .....	113
\$LBLOCKS .....	113
\$LINKS .....	114
\$LOCATION .....	114
\$META_TAGS.....	114
\$RBLOCKS .....	114
\$SHOP_CURRENCY .....	115
\$SHOP_LANGUAGE .....	115
\$SMARTY .....	115
\$TABLES.....	115
\$TABLES_RELS .....	116
\$TABS .....	117
\$msgs.....	117
\$err .....	117
8.2 Глобальные переменные в коде Smarty.....	117
\$BANNERS_PATH .....	117
\$BASE_CURRENCY .....	118
\$BASE_URL.....	118
\$CATS_PICS .....	118
\$CONFIG .....	118
\$CURRENCIES.....	118
\$CURRENCIES_COUNT.....	118
\$CURRENT_SECTION.....	118

\$DEMO_VERSION .....	119
\$error404 .....	119
\$FLAGS_PICS .....	119
\$IMGS_DIR .....	119
\$is_checkout .....	119
\$LABEL .....	119
\$LANGUAGES .....	119
\$LANGUAGES_COUNT .....	120
\$LIBS_PATH .....	120
\$LINKS .....	120
\$LOCATION .....	120
\$MANUF_PICS .....	120
\$META_TAGS .....	120
\$module .....	120
\$NEWS_PICS .....	120
\$PRODS_PICS .....	121
\$ROOT_DIR .....	121
\$SHOP_CURRENCY .....	121
\$SHOP_LANGUAGE .....	121
\$TABS .....	121
\$TPLS_PATH .....	121
8.3 Константы .....	121
_DEBUG .....	121
_FAST_SALES_ADMIN_STARTED_ .....	122
_FAST_SALES_STARTED_ .....	122
BACKUP_PATH .....	122
BANNERS_PATH .....	122
BASE_URL .....	122
BASE_URL_REW .....	123
CACHE_ENABLED .....	123
CATS_PICS .....	123
CHECK_REQUIREMENTS .....	123
COLOR_SCHEMES_ENABLED .....	123
CURRENT_TIMESTAMP .....	123
DB_HOST .....	124
DB_NAME .....	124
DB_NAME_PRECISION .....	124
DB_PASS .....	124
DB_USER .....	124
DEMO_VERSION .....	124

EGOODS_PATH.....	125
ENABLE_ERROR_LOG_ROTATION.....	125
ERRORS_LOG_FILE.....	125
ERRORS_LOG_LIMIT .....	125
EXPORT_PATH.....	125
FAULT_LOGIN_ATTEMPTS .....	125
FCKE_PATH .....	126
FLAGS_PICS .....	126
FONTS_PATH.....	126
FS_ONLY_TABLES .....	126
IMGS_DIR .....	126
KCAPTCHA_PATH.....	127
LIBS_PATH .....	127
MAIL_TEXTS_PATH.....	127
MANUF_PICS .....	127
MEMCACHE_ALLOWED.....	127
MEMCACHE_FLUSH_ON_RESTART .....	128
MEMCACHE_HOST .....	128
MEMCACHE_PORT .....	128
NEWS_PICS .....	128
NO_PICTURE .....	128
NOWM_PRODS_PICS .....	128
ROOT_DIR.....	128
SHOP_MODIF.....	129
SHOP_VERSION .....	129
TMP_PATH .....	129
TPLS_PATH.....	129
WM_PRODS_PICS.....	129
WMARK_PNG .....	130
WMARK_PNG_THUMB.....	130
WMARKS_PICS.....	130
8.4 Функции (PHP) .....	130
fn_301_redirect .....	130
fn_404_not_found .....	130
fn_add_to_cart .....	131
fn_cblock_enabled .....	131
fn_check_admin_rights .....	132
fn_check_phone_number.....	132
fn_check_user_fields.....	132
fn_del_db_entity.....	133

fn_filter_city .....	133
fn_explode.....	134
fn_get_countries.....	134
fn_img_check .....	134
fn_incline_word .....	135
fn_is_urlname_correct.....	135
fn_json2array.....	136
fn_make_http_post_request.....	136
fn_month2str .....	136
fn_msgs_array_to_str.....	137
fn_price_round .....	137
fn_set_products_list .....	137
fn_smarty_compile_tpls.....	138
fn_translit.....	139
fn_user_profile_fields .....	139
func_check_email.....	139
func_check_install_dir .....	140
func_check_ip .....	140
func_clear_dir.....	140
func_convert_sum .....	141
func_correct_engname.....	141
func_correct_number .....	142
func_count_array.....	142
func_course_correct.....	142
func_file_get_contents .....	143
func_file_put_contents .....	143
func_filesize.....	143
func_get_discounts .....	144
func_get_month_len.....	144
func_get_taxes.....	144
func_img_resize .....	145
func_mail_admin_cancel_order_notify.....	145
func_mail_admin_cancel_order_prods_notify .....	147
func_mail_admin_order_notify .....	148
func_mail_admin_product_review_notify .....	150
func_mail_admin_shop_review_notify .....	150
func_mail_confirm_news.....	151
func_mail_customer_free_egood_link .....	151
func_mail_fail_order .....	152
func_mail_news.....	154

func_mail_newstatus_order .....	154
func_mail_partner_notify .....	156
func_mail_pwd_recover .....	156
func_mail_success_order .....	157
func_mail_user_activated .....	159
func_mail_user_registered .....	159
func_msgs_append .....	160
func_msgs_assign .....	160
func_msgs_create .....	161
func_msgs_process .....	161
func_price_correct .....	161
func_save_watermarked_png .....	162
func_save_watermarked_text .....	162
func_send_mail .....	163
func_shop_err_handler .....	163
func_stripslashes .....	164
func_to_json .....	164
func_wrap_quotes .....	164
prod_options_variants_price .....	165
prods_process .....	165
8.5 Функции (JavaScript) .....	166
BannerClicked .....	166
ClearMsgs .....	166
ConfirmDialog .....	166
ConfirmSubmitForm .....	167
CorrectUrlName .....	167
dec2hex .....	168
getRGB .....	168
GetSelect .....	168
HideFastAuxPanel .....	169
isCookieEnabled .....	169
ProcessMsgs .....	169
ReloadCaptcha .....	170
SelectAllToDel .....	170
SelectAllToVis .....	171
set_active_tab .....	171
ShopHideSelectCat .....	171
ShopSelectCat .....	172
ShopSelectCategory .....	172
ShopShowSubcats .....	173



ShowBlockUI .....	173
ShowFastAuxPanel .....	174
ShowLoading.....	174
SubmitAction .....	175
SubmitForm.....	175
Translit.....	175
trim.....	176
UnSelectAllToDel .....	176
UnSelectAllToVis.....	177
<b>8.6 Классы и их методы .....</b>	<b>177</b>
<b>Класс Access .....</b>	<b>177</b>
Access::AddIP .....	177
Access::DeleteIP .....	178
Access::GetAllIP .....	178
Access::GetIPReason .....	178
Access::IPToId .....	179
Access::IsRestrictedByIP .....	179
Access::UpdateIP .....	179
<b>Класс AdminGroups.....</b>	<b>180</b>
AdminGroups::Add .....	180
AdminGroups::Delete .....	180
AdminGroups::GetAll .....	180
AdminGroups::GetOne .....	181
AdminGroups::Update .....	181
<b>Класс AdminMenus.....</b>	<b>182</b>
AdminMenus::GetAll .....	182
<b>Класс AdvLinks.....</b>	<b>182</b>
AdvLinks::AddRoot .....	182
AdvLinks::AddText .....	183
AdvLinks::Delete .....	183
AdvLinks::GetAll .....	183
AdvLinks::GetOne .....	184
AdvLinks::SetPos .....	184
AdvLinks::Update .....	184
<b>Класс AcquiredProducts .....</b>	<b>185</b>
AcquiredProducts::Add .....	185
AcquiredProducts::Delete .....	185
AcquiredProducts::DeleteByUser .....	186
AcquiredProducts::GetAll .....	186
<b>Класс Articles.....</b>	<b>186</b>

Articles::Add .....	186
Articles::CategoryAdd .....	187
Articles::CategoryDelete .....	188
Articles::CategoryGetAll .....	188
Articles::CategoryGetAllFullest .....	188
Articles::CategoryGetIdByName .....	189
Articles::CategoryGetOne .....	189
Articles::CategoryGetPath .....	189
Articles::CategoryIsNameExists .....	190
Articles::CategoryPathToCaptions .....	190
Articles::CategorySetCaption .....	191
Articles::CategorySetPos .....	191
Articles::CategoryUpdate .....	192
Articles::Delete .....	192
Articles::GetAll .....	192
Articles::GetAllByProduct .....	193
Articles::GetOne .....	193
Articles::GetOneByName .....	194
Articles::IsNameExists .....	194
Articles::RecProdAdd .....	195
Articles::RecProdDelete .....	195
Articles::RecProdGetAll .....	195
Articles::RecProdSetPos .....	196
Articles::SetCaption .....	196
Articles::SetPos .....	196
Articles::SetShowInList .....	197
Articles::Update .....	197
<b>Класс Banners .....</b>	<b>197</b>
Banners::Add .....	197
Banners::AddPlace .....	198
Banners::AllPagesPlaces .....	199
Banners::Click .....	199
Banners::Delete .....	200
Banners::DeletePlace .....	200
Banners::DelPlaceBanners .....	200
Banners::FindIntersepts .....	201
Banners::GetAll .....	201
Banners::GetAllPlaces .....	202
Banners::GetClicksStat .....	203
Banners::GetFilename .....	203
Banners::GetLocation .....	203

Banners::GetName .....	204
Banners::GetOne .....	204
Banners::GetOnePlace .....	204
Banners::GetPages .....	204
Banners::GetPlaceName .....	205
Banners::GetStat .....	205
Banners::GetWeight .....	206
Banners::SetPlace .....	206
Banners::Showed .....	206
Banners::Update .....	207
Banners::UpdatePlace .....	207
<b>Класс Bestsellers .....</b>	<b>207</b>
Bestsellers::GetAll .....	207
Bestsellers::GetCount .....	208
Bestsellers::GetIds .....	208
Bestsellers::SetSoldQty .....	209
<b>Класс Bill .....</b>	<b>209</b>
\$bill->Show .....	209
<b>Класс BillOfParcels .....</b>	<b>210</b>
\$bop->Generate .....	211
<b>Класс Cache .....</b>	<b>212</b>
\$CACHE->Clean .....	212
\$CACHE->Get .....	212
\$CACHE->IsActual .....	212
\$CACHE->Set .....	213
\$CACHE->UpdateTimestamp .....	213
<b>Класс Cart .....</b>	<b>214</b>
Cart::AddProduct .....	214
Cart::CalcDiscount .....	215
Cart::CalcToShow .....	215
Cart::Clear .....	215
Cart::ClearRTShippingDates .....	216
Cart::ClearRTShippingPrices .....	216
Cart::DelAllProducts .....	216
Cart::DelProduct .....	216
Cart::GetAllowEgoodsShipping .....	217
Cart::GetCurrentCarts .....	217
Cart::GetCurrentCartsCount .....	218
Cart::GetOrderID .....	218
Cart::GetPaymentID .....	218
Cart::GetProdsCount .....	218

Cart::GetProduct .....	219
Cart::GetProds .....	219
Cart::GetProdPrice .....	220
Cart::GetQty .....	220
Cart::GetRTShippingDates .....	220
Cart::GetRTShippingPrice .....	221
Cart::GetShipping .....	221
Cart::GetSum .....	221
Cart::GetWeight .....	222
Cart::IsOnlyEgoods .....	222
Cart::ReturnExpired .....	222
Cart::SaveComment .....	223
Cart::SetAllowEgoodsShipping .....	223
Cart::SetOrderID .....	223
Cart::SetPaymentID .....	223
Cart::SetRTShippingDates .....	224
Cart::SetRTShippingPrice .....	224
Cart::SetShipping .....	224
Cart::SetWeight .....	225
Cart::UpdateProduct .....	225
<b>Класс CartesianProduct.....</b>	<b>225</b>
CartesianProduct::Get .....	225
<b>Класс Categories .....</b>	<b>226</b>
Categories::Add .....	226
Categories::ContainsProduct .....	227
Categories::Delete .....	227
Categories::DeleteImage .....	227
Categories::GetAll .....	227
Categories::GetAllFloat .....	228
Categories::GetDiscountByCat .....	229
Categories::GetIdByName .....	229
Categories::GetNameById .....	229
Categories::GetOne .....	230
Categories::GetPath .....	230
Categories::GetSubcatsCount .....	231
Categories::GetSubIds .....	231
Categories::GetTaxByCat .....	231
Categories::GetTotalCount .....	232
Categories::IdToCaption .....	232
Categories::IdToCaptions .....	232
Categories::IsCaptionExists .....	233

Categories::IsExists	233
Categories::IsNameExists	233
Categories::IsVisible	234
Categories::ModPrices	234
Categories::PathToCaptions	234
Categories::RecalcProductsCount	235
Categories::SetDiscount	235
Categories::SetImage	235
Categories::SetPos	236
Categories::SetProductsCount	236
Categories::SetTax	236
Categories::SetVisible	237
Categories::Update	237
Categories::UploadImage	237
<b>Класс CBlocks</b>	<b>238</b>
CBlocks::GetAll	238
CBlocks::GetColumn	239
CBlocks::IsColumnFixed	239
CBlocks::SetColumn	239
CBlocks::SetInHomepage	239
CBlocks::SetNotInHomepage	240
CBlocks::SetWeight	240
<b>Класс Config</b>	<b>240</b>
Config::GetAllGroups	240
Config::GetAllSettings	241
Config::GetFullSettings	242
Config::GetSettingByName	242
Config::GetSettingType	242
Config::Init	243
Config::LoadModules	243
Config::LoadOptions	243
Config::LoadSettings	244
Config::UpdateSettings	244
<b>Класс Controller</b>	<b>244</b>
Controller::Controller	244
Controller::Run	244
Controller::SetDefModule	245
Controller::SetModules	245
<b>Класс Countries</b>	<b>245</b>
Countries::Add	246
Countries::AddState	246

Countries::CodeToCountry .....	246
Countries::CountryToCode .....	247
Countries::Delete .....	247
Countries::DeleteState .....	247
Countries::GetAll .....	248
Countries::GetAllCities .....	248
Countries::GetAllStates .....	248
Countries::GetAllVisible .....	249
Countries::GetCity .....	249
Countries::GetName .....	249
Countries::GetOne .....	249
Countries::GetOneState .....	250
Countries::GetStateName .....	250
Countries::IsCodeExists .....	250
Countries::IsNameExists .....	251
Countries::IsStateCodeExists .....	251
Countries::IsStateNameExists .....	251
Countries::Update .....	252
Countries::UpdateState .....	252
<b>Класс CThemes.....</b>	<b>253</b>
CThemes::GetAll .....	253
CThemes::GetColors .....	253
CThemes::GetFirstColor .....	254
CThemes::GetThemeCaption .....	254
CThemes::GetThemes .....	254
<b>Класс Currencies .....</b>	<b>254</b>
Currencies::Add .....	255
Currencies::Delete .....	255
Currencies::GetAll .....	255
Currencies::GetBase .....	256
Currencies::GetBaseID .....	256
Currencies::GetOne .....	256
Currencies::GetOneByName .....	257
Currencies::SetBaseID .....	257
Currencies::SetCourse .....	257
Currencies::Update .....	258
<b>Класс DB.....</b>	<b>258</b>
\$DB->Close .....	258
\$DB->DB .....	258
\$DB->Delete .....	259
\$DB->GetAssocs .....	259

\$DB->GetError .....	260
\$DB->GetLastId .....	260
\$DB->GetRow .....	260
\$DB->GetRows .....	261
\$DB->GetRowsNum .....	261
\$DB->GetServerVersion .....	261
\$DB->Insert .....	262
\$DB->IsConnected .....	262
\$DB->IsError .....	262
\$DB->Query .....	262
\$DB->Replace .....	263
\$DB->Select .....	263
DB::TestConnect .....	264
\$DB->Update .....	264
<b>Класс DBBackup.....</b>	<b>264</b>
DBBackup::AppendDump .....	264
DBBackup::GetDump .....	265
DBBackup::GetDumpsList .....	265
DBBackup::GetUploadedFile .....	265
<b>Класс DBClear.....</b>	<b>266</b>
DBClear::Access .....	266
DBClear::Articles .....	266
DBClear::Categories .....	266
DBClear::Discounts .....	267
DBClear::LoginHistory .....	267
DBClear::Manufacturers .....	267
DBClear::Messages .....	267
DBClear::MetaTags .....	268
DBClear::News .....	268
DBClear::NewsSubscribers .....	268
DBClear::Orders .....	269
DBClear::Products .....	269
DBClear::Stats .....	269
DBClear::Taxes .....	269
DBClear::Votes .....	270
<b>Класс Discounts.....</b>	<b>270</b>
Discounts::AddRoot .....	270
Discounts::AddText .....	270
Discounts::CartAdd .....	271
Discounts::CartDelete .....	271
Discounts::CartGetAll .....	271

Discounts::CartGetAmount .....	272
Discounts::CartGetCount .....	272
Discounts::Delete .....	272
Discounts::ETypeToId .....	273
Discounts::GetActiveConf .....	273
Discounts::GetActiveCount .....	273
Discounts::GetAll .....	274
Discounts::GetConfList .....	274
Discounts::GetGlobals .....	274
Discounts::GetOne .....	275
Discounts::GetTypes .....	275
Discounts::IsTextExists .....	275
Discounts::SetActiveConf .....	276
Discounts::Update .....	276
<b>Класс eDost .....</b>	<b>276</b>
\$edost->Calc .....	277
\$edost->SetSiteUTF .....	277
\$edost->SetSiteWin .....	278
<b>Класс EMSPost.....</b>	<b>278</b>
EMSPost::Calculate .....	278
EMSPost::GetLocations .....	279
EMSPost::GetMaxWeight .....	279
EMSPost::GetOption .....	279
EMSPost::SetOption .....	280
<b>Класс ErrorLogger .....</b>	<b>280</b>
ErrorLogger::ArrayToXML .....	280
ErrorLogger::Clear .....	280
ErrorLogger::DeleteOldNErrorsFromDB .....	281
ErrorLogger::DeleteOldNErrorsFromFile .....	281
ErrorLogger::GetAllXML.....	281
ErrorLogger::GetAllXMLFromDB .....	282
ErrorLogger::GetAllXMLFromFile .....	282
ErrorLogger::GetDBLogLength .....	282
ErrorLogger::GetFileLogLength .....	282
ErrorLogger::LogError .....	283
ErrorLogger::SaveErrorToDB .....	283
ErrorLogger::SaveErrorToFile .....	284
<b>Класс Excel.....</b>	<b>284</b>
\$csv->AddHeader .....	284
\$csv->AddItem .....	285
\$csv->Escape .....	285



\$csv->GetCSV .....	285
\$csv->GetCSVFileName .....	286
\$csv->LoadHeader .....	286
\$csv->Load .....	286
\$csv->Save .....	287
\$csv->UnEscape .....	287
<b>Класс FastMenu.....</b>	<b>287</b>
FastMenu::AddRoot .....	287
FastMenu::AddTexts .....	288
FastMenu::Delete .....	288
FastMenu::GetAll .....	288
FastMenu::GetAllVisible .....	289
FastMenu::SetPos .....	289
FastMenu::Update .....	289
<b>Класс FeaturedProducts.....</b>	<b>290</b>
FeaturedProducts::Delete .....	290
FeaturedProducts::GetAll .....	290
FeaturedProducts::GetIds .....	290
FeaturedProducts::IsFeatured .....	291
FeaturedProducts::Remove .....	291
FeaturedProducts::Set .....	291
FeaturedProducts::SetPos .....	291
<b>Класс FedEx .....</b>	<b>292</b>
\$fedex->getCurrency .....	292
\$fedex->getDates .....	292
\$fedex->getPrice .....	292
\$fedex->sendRequest .....	293
\$fedex->setAccountNumber .....	293
\$fedex->setCarrierCode .....	293
\$fedex->setDestCountryCode .....	294
\$fedex->setDestPostalCode .....	294
\$fedex->setDestStateOrProvinceCode .....	294
\$fedex->setDropoffType .....	294
\$fedex->setHeight .....	295
\$fedex->setLength .....	295
\$fedex->setMeterNumber .....	295
\$fedex->setOriginCountryCode .....	296
\$fedex->setOriginPostalCode .....	296
\$fedex->setOriginStateOrProvinceCode .....	296
\$fedex->setPackaging .....	296
\$fedex->setServer .....	297

\$fedex->setUnits .....	297
\$fedex->setWeight .....	297
\$fedex->setWeightUnits .....	297
\$fedex->setWidth .....	298
<b>Класс GBaseProds .....</b>	<b>298</b>
\$gbase->AddProduct .....	298
\$gbase->Escape .....	299
\$gbase->GetGBaseFileName .....	299
\$gbase->GetXML .....	299
\$gbase->Save .....	299
\$gbase->SetCurrency .....	300
\$gbase->SetShopDescr .....	300
\$gbase->SetShopName .....	300
\$gbase->SetShopUrl .....	301
<b>Класс HomePage.....</b>	<b>301</b>
HomePage::GetContent .....	301
HomePage::UpdateContent .....	301
<b>Класс InterShipper .....</b>	<b>302</b>
\$is->getCurrency .....	302
\$is->getDates .....	302
\$is->getPrice .....	302
\$is->sendRequest .....	303
\$is->setARBAccountNumber .....	303
\$is->setARBInvoiced .....	303
\$is->setCANAccountNumber .....	303
\$is->setCANInvoiced .....	304
\$is->setDeliveryType .....	304
\$is->setDestCountry .....	304
\$is->setDestPostal .....	304
\$is->setDHLAccountNumber .....	305
\$is->setDHLInvoiced .....	305
\$is->setDimHeight .....	305
\$is->setDimLength .....	305
\$is->setDimUnit .....	306
\$is->setDimWidth .....	306
\$is->setFDXAccountNumber .....	306
\$is->setFDXInvoiced .....	307
\$is->setOptions .....	307
\$is->setOrigCountry .....	307
\$is->setOrigPostal .....	307
\$is->setPackaging .....	308

\$is->setPassword .....	308
\$is->setServer .....	308
\$is->setShipMethod .....	308
\$is->setUPSAccountNumber .....	309
\$is->setUPSInvoiced .....	309
\$is->setUsername .....	309
\$is->setUSPAccountNumber .....	310
\$is->setUSPInvoiced .....	310
\$is->setWeight .....	310
<b>Класс Labels .....</b>	<b>310</b>
Labels::Add .....	310
Labels::Delete .....	311
Labels::GetAdminCount .....	311
Labels::GetAll .....	312
Labels::GetAllEmpty .....	312
Labels::GetCaption .....	313
Labels::GetCustomerCount .....	313
Labels::GetOne .....	313
Labels::GetTotalCount .....	314
Labels::IsNameExists .....	314
Labels::Search .....	314
Labels::SetCaption .....	315
Labels::Update .....	315
<b>Класс Languages.....</b>	<b>315</b>
Languages::Add .....	316
Languages::BindCurrencies .....	316
Languages::CopyTexts .....	316
Languages::Delete .....	317
Languages::Exists .....	317
Languages::GetAll .....	317
Languages::GetCurrencies .....	317
Languages::GetDefault .....	318
Languages::GetDefaultID .....	318
Languages::GetIDByShort .....	318
Languages::GetIDByTitle .....	319
Languages::GetOne .....	319
Languages::GetShortByID .....	319
Languages::GetTitleByID .....	319
Languages::IsCurrencyAllowed .....	320
Languages::SetDefaultID .....	320
Languages::Update .....	320

<b>Класс Links</b> .....	<b>321</b>
\$LINKS->Account .....	321
\$LINKS->AccountActivate .....	321
\$LINKS->AccountPwdRecover .....	321
\$LINKS->Article .....	322
\$LINKS->Articles .....	322
\$LINKS->Bestsellers .....	322
\$LINKS->Cart .....	323
\$LINKS->CompareProds .....	323
\$LINKS->Contacts .....	323
\$LINKS->GetEgoodFile .....	323
\$LINKS->GetEgoodFree .....	324
\$LINKS->Home .....	324
\$LINKS->Manufacturer .....	324
\$LINKS->Manufacturers .....	325
\$LINKS->Newest .....	325
\$LINKS->News .....	325
\$LINKS->NewsArchive .....	325
\$LINKS->NewsConfirm .....	326
\$LINKS->NewsRSS .....	326
\$LINKS->NewsSubscribe .....	326
\$LINKS->NewsUnsubscribe .....	326
\$LINKS->OrderStatus .....	327
\$LINKS->Partner .....	327
\$LINKS->PriceList .....	327
\$LINKS->Product .....	328
\$LINKS->Products .....	328
\$LINKS->ProductsByManuf .....	328
\$LINKS->ProductsByPrice .....	329
\$LINKS->ReviewAdd .....	329
\$LINKS->Reviews .....	329
\$LINKS->Search .....	330
\$LINKS->SiteMap .....	330
\$LINKS->Tags .....	330
<b>Класс Location</b> .....	<b>331</b>
\$LOCATION->AddItem .....	331
\$LOCATION->AppendItem .....	331
\$LOCATION->ClearList .....	331
<b>Класс Manufacturers</b> .....	<b>332</b>
Manufacturers::AddRoot .....	332
Manufacturers::AddTexts .....	332

Manufacturers::Delete .....	333
Manufacturers::DeleteImage .....	333
Manufacturers::GetAll .....	333
Manufacturers::GetAllVisible .....	334
Manufacturers::GetIdByName .....	334
Manufacturers::GetOne .....	335
Manufacturers::GetOneByName .....	335
Manufacturers::IsExists .....	336
Manufacturers::IsCaptionExists .....	336
Manufacturers::IsNameExists .....	337
Manufacturers::SetCaption .....	337
Manufacturers::SetImage .....	337
Manufacturers::SetPos .....	337
Manufacturers::SetVisible .....	338
Manufacturers::Update .....	338
Manufacturers::UploadImage .....	339
<b>Класс Messages .....</b>	<b>339</b>
Messages::AddMessage .....	339
Messages::AddTopic .....	340
Messages::GetTopicById .....	340
Messages::GetTopicsByStatus .....	341
Messages::GetTopicsByUser .....	341
Messages::SetTopicStatus .....	341
Messages::SetTopicUserRead .....	342
<b>Класс MetaTags.....</b>	<b>342</b>
MetaTags::AddMeta .....	342
MetaTags::DeleteMeta .....	343
MetaTags::GetMeta .....	343
MetaTags::UpdateMeta .....	344
<b>Класс Modules .....</b>	<b>344</b>
Modules::GetAdminActions .....	344
Modules::GetAll .....	345
Modules::GetOne .....	345
Modules::GetOneByName .....	345
Modules::Update .....	346
<b>Класс Newest.....</b>	<b>346</b>
Newest::GetAll .....	346
Newest::GetIds .....	346
<b>Класс News.....</b>	<b>347</b>
News::Add .....	347
News::AddSubscriber .....	348

News::ConfirmSubscriber .....	348
News::Delete .....	348
News::DeleteImage .....	349
News::DelSubscriber .....	349
News::DelSubscriberById .....	349
News::DelSubscriberByKey .....	349
News::EditHistory .....	350
News::GetAll .....	350
News::GetAllSubscribers .....	351
News::GetConfirmedSubscribers .....	351
News::GetOne .....	352
News::GetOneSubscriber .....	352
News::IsSubscriberExists .....	352
News::SetCaption .....	353
News::SetSentFlag .....	353
News::SetShowInHomepage .....	353
News::Update .....	354
News::UpdateSubscriber .....	354
News::UploadImage .....	354
<b>Класс Options .....</b>	<b>355</b>
Options::Get .....	355
Options::Set .....	355
<b>Класс Orders.....</b>	<b>355</b>
Orders::Add .....	356
Orders::AddOrderedProd .....	356
Orders::CancelOrderedProd .....	357
Orders::Delete .....	357
Orders::DelOrderedProd .....	357
Orders::GetAll .....	358
Orders::GetAllByCustomer .....	358
Orders::GetCount .....	359
Orders::GetCustomers .....	359
Orders::GetOne .....	359
Orders::GetOrderedProd .....	360
Orders::GetOrderedProds .....	360
Orders::GetPendingProds .....	360
Orders::GetStatuses .....	362
Orders::GetStatusesHistory .....	362
Orders::GetStatusId .....	363
Orders::GetTotalCount .....	363
Orders::MayCancelStatus .....	363

Orders::MayChangedStatus .....	364
Orders::ReadyForShipment .....	364
Orders::SetChangeEmailed .....	364
Orders::SetComment .....	364
Orders::SetDiscountPer .....	365
Orders::SetEmailed .....	365
Orders::SetProdPending .....	365
Orders::SetProdQty .....	366
Orders::SetProdSupplierNotified .....	366
Orders::SetShippingInfo .....	366
Orders::SetStatus .....	367
Orders::SetStatusCaption .....	367
Orders::StatusIdToString .....	367
Orders::StatusWas .....	368
Orders::UpdateCustomerInfo .....	368
Orders::UpdateOrderedProd .....	368
Orders::UpdateSum .....	369
Orders::UpdateWeight .....	369
<b>Класс Pager .....</b>	<b>369</b>
\$pager->getCurrentPage .....	369
\$pager->getNextPage .....	370
\$pager->getOrderField .....	370
\$pager->getOrderHref .....	370
\$pager->getPagesCount .....	371
\$pager->getPageSize .....	371
\$pager->getPaging .....	371
\$pager->getPrevPage .....	372
\$pager->getSortDir .....	372
\$pager->getSQL .....	372
\$pager->Pager .....	373
\$pager->setDefaultSort .....	373
\$pager->setPage .....	373
\$pager->setPageSize .....	374
\$pager->setRecordCount .....	374
\$pager->setSort .....	374
\$pager->setSortings .....	374
\$pager->setViewPages .....	375
<b>Класс Payments.....</b>	<b>375</b>
Payments::AddManual .....	375
Payments::DelManual .....	376
Payments::GetAll .....	376

Payments::GetAllActive .....	376
Payments::GetAllSettings .....	377
Payments::GetCaption .....	377
Payments::GetOne .....	377
Payments::GetSettingValue .....	378
Payments::IsNameExists .....	378
Payments::SaveTransactionData .....	378
Payments::SetActive .....	379
Payments::SetAuto .....	379
Payments::SetCaption .....	379
Payments::SetDescr .....	380
Payments::SetLastStepText .....	380
Payments::SetPos .....	380
Payments::SetSettingValue .....	381
<b>Класс ProdDetails .....</b>	<b>381</b>
ProdDetails::AddDetailRoot .....	381
ProdDetails::AddDetailValue .....	381
ProdDetails::AddParamRoot .....	382
ProdDetails::AddParamText .....	382
ProdDetails::DelDetail .....	383
ProdDetails::DelDetailsByParamID .....	383
ProdDetails::DelParam .....	383
ProdDetails::GetAllDetails .....	384
ProdDetails::GetAllNonEmptyDetails .....	384
ProdDetails::GetAllParams .....	385
ProdDetails::GetDetail .....	385
ProdDetails::GetDetailValues .....	385
ProdDetails::GetOneDetail .....	386
ProdDetails::GetParam .....	386
ProdDetails::GetParamsCount .....	386
ProdDetails::IsDetailExists .....	387
ProdDetails::SetDetailValue .....	387
ProdDetails::SetParamCaption .....	387
ProdDetails::SetParamFilterFlag .....	388
ProdDetails::SetPos .....	388
<b>Класс ProdFiles .....</b>	<b>388</b>
ProdFiles::ClearActiveUser .....	388
ProdFiles::CreateKey .....	389
ProdFiles::Delete .....	389
ProdFiles::DeleteKey .....	390
ProdFiles::DelExpiredKeys .....	390



ProdFiles::FileExists .....	390
ProdFiles::Get .....	390
ProdFiles::GetAllKeys .....	391
ProdFiles::GetAllKeysCount .....	391
ProdFiles::GetIdByProdId .....	391
ProdFiles::GetOne .....	392
ProdFiles::GetOneKey .....	392
ProdFiles::GetProdIdById .....	392
ProdFiles::GetTimeToAllowFree .....	392
ProdFiles::IsActiveUser .....	393
ProdFiles::ProlongKey .....	393
ProdFiles::SetActiveUser .....	394
ProdFiles::SetKeyAllowed .....	394
ProdFiles::SetKeyLastAccess .....	394
ProdFiles::SetName .....	395
ProdFiles::SetPath .....	395
ProdFiles::Upload .....	395
<b>Класс ProdOptions.....</b>	<b>396</b>
ProdOptions::AddException .....	396
ProdOptions::AddRoot .....	396
ProdOptions::AddTexts .....	396
ProdOptions::AddVarRoot .....	397
ProdOptions::AddVarTexts .....	397
ProdOptions::AllowOnlyVariants .....	398
ProdOptions::ClearDefVarsCombination .....	398
ProdOptions::CloneGlobal .....	398
ProdOptions::Delete .....	399
ProdOptions::DeleteByProd .....	399
ProdOptions::DeleteVar .....	399
ProdOptions::DelException .....	399
ProdOptions::GetAll .....	400
ProdOptions::GetAllVars .....	400
ProdOptions::GetCaption .....	401
ProdOptions::GetDefVarsCombination .....	401
ProdOptions::GetExceptions .....	401
ProdOptions::GetIdByCaption .....	401
ProdOptions::GetOne .....	402
ProdOptions::GetOneVar .....	402
ProdOptions::GetOptionVariants .....	403
ProdOptions::GetTypes .....	403
ProdOptions::GetVarCaption .....	403

ProdOptions::GetVarIdByCaption .....	404
ProdOptions::IsExceptioned .....	404
ProdOptions::IsOnlyVariantsAllowed .....	404
ProdOptions::IsOptionsCorrect .....	405
ProdOptions::ModifProdByOptions .....	405
ProdOptions::ModifProdPriceByOptions .....	406
ProdOptions::SetDefVarsCombination .....	406
ProdOptions::SetPos .....	406
ProdOptions::SetVarPos .....	407
ProdOptions::Update .....	407
ProdOptions::UpdateVar .....	407
<b>Класс ProdPictures .....</b>	<b>408</b>
ProdPictures::AddRoot .....	408
ProdPictures::AddText .....	408
ProdPictures::Delete .....	409
ProdPictures::GetAll .....	409
ProdPictures::GetAlt .....	409
ProdPictures::LoadImage .....	410
ProdPictures::SetFilename .....	410
ProdPictures::SetPos .....	410
ProdPictures::UpdateAlt .....	411
<b>Класс ProdPins .....</b>	<b>411</b>
ProdPins::Add .....	411
ProdPins::Clear .....	411
ProdPins::Count .....	412
ProdPins::GetAll .....	412
ProdPins::Pop .....	412
<b>Класс ProdsDiscuss.....</b>	<b>412</b>
ProdsDiscuss::Add .....	413
ProdsDiscuss::Approve .....	413
ProdsDiscuss::Decline .....	413
ProdsDiscuss::Delete .....	414
ProdsDiscuss::GetAll .....	414
ProdsDiscuss::GetAllApproved .....	414
ProdsDiscuss::GetAllInapproved .....	415
ProdsDiscuss::GetCount .....	415
ProdsDiscuss::GetInapprovedCount .....	415
ProdsDiscuss::GetTopRated .....	415
ProdsDiscuss::IsDiscussed .....	416
ProdsDiscuss::SetComment .....	416
<b>Класс Products .....</b>	<b>417</b>

# Руководство разработчика Fast-Sales Pro

---

Products::Add	417
Products::AddAdditionalCat	418
Products::DecQty	418
Products::DelAdditionalCat	419
Products::DelAll	419
Products::Delete	419
Products::DelImage	420
Products::DelThumb	420
Products::GetAll	420
Products::GetCatIds	421
Products::GetDiscount	421
Products::GetFinalPrice	422
Products::GetIdByName	422
Products::GetNameById	422
Products::GetOne	423
Products::GetTax	423
Products::GetTexts	423
Products::GetTotalCount	424
Products::IdBySKU	424
Products::IdToCaption	424
Products::IncQty	425
Products::IncSoldQty	425
Products::IsExists	425
Products::IsNameExists	426
Products::IsSKUExists	426
Products::IsVisible	426
Products::GetQty	427
Products::LoadImage	427
Products::LoadThumbnail	427
Products::MakeThumbnail	428
Products::ModPrice	428
Products::Search	428
Products::SearchCount	431
Products::SetCaption	432
Products::SetCatDiscount	432
Products::SetCatTax	432
Products::SetDiscount	432
Products::SetImage	433
Products::SetMainCat	433
Products::SetPosition	433
Products::SetPrice	434

Products::SetQty .....	434
Products::SetSKU .....	434
Products::SetTax .....	435
Products::SetThumb .....	435
Products::SetVisibility .....	435
Products::SetWeight .....	436
Products::Update .....	436
<b>Класс ProdVariants.....</b>	<b>436</b>
ProdVariants::Add .....	436
ProdVariants::DecQty .....	437
ProdVariants::Delete .....	437
ProdVariants::DeleteByProd .....	437
ProdVariants::DelImage .....	438
ProdVariants::DelThumb .....	438
ProdVariants::GetAll .....	438
ProdVariants::GetIdByOptions .....	439
ProdVariants::GetOne .....	439
ProdVariants::GetQty .....	439
ProdVariants::GetSKU .....	440
ProdVariants::GetVarsCombs .....	440
ProdVariants::IncQty .....	440
ProdVariants::LoadImage .....	440
ProdVariants::LoadThumbnail .....	441
ProdVariants::MakeThumb .....	442
ProdVariants::SetImage .....	442
ProdVariants::SetQty .....	442
ProdVariants::SetSKU .....	442
ProdVariants::SetThumb .....	443
ProdVariants::SetWeight .....	443
ProdVariants::Update .....	443
<b>Класс RecommendedProds.....</b>	<b>444</b>
RecommendedProds::Add .....	444
RecommendedProds::Delete .....	444
RecommendedProds::GetAll .....	444
RecommendedProds::SetPos .....	445
<b>Класс RelatedProds.....</b>	<b>445</b>
RelatedProds::Add .....	445
RelatedProds::Delete .....	446
RelatedProds::GetAll .....	446
RelatedProds::SetPos .....	446
<b>Класс RememberMe.....</b>	<b>446</b>

RememberMe::ClearAdmin .....	447
RememberMe::ClearCustomer .....	447
RememberMe::GetAdminAuth .....	447
RememberMe::GetAdminLang .....	447
RememberMe::GetCustomerAuth .....	448
RememberMe::IsAdminAuth .....	448
RememberMe::IsAdminLang .....	448
RememberMe::IsCustomerAuth .....	449
RememberMe::SetAdminAuth .....	449
RememberMe::SetAdminLang .....	449
RememberMe::SetCustomerAuth .....	449
<b>Класс Request .....</b>	<b>450</b>
\$req->DelAllParams .....	450
\$req->DelParamValue .....	450
\$req->GetParamValue .....	450
\$req->ParamValueExists .....	451
\$req->Request .....	451
\$req->SetParamValue .....	452
<b>Класс Reviews .....</b>	<b>452</b>
Reviews::Add .....	452
Reviews::Approve .....	452
Reviews::Decline .....	453
Reviews::Delete .....	453
Reviews::GetAll .....	453
Reviews::GetAllApproved .....	453
Reviews::GetAllInapproved .....	454
Reviews::GetInapprovedCount .....	454
Reviews::SetComment .....	454
<b>Класс RSS .....</b>	<b>455</b>
\$rss->AddItem .....	455
\$rss->Get .....	455
\$rss->GetChannelDescription .....	455
\$rss->GetChannelTitle .....	456
\$rss->GetChannelURL .....	456
\$rss->GetDate .....	456
\$rss->GetImage .....	456
\$rss->GetItems .....	457
\$rss->GetLang .....	457
\$rss->GetRSSVersion .....	457
\$rss->GetXMLVersion .....	458
\$rss->Publish .....	458

\$rss->RSS .....	458
\$rss->SetChannelInfo .....	459
\$rss->SetImage .....	459
\$rss->SetLang .....	459
<b>Класс RusPost .....</b>	<b>459</b>
RusPost::GetAllOptions .....	459
RusPost::GetAllRegions .....	460
RusPost::GetOption .....	460
RusPost::GetRegionByNum .....	460
RusPost::SetAvailAvia .....	461
RusPost::SetOption .....	461
RusPost::SetRegion .....	461
RusPost::SetRegionAddCost .....	462
RusPost::SetRegionCost .....	462
<b>Класс Sberbank .....</b>	<b>462</b>
Sberbank::GenerateReceipt .....	462
<b>Класс Security .....</b>	<b>463</b>
Security::ToInt .....	463
Security::ToSqlStr .....	463
Security::ToStr .....	464
<b>Класс Slider .....</b>	<b>464</b>
Slider::Add .....	464
Slider::Delete .....	465
Slider::GetAll .....	465
Slider::SetPos .....	465
<b>Класс Shippings .....</b>	<b>466</b>
Shippings::CustomGetAll .....	466
Shippings::CustomGetOne .....	466
Shippings::CustomGetOneByName .....	467
Shippings::CustomSetDeliveryTime .....	467
Shippings::CustomSetIsActive .....	467
Shippings::CustomSetTexts .....	468
Shippings::PaymentBindAdd .....	468
Shippings::PaymentBindDelete .....	469
Shippings::PaymentBindGetAll .....	469
Shippings::RTGetAll .....	469
Shippings::RTGetAllActive .....	470
Shippings::RTGetCaption .....	470
Shippings::RTGetGroup .....	470
Shippings::RTGetGroupSettingByName .....	471
Shippings::RTGetOne .....	471

Shippings::RTUpdate .....	471
Shippings::RTUpdateSetting .....	472
Shippings::StaticAddRoot .....	472
Shippings::StaticAddTexts .....	472
Shippings::StaticDelete .....	473
Shippings::StaticGetAll .....	473
Shippings::StaticGetBands .....	473
Shippings::StaticGetCaption .....	474
Shippings::StaticGetCost .....	474
Shippings::StaticGetOne .....	474
Shippings::StaticSetBands .....	475
Shippings::StaticSetTexts .....	475
Shippings::StaticUpdate .....	476
<b>Класс Shortcuts .....</b>	<b>476</b>
Shortcuts::Add .....	476
Shortcuts::Delete .....	476
Shortcuts::GetAll .....	477
<b>Класс SMSGates .....</b>	<b>477</b>
SMSGates::GetActive .....	477
SMSGates::GetAll .....	477
SMSGates::GetAllSettings .....	478
SMSGates::GetOne .....	478
SMSGates::GetSettingValue .....	478
SMSGates::SetActive .....	479
SMSGates::SetSettingValue .....	479
<b>Класс Stats .....</b>	<b>479</b>
Stats::GetCatViews .....	479
Stats::GetLoginCount .....	480
Stats::GetLoginHistory .....	480
Stats::GetOrderedProdsStats .....	480
Stats::GetOrderedSum .....	481
Stats::GetOrdersCount .....	481
Stats::GetProdsCount .....	481
Stats::GetSales .....	482
Stats::GetSearchStats .....	483
Stats::GetUsersCount .....	483
Stats::RegisterCatView .....	483
Stats::RegisterLogin .....	484
Stats::RegisterSearch .....	484
<b>Класс Tabs.....</b>	<b>484</b>
\$TABS->GetActive .....	484

\$TABS->IsActive .....	485
\$TABS->SetActive .....	485
<b>Класс Tags.....</b>	<b>485</b>
Tags::Add .....	485
Tags::AddProdTag .....	486
Tags::Delete .....	486
Tags::DelProdTags .....	486
Tags::GetAll .....	486
Tags::GetProdTags .....	487
Tags::IdToName .....	487
Tags::NameToId .....	487
<b>Класс Taxes.....</b>	<b>488</b>
Taxes::AddRoot .....	488
Taxes::AddText .....	488
Taxes::Delete .....	489
Taxes::GetAll .....	489
Taxes::GetCount .....	489
Taxes::GetOne .....	490
Taxes::IsTextExists .....	490
Taxes::Update .....	490
<b>Класс TplEditor .....</b>	<b>491</b>
TplEditor::CreateDir .....	491
TplEditor::GetContent .....	491
TplEditor::GetDir .....	491
TplEditor::PutContent .....	492
TplEditor::UploadFile .....	492
<b>Класс UriNameHistory .....</b>	<b>493</b>
UriNameHistory::Add .....	493
UriNameHistory::Delete .....	493
UriNameHistory::Find .....	493
<b>Класс Users .....</b>	<b>494</b>
Users::Add .....	494
Users::AddMoney .....	496
Users::Delete .....	496
Users::GenPwdSalt .....	497
Users::GetAll .....	497
Users::GetEmail .....	497
Users::GetFields .....	498
Users::GetIdByType .....	498
Users::GetLangID .....	498
Users::GetOne .....	499



Users::GetPartners .....	499
Users::GetPwdHash .....	499
Users::GetPwdSalt1 .....	500
Users::GetPwdSalt2 .....	500
Users::GetTypeCount .....	500
Users::GetTypes .....	501
Users::GetUserId .....	501
Users::IdByActivationKey .....	501
Users::IdByPwdRecoverKey .....	501
Users::IncTotalOrdersSum .....	502
Users::IsActivated .....	502
Users::IsExists .....	502
Users::IsFieldAvail .....	503
Users::IsFieldRequired .....	503
Users::IsUser .....	503
Users::PayPartner .....	503
Users::SaveField .....	504
Users::SetActivationKey .....	504
Users::SetDiscount .....	504
Users::SetLangID .....	505
Users::SetPwdHash .....	505
Users::SetPwdRecoverKey .....	505
Users::Update .....	506
Users::UpdateCurrency .....	506
<b>Класс Visitors .....</b>	<b>506</b>
Visitors::GetAll .....	506
Visitors::GetCount .....	507
Visitors::GetIP .....	507
Visitors::GetPartnerId .....	507
Visitors::GetUserId .....	508
Visitors::GuestsOnline .....	508
Visitors::Register .....	508
Visitors::SetAsUser .....	509
Visitors::SetPartnerId .....	509
Visitors::UpdateActTime .....	509
Visitors::UsersOnline .....	510
<b>Класс Votes .....</b>	<b>510</b>
Votes::AddVarRoot .....	510
Votes::AddVarText .....	510
Votes::AddVoteRoot .....	511
Votes::AddVoteText .....	511

Votes::CanVote .....	511
Votes::DelVariant .....	511
Votes::DelVote .....	512
Votes::GetActiveVote .....	512
Votes::GetActiveVoteId .....	512
Votes::GetFirstVoteId .....	513
Votes::GetVariant .....	513
Votes::GetVariants .....	513
Votes::GetVote .....	514
Votes::GetVotes .....	514
Votes::IsVarTextExists .....	514
Votes::IsVoteTextExists .....	515
Votes::Register .....	515
Votes::SetActiveVote .....	516
Votes::SetPos .....	516
Votes::SetVarVotes .....	516
Votes::UpdateVariant .....	516
Votes::UpdateVote .....	517
Votes::VoteIdToCaption .....	517
Votes::VoteVar .....	517
<b>Класс Watermark.....</b>	<b>518</b>
Watermark::CreateGraphical .....	518
Watermark::CreateTextual .....	518
<b>Класс WishedProducts .....</b>	<b>519</b>
WishedProducts::Add .....	519
WishedProducts::DeleteByUser .....	519
WishedProducts::DeleteForUser .....	520
WishedProducts::GetAll .....	520
<b>Класс XmlParser.....</b>	<b>520</b>
XmlParser::GetXMLTree .....	520
<b>Класс Ymarket .....</b>	<b>521</b>
\$yaml->AddCategory .....	521
\$yaml->AddCurrency .....	521
\$yaml->AddOffer .....	522
\$yaml->Escape .....	522
\$yaml->GetYML .....	523
\$yaml->GetYMLFileName .....	523
\$yaml->Save .....	523
\$yaml->SetCompany .....	524
\$yaml->SetDate .....	524
\$yaml->SetShopName .....	524

\$yml->SetSiteUrl .....	524
<b>8.7 Таблицы и поля БД.....</b>	<b>525</b>
proshop_access_ip .....	525
proshop_access_ip_texts.....	525
proshop_admins_allowed_actions .....	526
proshop_admins_groups.....	526
proshop_admins_restricted_modules .....	527
proshop_admin_menus.....	527
proshop_adv_links .....	528
proshop_adv_links_texts.....	528
proshop_articles .....	529
proshop_articles_categories .....	529
proshop_articles_categories_texts.....	530
proshop_articles_products .....	530
proshop_articles_texts .....	531
proshop_banners .....	532
proshop_banners_clicks .....	533
proshop_banners_pages .....	533
proshop_banners_places.....	534
proshop_banners_shows .....	534
proshop_cache_data.....	535
proshop_cache_times .....	536
proshop_carts.....	536
proshop_categories.....	537
proshop_categories_params.....	539
proshop_categories_params_texts.....	539
proshop_categories_paths_texts .....	540
proshop_categories_texts .....	541
proshop_cblocks .....	541
proshop_cities .....	543
proshop_countries.....	543
proshop_cshippings .....	544
proshop_cshippings_texts.....	544
proshop_currencies.....	545
proshop_cthemes.....	546
proshop_cthemes_colors .....	546
proshop_discounts .....	547
proshop_discounts_cart .....	548
proshop_discounts_conf .....	548
proshop_discounts_texts .....	549

proshop_discounts_types .....	549
proshop_emspost_options .....	550
proshop_errors .....	550
proshop_fastmenu.....	551
proshop_fastmenu_texts.....	551
proshop_homepage_texts.....	552
proshop_labels .....	552
proshop_labels_texts .....	553
proshop_languages.....	553
proshop_languages_currencies .....	554
proshop_login_history .....	555
proshop_manufacturers .....	555
proshop_manufacturers_texts.....	556
proshop_meta_tags.....	557
proshop_meta_texts.....	557
proshop_meta_types.....	558
proshop_modules.....	559
proshop_modules_admin_actions .....	560
proshop_msgs.....	560
proshop_msgs_topics .....	561
proshop_news .....	562
proshop_news_edit_history .....	563
proshop_news_subscribers .....	564
proshop_news_texts .....	564
proshop_options.....	565
proshop_orders .....	566
proshop_orders_products .....	570
proshop_orders_statuses.....	572
proshop_orders_statuses_texts .....	572
proshop_orders_statuses_were .....	573
proshop_payments.....	574
proshop_payments_settings .....	575
proshop_payments_texts .....	575
proshop_payments_transactions .....	576
proshop_products.....	577
proshop_products_acquired.....	579
proshop_products_categories.....	580
proshop_products_details .....	581
proshop_products_details_texts .....	581
proshop_products_discuss .....	582

proshop_products_featured .....	583
proshop_products_files .....	584
proshop_products_files_keys.....	584
proshop_products_files_sessions .....	585
proshop_products_options.....	586
proshop_products_options_def_combs .....	586
proshop_products_options_exceptions.....	587
proshop_products_options_texts .....	587
proshop_products_options_types .....	588
proshop_products_options_vars .....	589
proshop_products_options_vars_combs .....	589
proshop_products_options_vars_texts .....	590
proshop_products_pictures.....	590
proshop_products_pictures_texts .....	591
proshop_products_pins.....	592
proshop_products_recommended .....	592
proshop_products_related .....	593
proshop_products_texts.....	593
proshop_products_variants.....	594
proshop_products_wished .....	595
proshop_reviews .....	595
proshop_rtshippings.....	596
proshop_rtshippings_groups.....	597
proshop_rtshippings_groups_settings .....	597
proshop_ruspost_options.....	598
proshop_ruspost_regions.....	599
proshop_settings.....	599
proshop_settings_groups.....	600
proshop_settings_types .....	601
proshop_slider.....	602
proshop_shippings .....	602
proshop_shippings_costs.....	603
proshop_shippings_payments .....	603
proshop_shippings_texts .....	604
proshop_shortcuts.....	605
proshop_smsgates.....	605
proshop_smsgates_settings .....	606
proshop_states.....	606
proshop_stats_cats_views .....	607
proshop_stats_search.....	608

proshop_tags_products.....	608
proshop_tags_texts .....	609
proshop_taxes.....	609
proshop_taxes_texts .....	610
proshop_url_name_history.....	611
proshop_users.....	611
proshop_users_fields .....	615
proshop_users_types .....	615
proshop_visitors .....	616
proshop_votes.....	617
proshop_votes_antiflood .....	617
proshop_votes_texts .....	618
proshop_votes_vars .....	619
proshop_votes_vars_texts .....	619
<b>8.8 Описание конфигурационных файлов .....</b>	<b>620</b>
8.8.1 includes/config.php .....	620
8.8.2 includes/defs.php .....	620

## О руководстве

Данное руководство предназначено исключительно для разработчиков, программистов и интеграторов дизайна, использующих CMS Fast-Sales Pro в качестве базовой системы управления контентом для создания интернет-магазина, и рекомендовано для прочтения с целью более быстрого освоения системы изнутри.

Руководство охватывает следующие вопросы:

- Архитектура CMS Fast-Sales Pro и принципы ее функционирования
- Создание дополнительных модулей
- Взаимодействие с базой данных
- Требования к именованию элементов кода и объектов файловой системы
- Требования к оформлению PHP-кода и кода шаблонов дизайна
- Смена/модификация дизайна и его интеграция
- Встраивание платежных систем
- Справочник глобальных переменных
- Справочник констант
- Справочник функций
- Справочник классов и их методов
- Справочник таблиц и полей базы данных
- Описание конфигурационных файлов

Данное руководство предназначено для системы, версия которой совпадает с версией, указанной на [титальной странице](#) данного документа.

При чтении руководства и кода CMS необходимо также учитывать, что ранее CMS была рассчитана на работу с поддержкой PHP версии 4, поэтому при разработке не учитывались особенности объектно-ориентированного подхода, поддержка которого отсутствует в PHP 4.

## 1. Архитектура системы и принципы ее функционирования

Система управления контентом Fast-Sales Pro (далее – просто CMS) предназначена для создания интернет-магазина и является готовым к применению программным продуктом. Набор модулей и функционала вполне достаточен для того, чтобы на базе данной системы организовать полноценный интернет-магазин, но данная система (как и множество других подобных систем) не является универсальным программным средством, поэтому в ряде случаев требуется ее доработка под требования заказчика.

CMS написана на языке программирования PHP и использует в качестве средства хранения данных СУБД MySQL. Требуемая для корректного функционирования CMS версия PHP – 5.\*.\*, версия MySQL – не ниже 4.1.2. В качестве web-сервера используется Apache, но при помощи небольших доработок может использоваться и любой другой web-сервер.

В данной CMS код PHP полностью отделен от дизайна с помощью шаблонизатора [Smarty](#) версии 2.6. В качестве JavaScript Framework используется библиотека [jQuery](#) версии 1.3.2.

В качестве дополнительных средств для реализации некоторых функций используются готовые библиотеки с открытым исходным кодом (такие, как FCKEditor, kcaptcha, PclZip, PHPMailer, TCPDF).

### 1.1 Общая архитектура

Схема общей архитектуры представлена на рис. 1.

Ядро CMS состоит из двух файлов – **index.php** и **setup.php**. Данные файлы находятся в корневой директории CMS, а также в директории **admin** – их аналоги для административной части магазина. Данные файлы закодированы и не подлежат какой-либо модификации. Модифицировать эти файлы может только автор CMS по своему усмотрению (например, в каком-либо из обновлений продукта).

Ядро подключает различные модули, а также использует классы для взаимодействия с базой данных или иной работы. В ядре формируются все глобальные переменные, подключаются константы (из файлов **includes/config.php** и **includes/defs.php**) и подключаются различные функции (находятся в файле **includes/functions.php**). Все



константы, глобальные переменные и функции могут быть использованы в коде любого модуля и/или класса.

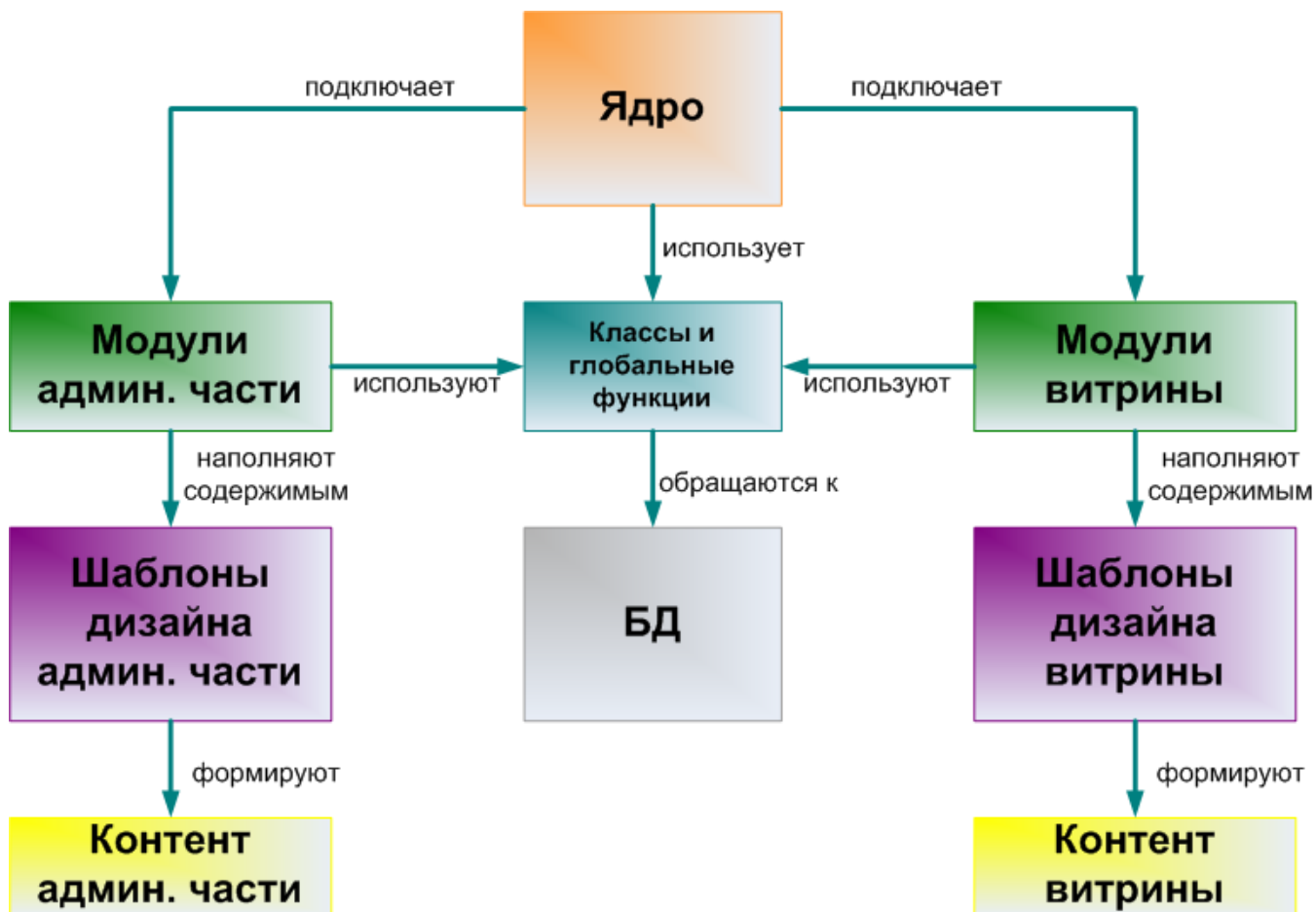


Рисунок 1. Схема общей архитектуры CMS Fast-Sales Pro.

Модули используют методы классов, вызывают функции (как имеющиеся в PHP, так и [заданные программистом](#) в файле `includes/functions.php`), читают/модифицируют значения [глобальных переменных](#) и используют [константы](#). При этом модули не обращаются к базе данных напрямую. Вместо этого они используют [классы](#), каждый из которых предназначен для решения определенных задач в рамках одного или нескольких модулей.

Каждый модуль формирует контентное содержимое сайта, наполняя тот или иной шаблон дизайна информацией. У каждого модуля имеется соответствующий шаблон (или набор шаблонов) дизайна.

## 1.2 Описание директорий и файлов

Корневая директория CMS содержит следующие поддиректории:

- [admin](#)

- [backups](#)
- [images](#)
- [includes](#)
- [logs](#)
- [modules](#)
- [payments](#)
- [shippings](#)
- [templates](#)
- [uploads](#)

Корневая директория CMS содержит следующие файлы:

- [.htaccess](#)
- [bug\\_report.php](#)
- [captcha.php](#)
- [check\\_requirements.php](#)
- [crc32.dat](#)
- [index.php](#)
- [messages.php](#)
- [mind\\_report.php](#)
- [robots.php](#)
- [robots.txt](#)
- [setup.php](#)
- [thumb.php](#)

### Директория **admin**

Содержит файлы и модули управления административной частью магазина (админкой). Состоит из следующих поддиректорий и файлов:

- Директория **modules** – содержит набор поддиректорий, каждая из которых относится к определенному модулю административной части и содержит PHP-файлы для работы с ним. Название директории соответствует названию модуля (например, директория **admin/modules/products** содержит PHP-файлы модуля продуктов).

- Файл **auth\_box.php** – отвечает за формирование и обработку формы входа в административную часть.
- Файл **editor.php** – отвечает за формирование HTML-редактора описаний (работает на основе FCKEditor) в административной части.
- Файл **index.php** – ядро административной части. Файл закодирован и не подлежит изменению.
- Файл **messages.php** – отвечает за формирование окна сообщения об успешности операции или ошибке, которое администратор видит после совершения какого-либо действия, связанного с изменением, сохранением, удалением информации и т.д.
- Файл **old\_browser.php** – отвечает за отображение Smarty-шаблона (templates/admin/old\_browser.html) с сообщением о том, что используемый администратором браузер устарел.
- Файл **setup.php** – часть ядра административной части. Файл закодирован и не подлежит изменению.
- Файл **tabs.php** – отвечает за сохранение состояния какой-либо вкладки (таба) любого модуля в сессии, чтобы при перезагрузке страницы выбор определенной вкладки сохранялся.

### Директория **backups**

Предназначена для хранения резервных копий данных магазина, созданных с помощью модуля управления резервными копиями **db\_backup**. Данная директория должна иметь права на запись (777), чтобы модуль **db\_backup** мог сохранять в ней архивы. В этой директории имеется файл **.htaccess**, предназначенный для запрета доступа к директории извне (чтобы любой пользователь сайта не смог скачать резервную копию, поскольку в ней может содержаться ценная или конфиденциальная информация).

Путь к данной директории хранится в константе [BACKUP\\_PATH](#).

### Директория **images**

Содержит изображения категорий, товаров и производителей. Данная директория и все ее поддиректории должны иметь права для записи (777). В директории **images** содержатся следующие поддиректории:

- **categories** – изображения категорий. Путь к данной директории хранится в константе [CATS\\_PICS](#).
- **manufacturers** – изображения производителей. Путь к данной директории хранится в константе [MANUF\\_PICS](#).
- **news** – превью-изображения новостей. Путь к данной директории хранится в константе [NEWS\\_PICS](#).
- **products** – изображения товаров. Путь к данной директории хранится в константе [NOWM\\_PRODS\\_PICS](#).
- **watermarks** – PNG-изображения водяных знаков, загруженные с помощью модуля **watermarks**, применяемые для наложения водяного знака на изображения товаров. Данные изображения разделяются на два файла: **img.png** (см. константу [WMARK\\_PNG](#)) и **thumb.png** (см. константу [WMARK\\_PNG\\_THUMB](#)) – для обычного и уменьшенного размеров изображения товара соответственно. Путь к данной директории хранится в константе [WMARKS\\_PICS](#).
- **watermarks/products** – изображения товаров, покрытые водяным знаком с помощью модуля **watermarks**. Путь к данной директории хранится в константе [WM\\_PRODS\\_PICS](#).
- **flags** – png-иконки флагов разных стран и территорий. Путь к данной директории хранится в константе [FLAGS\\_PICS](#).

Каждая из данных поддиректорий (кроме **flags**) содержит изображение **no\_picture.jpg**, которое выводится в том случае, если изображение определенного объекта (категории, производителя, продукта) не найдено. Имя файла данного изображения хранится в константе [NO\\_PICTURE](#).

Если в магазине применяется модуль наложения водяных знаков, на витрине магазина изображения товаров выводятся из директории **images/watermarks/products**, а в директории **images/products** при этом появляется файл `.htaccess` для защиты от скачивания оригиналов изображений (без водяного знака).

Директория **images/products** содержит как изображения продуктов обычного размера, так и уменьшенные варианты их изображений. Также данная директория содержит дополнительные варианты изображений каждого продукта (т.к. модуль продуктов позволяет добавлять несколько изображений одного и того же продукта).

## Директория `includes`

Содержит в себе следующие директории и файлы:

- Директория **classes** – PHP-файлы классов, имена которых строго соответствуют названиям классов, в них прописанных (например, класс `Products` содержится в файле `Products.class.php`). Кроме этого, в данной директории содержится директория **Shippings**, в которой находятся PHP-файлы классов, отвечающих за службы доставки (расчет стоимости доставки в реальном времени). Файл **.htaccess** предназначен для защиты от доступа к директории **includes/classes** извне.
- Директория **fonts** – содержит файлы шрифтов, применяемых где-либо в коде (например, файл **arial.ttf** хранит описание шрифта Arial и применяется в модуле **watermarks** при наложении текстового водяного знака на изображения товаров). Путь к данной директории хранится в константе [FONTS\\_PATH](#).
- Директория **libs** – содержит директории сторонних библиотек, которые используются магазином в некоторых модулях. При добавлении какой-либо библиотеки в функционал магазина ее директорию следует поместить именно в **includes/libs**. Путь к данной директории хранится в константе [LIBS\\_PATH](#).
- Файл [config.php](#) – [конфигурационный файл](#), который формирует некоторые [глобальные переменные](#) и [константы](#), используемые в магазине, загружает различные настройки магазина в сессию, включает механизм кэширования страниц, устанавливает необходимые значения некоторых изменяемых настроек `php.ini`. В **includes/config.php** хранится номер версии магазина. Кроме этого, в данном файле содержится очень важная глобальная переменная [\\$TABLES\\_RELS](#), которая содержит связи первичных ключей таблиц базы данных с вторичными (для [правильного удаления информации из базы данных](#) и сохранения ссылочной целостности), поскольку магазин использует MySQL с движком MyISAM, который не обеспечивает ссылочную целостность.
- Файл [defs.php](#) – [конфигурационный файл](#), хранящий в себе различные настройки, редко изменяемые программистом.
- Файл **functions.php** – [функции](#), определяемые программистом. В данный файл принято добавлять функции, которые не обращаются к базе данных

напрямую (используют [методы классов](#) вместо использования класса [DB](#)), и которые могут быть общими между административной частью магазина и витриной. Функции, добавляемые в данный файл, необходимо именовать с префиксом **fn\_** (в `includes/functions.php` можно встретить функции с префиксом **func\_**, что обусловлено историческим развитием CMS). Подробнее про требования к именованию функций читайте в [соответствующем разделе](#).

## Директория `logs`

Предназначена для хранения различных логов, генерируемых PHP-скриптами в ходе работы CMS. Данная директория должна быть доступна для записи (иметь права 777). В частности, в данной директории создается файл **errors.log**, в который записываются все ошибки, возникающие в ходе работы CMS (если константа [DEBUG](#) из файла `includes/defs.php` имеет значение **false**). Поэтому, если в каком-либо из модулей потребуется генерировать логи – их необходимо хранить в директории **logs**. В данной директории содержится файл **.htaccess** для защиты от чтения логов извне (логи можно прочитать только через FTP).

## Директория `modules`

Содержит набор поддиректорий, каждая из которых относится к определенному модулю витрины и содержит PHP-файлы для работы с ним. Название директории соответствует названию модуля (например, директория **modules/products** содержит PHP-файлы модуля продуктов).

## Директория `payments`

Содержит PHP-файлы, обеспечивающие обращение CMS к платежным системам. PHP-файлы имеют имена, соответствующие платежным системам, прописанным в таблице [proshop\\_payments](#) (например, файл **webmoney.method.php** соответствует платежной системе **WebMoney**). Файл **mail.php** вызывается любой платежной системой автоматически, когда в магазин поступает сигнал об оплате, и отвечает за смену статуса заказа на «Оплачен», за отправку письма-уведомления покупателю о том, что оплата за его заказ принята, и за начисление партнерских бонусов (для модуля партнерской программы). Подробнее о принципах работы платежных систем читайте в разделе про [встраивание платежных систем](#).

## Директория `shippings`

Содержит PHP-файлы, обеспечивающие работу онлайн-подсчета стоимости доставки заказа. Данные файлы имеют имена, соответствующие службам доставки, прописанным в таблицах [proshop\\_cshippings](#) и [proshop\\_rtshippings\\_groups](#) (например, файл `fedex.method.php` соответствует службе доставки **FedEx**).

## Директория `templates`

Содержит все шаблоны дизайна магазина. Состоит из следующих поддиректорий:

- **admin** – шаблоны дизайна административной части. Путь к данной директории хранится в константе [TPLS\\_PATH](#).
- **closed** – шаблон дизайна страницы закрытого на обслуживание магазина.
- **customer** – темы оформления дизайна витрины. Для каждой темы предназначена отдельная поддиректория. Путь к директории текущей темы оформления хранится в константе [TPLS\\_PATH](#).
- **mail\_texts** – шаблоны дизайна писем электронной почты.

Подробнее про структуру шаблонов и описание некоторых директорий и файлов смотрите в [соответствующем разделе](#).

## Директория `uploads`

Хранит различные файлы, загружаемые/скачиваемые из магазина. Данная директория (и все ее поддиректории) должна иметь права на запись (777). Состоит из следующих поддиректорий:

- **banners** – хранит файлы баннеров (HTML-файлы, изображения, flash-файлы). Имена файлов состоят из чисел и соответствуют идентификаторам баннеров, которые хранятся в таблице [proshop\\_banners](#). Путь к данной директории хранится в константе [BANNERS\\_PATH](#).
- **egoods** – хранит файлы электронных продуктов, которые скачиваются покупателями после покупки, либо бесплатно. Данная директория содержит поддиректории, имена которых соответствуют идентификаторам электронных продуктов (из таблицы [proshop\\_products](#)), а также файл `.htaccess` для блокировки доступа к файлам электронных продуктов извне. Путь к данной директории хранится в константе [EGOODS\\_PATH](#).

- **export** – хранит файлы форматов CSV, XML и т.п. для экспорта в такие системы, как Яндекс.Маркет, Google Base и т.д., созданные модулем экспорта. Путь к данной директории хранится в константе [EXPORT\\_PATH](#).
- **fcke** – директория для хранения загруженных через HTML-редактор файлов (изображений, флэш-анимаций и т.п.).
- **tmp** – директория временных файлов, которая может быть использована для хранения временных файлов, генерируемых в ходе работы модулей. Путь к данной директории хранится в константе [TMP\\_PATH](#).

### Файл **.htaccess**

Файл с настройками для сервера Apache. В данном файле прописаны настройки модуля modRewrite веб-сервера Apache, который позволяет использовать человеко-понятные URL на витрине магазина.

### Файл **bug\_report.php**

Скрипт для функции «Сообщить об ошибке», которая доступна только в демо-версии. Данный скрипт не применяется в работе Fast-Sales Pro – он применяется только при включенном демо-режиме на сайте <http://fast-sales.ru/>. При эксплуатации системы данный файл можно удалить, но он присутствует в дистрибутиве CMS, поэтому приведен здесь для справки.

### Файл **captcha.php**

Код в данном файле отвечает за смену изображения captcha на витрине, при нажатии на специальную кнопку перезагрузки изображения (или при нажатии на само изображение), и в форме входа в административную часть. Обновление изображения происходит AJAX-запросом без перезагрузки страницы, поэтому код генерирования captcha был вынесен в данный отдельный PHP-скрипт.

### Файл **check\_requirements.php**

Данный скрипт отвечает за проверку CMS на совместимость с сервером и включается в действие при каждой загрузке какой-либо страницы CMS, если константа [CHECK\\_REQUIREMENTS](#) установлена в значение **true**. Если проверка на совмести-



мость неуспешна, загрузка страницы останавливается и в окно браузера выводятся рекомендации по устранению несовместимостей.

### **Файл crc32.dat**

Данный файл содержит контрольные суммы всех файлов CMS, вычисленные по алгоритму CRC32, и предназначен для корректной работы модуля установки обновлений (для отслеживания изменений файлов текущей версии CMS программистами). Данный файл нельзя редактировать, переименовывать и удалять, иначе при установке обновлений на CMS будут проблемы.

### **Файл index.php**

Данный файл – ядро открытой части магазина (витрины). Файл закодирован и не подлежит изменению.

### **Файл messages.php**

Отвечает за формирование окна сообщения об успешности операции или ошибке, которое пользователь видит после совершения какого-либо действия, связанного с изменением, сохранением, удалением информации и т.д.

### **Файл mind\_report.php**

Скрипт для функции «Ваше мнение», которая доступна только в демо-версии. Данный скрипт не применяется в работе Fast-Sales Pro – он применяется только при включенном демо-режиме на сайте <http://fast-sales.ru/>. При эксплуатации системы данный файл можно удалить, но он присутствует в дистрибутиве CMS, поэтому приведен здесь для справки.

### **Файл robots.php**

Скрипт для автоматического генерирования файла [robots.txt](#), находящегося в корневой директории магазина. Данный скрипт должен вызываться из других скриптов магазина. Вызов извне не сработает в целях безопасности.

### **Файл robots.txt**

Текстовый файл для запрета индексации некоторых разделов магазина. Генерируется автоматически с помощью [robots.php](#).

## Файл `setup.php`

Данный файл – часть ядра открытой части магазина (витрины). Файл закодирован и не подлежит изменению.

## Файл `thumb.php`

Отвечает за генерирование уменьшенного изображения продукта «на лету». Применяется там, где необходимо уменьшить изображение продукта, уменьшенного варианта которого нет (например, при формировании фотогалереи продукта на его странице нужно показать уменьшенные варианты дополнительных изображений). Используется следующим образом (пример):

```

```

`src` здесь – имя файла изображения из директории **images/products** или **images/watermarks/products** (если включен модуль **watermarks**), которое нужно уменьшить до размеров, прописанных в настройках магазина и хранящихся в переменных [\\$CONFIG](#)['settings']['product\_thumb\_width']

и

[\\$CONFIG](#)['settings']['product\_thumb\_height'].

## 1.3 Работа модулей

В данном разделе описано, что из себя представляет модуль в Fast-Sales Pro, что необходимо, чтобы модуль функционировал, и принцип работы модуля. Данный раздел является введением в модули Fast-Sales Pro и не претендует на полноту информации по созданию модулей. Созданию модулей с подробным освещением большинства вопросов посвящена [глава 2](#).

Модуль в Fast-Sales Pro – это набор кода, выполняющего определенные действия в рамках какой-либо задачи. Каждый модуль в CMS имеет определенное имя, которое может состоять из английских букв нижнего регистра, цифр и знака подчеркивания (\_).

Каждый модуль может иметь свою директорию в **modules** (для витрины) и/или в **admin/modules** (для административной части). Директория модуля должна называться строго его именем. В директории модуля должен содержаться файл **[имя\_модуля].php**, который является главным файлом модуля и содержит его программный код. Например, модуль продуктов имеет главные файлы **admin/modules/products/products.php** и **mod-**

`ules/products/products.php` для административной и публичной частей (витрины) соответственно.

Также модуль имеет свои шаблоны дизайна, главный из которых располагается в директории `templates/admin/[имя_модуля]` или в директории `templates/customer/[тема_оформления]/[имя_модуля]` и имеет имя файла `[имя_модуля].html`. Главный шаблон модуля – это HTML-шаблон, содержащий код Smarty, и, возможно, подключающий другие шаблоны.

Чтобы магазин подключил модуль к работе, в большинстве случаев необходимо его прописать в таблице [proshop\\_modules](#), но это не обязательно для тех модулей (пример – модуль корзины **cart** на витрине), которые:

- не управляются через административную часть;
- должны быть всегда включены;
- обязательно имеют свои страницы на витрине;
- не вызываются автоматически при открытии любой страницы витрины или административной части.

Если поле **avail** таблицы [proshop\\_modules](#) имеет значение **Y**, модуль включен (если значение **N** - выключен). При создании нового модуля, значение поля **avail** следует ставить в **N**, а затем включать через управление модулями в админке, поскольку таким образом информация в кэше о составе модулей будет обновлена корректно.

Если поле **changeable** имеет значение **Y**, администратор может включать и отключать модуль через административный интерфейс (для неотключаемых модулей нужно ставить значение **N**).

Поле **admin\_autorun** отвечает за запуск PHP-файла модуля (`admin/modules/[имя_модуля]/[имя_модуля].php`) при каждой загрузке любой страницы административной части. Например, так работают модули **reviews** (отзывы о сайте) и **discussions** (отзывы о продуктах) – каждый раз проверяя, нет ли новых отзывов (если есть, ссылка на модуль в меню меняет свой цвет и показывает в скобках число неодобренных отзывов). Если поле **admin\_autorun** имеет значение **N**, модуль будет вызываться только при переходе на страницу данного модуля в административной части.

Поле **customer\_autorun** отвечает за запуск PHP-файла модуля (`modules/[имя_модуля]/[имя_модуля].php`) при каждой загрузке любой страницы витрины. Например, так работает модуль **news** (новости), каждый раз загружая новости (из базы данных, или из кэша) в боковой блок. Если поле **customer\_autorun** имеет значение

**N**, модуль будет вызываться только при переходе на страницу данного модуля на витрине.

Назначение остальных полей таблицы `proshop_modules` смотрите в [ее описании](#).

Модуль может не иметь своих отдельных страниц (и, соответственно, директорий в **admin/modules** и **modules**). Например, управление модулями **recommended\_products** и **related\_products** встроено в редактирование товара в административной части, а на витрине эти модули выводят свое содержимое на странице карточки товара.

Модуль может иметь свой информационный боковой блок на витрине (см. [пример создания модуля «Распродажа»](#)).

Для взаимодействия с базой данных или совершения других действий модуль должен иметь свой отдельный класс (в директории **includes/classes**), но может использовать и [другие классы CMS](#).

Модуль может использовать собственную таблицу или набор взаимосвязанных таблиц базы данных (при этом желательно, чтобы имена таблиц соответствовали имени модуля) для хранения информации, либо использовать другие [имеющиеся таблицы](#). О требованиях по именованию таблиц и полей базы данных читайте в [соответствующем разделе](#).

Модуль должен учитывать [многоязычность CMS](#).

Модуль должен учитывать [права](#) текущего администратора в админке, если он управляется через админку.

**Важное замечание!** В процессе создания любого модуля необходимо отключать кэширование с помощью константы [CACHE\\_ENABLED](#), поскольку при включенном кэшировании Smarty работает в режиме игнорирования изменений в шаблонах, используя их откомпилированные ранее варианты. Лучше всего очищать шаблоны через админку, используя модуль **Оптимизация** (путем перекомпилирования всех шаблонов дизайна), а не вручную.

Подробная информация о большинстве нюансов создания модулей и примеры их создания приведены далее в [соответствующем разделе](#).

## 2. Создание модуля

Пожалуйста, перед чтением данной главы прочитайте полностью [главу 1](#), если Вы еще этого не сделали.

### 2.1 Программный каркас модуля

Главный PHP-файл модуля содержит весь PHP-код модуля (за исключением кода взаимодействия с БД, который должен быть вынесен в класс). Не рекомендуется разбивать его на несколько файлов при помощи **require** и **include**, поскольку это отрицательно сказывается на быстродействии. Главный файл модуля состоит из нескольких частей:

- блок лицензионного соглашения (на английском языке);
- код проверки того, что модуль запускается ядром магазина, а не обращением к PHP-файлу извне, а также проверки того, что модуль включен администратором;
- блок функций;
- общий код, выполняющийся всегда (при любом действии модуля);
- код-развилка, выполняющий ту или иную функцию, определенную в блоке функций, в зависимости от действия модуля (модуль может выполнять разный код – в зависимости от того, что требуется пользователю);
- код сохранения сообщений модуля в сессии (для их вывода пользователю).

Таким образом, главный PHP-файл модуля в упрощенном виде выглядит так:

```
<?php
/* лицензионное соглашение */
// ...

/* код проверки (приведен пример для модуля админки) */
global $CONFIG;

if (!defined('_FAST_SALES_ADMIN_STARTED_') || $CONFIG['modules']['hello_world']['avail'] != 'Y') {
    die(basename(__FILE__).' : Access denied');
}

/* блок функций */
function hw_location_bar() {
    // ...
}

function hw_add() {
```

```
// ...
}

function hw_add_save() {
    # входные параметры
    $name = $_REQUEST['name'];
    $text = $_REQUEST['text'];
    # /входные параметры

    // ...
}

function hw_default() {
    // ...
}

/* общий код */
hw_location_bar();

/* код-развилка */
$act = isset($_REQUEST['act']) ? $_REQUEST['act'] : '';

if ($act == 'add') {
    hw_add();
} elseif ($act == 'add_save') {
    hw_add_save();
} else {
    hw_default();
}

/* код сохранения сообщений (это всегда вызов функции func_msgs_assign) */
func_msgs_assign($err, $msgs);
?>
```

Рассмотрим каждую из этих частей по отдельности.

## **Блок лицензионного соглашения**

Выглядит следующим образом (одинаков для любого PHP-файла CMS):

```
/*-----\
+-----+
| Fast-Sales Pro |
| Copyright (c) 2007-2010 Fsales development group <support@fast-sales.ru> |
| All rights reserved. |
+-----+
| PLEASE READ THE FULL TEXT OF SOFTWARE LICENSE AGREEMENT IN THE "COPYRIGHT" |
| FILE PROVIDED WITH THIS DISTRIBUTION. THE AGREEMENT TEXT IS ALSO AVAILABLE |
| AT THE FOLLOWING URL: http://fast-sales.ru/fsales_pro_license.html |
| |
| THIS AGREEMENT EXPRESSES THE TERMS AND CONDITIONS ON WHICH YOU MAY USE |
```

```
| THIS SOFTWARE PROGRAM AND ASSOCIATED DOCUMENTATION THAT Fsales |
| development group (hereinafter referred to as "THE AUTHOR") IS FURNISHING |
| OR MAKING AVAILABLE TO YOU WITH THIS AGREEMENT (COLLECTIVELY, THE |
| "SOFTWARE"). PLEASE REVIEW THE TERMS AND CONDITIONS OF THIS |
| LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE |
| SOFTWARE. BY INSTALLING, COPYING OR OTHERWISE USING THE |
| SOFTWARE, YOU AND YOUR COMPANY (COLLECTIVELY, "YOU") ARE ACCEPTING |
| AND AGREEING TO THE TERMS OF THIS LICENSE AGREEMENT. IF YOU ARE |
| NOT WILLING TO BE BOUND BY THIS AGREEMENT, DO NOT INSTALL OR USE THE |
| SOFTWARE. VARIOUS COPYRIGHTS AND OTHER INTELLECTUAL PROPERTY |
| RIGHTS PROTECT THE SOFTWARE. THIS AGREEMENT IS A LICENSE AGREEMENT |
| THAT GIVES YOU LIMITED RIGHTS TO USE THE SOFTWARE AND NOT AN |
| AGREEMENT FOR SALE OR FOR TRANSFER OF TITLE. THE AUTHOR RETAINS ALL RIGHTS |
| NOT EXPRESSLY GRANTED BY THIS AGREEMENT. |
|
| The Initial Developers of the Original Code are Fsales development group |
| Portions created by Fsales development group are Copyright (C) 2007-2009 |
| Fsales development group. All Rights Reserved. |
+-----+
\*****/
```

## **Код проверки**

Может быть разным (в зависимости от местоположения модуля – на витрине или в админке). Если модуль работает в админке, код проверки будет выглядеть следующим образом:

```
global $CONFIG;

if (!defined('_FAST_SALES_ADMIN_STARTED_') || $CONFIG['modules']['hello_world']['avail'] != 'Y') {
    die(basename(__FILE__).' : Access denied');
}
```

В данном коде проверяется на существование константа

[FAST\\_SALES\\_ADMIN\\_STARTED\\_](#), которая устанавливается в ядре, и проверяется, включен ли модуль – с помощью чтения настроек из глобальной переменной [\\$CONFIG](#).

Если модуль работает на витрине, код проверки будет выглядеть так:

```
global $CONFIG;

if (!defined('_FAST_SALES_STARTED_') || $CONFIG['modules'][' hello_world ']['avail'] != 'Y') {
    header('HTTP/1.1 404 Not Found');
    header('Status: 404 Not Found');
    die(basename(__FILE__).' : Access denied');
}
```

В приведенных примерах предполагается, что модуль имеет название **hello\_world**.

## **Блок функций**

Содержит определения всех функций (их код), используемых в модуле. Имя функции должно начинаться с одинакового префикса, по которому должно быть понятно, что функция принадлежит именно текущему модулю (например, для модуля **hello\_world** – префикс **hw\_**). После префикса в имени функции следует название действия, которое совпадает со значением переменной **act** (см. далее про код-развилку). Например, для действия **add** название функции в примере выше – **hw\_add**. Для действия **add\_save** – **hw\_add\_save**. Если функция не участвует в коде-развилке, ее название может быть любым, но с префиксом модуля (например, функция **hw\_location\_bar**).

Если в модуле должно содержаться не более 2-3 функций, код данных функций лучше вынести за пределы самих функций, а функции не создавать, т.к. это сэкономит время при работе синтаксического анализатора PHP (опытным путем было установлено, что чем больше функций объявляется в PHP-файле, тем медленнее работает сайт, – особенно это заметно на виртуальных хостингах).

### **Общий код**

В приведенном выше примере общий код состоит всего из одной строчки – вызова функции **hw\_location\_bar** (формирование строки навигации магазина). Общий код выполнится при любом действии модуля (независимо от значения переменной **act**). Если в общем коде используется какая-либо [глобальная переменная](#), перед использованием она обязательно должна быть объявлена с помощью ключевого слова **global**, иначе интерпретатор PHP будет считать ее значение пустым.

### **Код-развилка**

В данном коде происходит вызов функции модуля в зависимости от необходимого действия, название которого передается через переменную **act** (`$_REQUEST['act']`). Например, для вызова страницы добавления какого-либо элемента в модуле значение **act** будет равно **add**, а для сохранения информации, введенной на странице добавления – **add\_save**. Если переменная **act** содержит пустое значение, то нужно выполнить действие по умолчанию (например, считывание информации из базы данных и ее вывод через Smarty). Также необходимо заметить, что функции, участвующие в коде-развилке, не принимают никаких параметров (кроме тех случаев, когда это действительно необходимо). При совершении любого действия модуля вызывается функция без параметров,



в теле которой происходит считывание необходимых переменных, переданных GET- или POST-запросом (см. тело функции `hw_add_save` выше).

## **Код сохранения сообщений**

Практически всегда данный код состоит всего из одной строчки – вызов функции [func\\_msgs\\_assign](#) с передачей двух параметров – глобальных переменных `$err` и `$msgs`.

**Важное замечание!** Если модуль автозапускаемый, т.е. его запись в таблице [proshop\\_modules](#) имеет значение поля `admin_autorun` или поля `customer_autorun`, равное `Y`, перед выполнением кода-развилки (и, возможно, части общего кода, который при автозапуске выполняться не должен – например, формирование навигационной строки) необходимо проверять значение GET-переменной `module`. Если значение `$_REQUEST['module']` совпадает с именем модуля, то код-развилку выполнить можно. Иначе код выполнять нельзя, поскольку при пустом значении `act` может выполняться ненужный при автозапуске модуля код (для примера см. файл модуля `news` на витрине).

## **2.2 Состав функции модуля**

В данном разделе рассмотрено, из чего состоит функция модуля и какой код необходимо включать в ее тело при определенных условиях.

### **2.2.1 Типы функций модуля**

Любую функцию модуля можно отнести к одному из двух типов:

1. Функция, сохраняющая/изменяющая данные в БД.
2. Функция, выводящая информацию из БД.

Рассмотрим каждый тип по отдельности.

### **2.2.2 Сохраняющая/изменяющая данные в БД функция**

Тело такой функции состоит из следующих частей:

- a. блок перечисления используемых глобальных переменных;
- b. блок описания входных параметров;
- c. код действия;
- d. код перехода (возврата);

Рассмотрим каждую часть по отдельности.

## **Блок перечисления используемых глобальных переменных**

В данном блоке перечисляются все глобальные переменные, которые используются в теле функции. Например:

```
global $SHOP_LANGUAGE;  
global $LANGUAGES;  
global $LOCATION;  
global $CONFIG;  
global $LABEL;  
global $msgs;  
global $err;
```

## **Блок описания входных параметров**

Входные параметры – параметры, передаваемые методом GET или POST скрипту модуля при совершении определенного действия. Например:

```
# входные параметры  
$id      = $_REQUEST['id'];  
$caption = isset($_REQUEST['caption']) ? $_REQUEST['caption'] : '';  
$content = isset($_REQUEST['content']) ? $_REQUEST['content'] : '';  
# /входные параметры
```

Для параметров, которые не всегда могут быть заданы при совершении определенного действия, необходимо проверять существование с помощью **isset** и устанавливать пустое значение, если **isset** возвращает **false**. Это необходимо для того, чтобы PHP-интерпретатор не выдавал notice-предупреждение о том, что элемент массива **\$\_REQUEST** с заданным ключом не найден.

## **Код действия**

PHP-код, выполняющий действие функции. В ходе выполнения действия может возникнуть какая-либо ошибка (например, ошибка записи в БД или пользователь неправильно заполнил какое-то поле формы), которая должна быть записана в глобальную переменную-массив [\\$err](#). Если ошибок нет, нужно обязательно записать сообщение об успешности операции в глобальную переменную-массив [\\$msgs](#).

## **Код перехода (возврата)**

Обычно состоит из вызовов двух функций:

```
func_msgs_create($err, $msgs);  
header('Location: '); exit;
```

Функция [func\\_msgs\\_create](#) сохраняет в сессии сообщения, которые будут выведены пользователю. Функция `header` делает перенаправление на страницу, на которой должен оказаться пользователь после выполнения действия (обычно это та же самая страница, на которой он находился). После перенаправления пользователь увидит окно с сообщениями, созданными в ходе работы функции (в глобальных переменных-массивах [\\$err](#) и [\\$msgs](#)). **Очень важно** сразу после вызова функции `header('Location: ...')` сделать выход с помощью `exit`, поскольку было установлено, что в некоторых случаях сессии работают неверно, если сразу после перенаправления не завершить работу скрипта.

**Важное замечание!** Если пользователь не заполнил или неправильно заполнил какое-либо поле формы, после перенаправления на ту же самую страницу, на которой он совершил ошибку, необходимо отразить во всех полях формы данные, которые он вводил, чтобы не заставлять его вводить их еще раз при возникновении каждой ошибки. Для этого перед кодом перехода (возврата) нужно сохранить введенные данные в сессию, воспользовавшись классом [Request](#), а после возврата на страницу – восстановить их из сессии и отразить в форме. Как это сделать – смотрите для примера функции `news_edit_save` и `news_edit` в файле `admin/modules/news/news.php` модуля новостей (обратите внимание на использование GET-переменной `again`).

### 2.2.3 Выводящая данные из БД функция

Тело такой функции состоит из следующих частей:

- a. блок перечисления используемых глобальных переменных;
- b. блок описания входных параметров;
- c. код действия;
- d. код установки значений переменных Smarty.

Первые три части аналогичны [описанным выше](#), за исключением того, что блок описания входных параметров может отсутствовать (если функция не принимает никаких параметров через GET или POST).

Код установки значений переменных Smarty состоит из вызовов `assign`-методов объекта Smarty, например:

```
$SMARTY->assign('fact_info', $fact_info);  
$SMARTY->assign('action', 'edit');
```

Таким образом, мы устанавливаем переменные для главного шаблона модуля (`templates/admin/[имя_модуля]/[имя_модуля].html` или

`templates/customer/[тема_оформления]/[имя_модуля]/[имя_модуля].html`), который может подключать другие шаблоны (в зависимости от заданного действия).

## 2.3 Страницы модуля в админке

Чтобы вызвать главную страницу модуля в админке, нужно перейти по адресу `/admin/index.php?module=[имя_модуля]`. При этом CMS вызовет главный файл модуля, находящийся в директории `admin/modules/[имя_модуля]`, и имеющий имя `[имя_модуля].php`. Шаблон `templates/admin/index.html` при этом подключит главный шаблон модуля, находящийся в `templates/admin/[имя_модуля]`, и имеющий имя `[имя_модуля].html`.

Чтобы модуль совершил какое-либо действие, нужно вызвать его с параметром `act`, например: `/admin/index.php?module=news&act=add` – модуль новостей откроет страницу добавления новости. При этом в главный PHP-файл модуля будет передан параметр `act`, на значение которого должен отреагировать код-развилка, о котором говорилось [выше](#), и правильно подготовить информацию для вывода через главный HTML-шаблон модуля. Шаблон модуля, в свою очередь, должен отобразить подготовленную информацию, подключив дополнительный шаблон в зависимости от совершаемого действия, либо выполнив заложенный в него Smarty-код без дополнительного шаблона. Предпочтительнее вариант с подключением (деление шаблона на несколько более простых, отвечающих за определенные действия).

Например, в модуле `news` главный шаблон подключает вспомогательный шаблон `news_edit.html` (редактирование новости), если `act` равен `edit`, или подключает шаблон `news_add.html` (добавление новости), если `act` равен `add`.

## 2.4 Страницы модуля на витрине

Страницы модуля на витрине работают аналогичным образом, но имеют понятные для человека URL, правила для которых прописаны в файле `.htaccess` (используется `modRewrite`). Перед созданием модуля для витрины необходимо определиться с тем, какой вид будут иметь адреса страниц модуля (в зависимости от выполняемых действий).

Для формирования URL всех страниц на витрине используется класс [Links](#), или его объект в виде глобальной переменной `$LINKS`. Каждый модуль витрины, если он имеет какие-либо адреса страниц, должен иметь хотя бы один метод, определенный в классе

[Links](#), который возвращает адрес страницы того или иного действия модуля, или просто адрес главной страницы модуля. Для примера рекомендуем обратиться к модулю **news**.

Чтобы переход по адресу страницы модуля состоялся, необходимо прописать правило в файл [.htaccess](#), которое не пересекается с остальными правилами. Чтобы прописать правило, программист должен быть знаком с регулярными выражениями.

Рассмотрим пример правила:

```
RewriteRule ^([a-z][a-z])/news/([0-9]+)\.html$  
index.php?module=news&fact_id=$2&customer_lang_short=$1
```

Поскольку CMS поддерживает многоязычность контента, каждая страница CMS принадлежит определенному языку, поэтому адрес страницы всегда начинается с дву-буквенного обозначения языка страницы, которое необходимо передать в ядро с помощью параметра **customer\_lang\_short**. Например, для приведенного примера адрес страницы новости может быть таким: **http://example.com/ru/news/2.html**.

Для примера, приведенного выше, мы видим, что после определения языка страницы следует название модуля. После названия модуля может следовать все, что угодно для работы модуля. В приведенном примере за **news** следует идентификатор новости (с постфиксом **.html**), который берется из базы данных.

Когда пользователь набирает в адресной строке адрес страницы (или переходит по ссылке), этот адрес проверяется веб-сервером на соответствие какому-либо правилу из [.htaccess](#). Если веб-сервер находит соответствие, происходит соответствующий правилу переход – для приведенного выше примера это

**index.php?module=news&fact\_id=\$2&customer\_lang\_short=\$1**.

Таким образом, ядро вызывает модуль, указанный в GET-переменной **module**, т.е. выполняется файл **modules/[module]/[module].php**, а шаблон **templates/customer/[тема\_оформления]/index.html** подключает главный шаблон модуля (если он имеется) - **templates/customer/[тема\_оформления]/[module]/[module].html**.

Чтобы обратиться к какой-либо странице модуля с любого HTML-шаблона, необходимо использовать глобальную переменную [\\$LINKS](#). Например, формирование ссылки на главную страницу модуля **news** в любом HTML-шаблоне будет выглядеть следующим образом: **{ \$LINKS->News() }**.

## 2.5 Многоязычные особенности

CMS поддерживает формирование контента на разных языках, заданных администратором с помощью модуля **languages**. В связи с этим, функционал большинства модулей (тех модулей, где используются языкозависимые тексты) заточен под разные языки, как и интерфейс CMS (различные надписи в нем).

### 2.5.1 Текстовые метки

Все надписи магазина (небольшие тексты, такие, как надписи на кнопках, ссылках, и т.п.), кроме тех, которые хранятся в таблицах модулей, называются **метками**. Каждая метка имеет свое имя, которое может состоять из английских букв нижнего регистра, цифр и знака подчеркивания (см. [требования к именованию меток](#)). Каждая метка принадлежит к конкретной зоне (зона А - административная часть, зона С - витрина). Каждая метка имеет свой текст, принадлежащий конкретному языку. Например, если в магазине задано два языка, Английский и Русский, - метка будет содержать текстовые эквиваленты на данных языках, которые будут выводиться в интерфейсе магазина в зависимости от текущего выбранного языка пользователя.

Метки хранятся в таблице [proshop\\_labels](#). Их тексты для каждого языка – в таблице [proshop\\_labels\\_texts](#).

Чтобы использовать метку в PHP-коде, достаточно обратиться к глобальной переменной [\\$LABEL](#) следующим образом: `$LABEL['имя_метки']` – значение будет содержать текст метки для текущего языка, который задан в глобальной переменной [\\$SHOP\\_LANGUAGE](#). Массив [\\$LABEL](#) формируется в ядре при каждой загрузке страницы.

Чтобы использовать метку в Smarty-коде, достаточно обратиться к глобальной переменной [\\$LABEL](#) следующим образом: `{$LABEL.имя_метки}`.

**Важное замечание!** В PHP-файлах и HTML-шаблонах не должно быть никаких текстовых записей, «оставленных руками», т.е., например, если Вы хотите вывести сообщение об ошибке, то нужно сделать вот так:

```
$err[] = $LABEL['err...']; // присвоить массиву $err значение текстовой метки
```

Следующее решение будет неверным (даже если в магазине всего один язык):

```
$err[] = 'Произошла ошибка ...';
```

Неверным, поскольку если администратор добавит в магазин еще один язык (например, немецкий), на новом языке будет выводиться русское сообщение, что является ошибкой локализации.

То же самое справедливо по отношению к HTML-шаблонам (Smarty-коду).

Чтобы добавить/изменить метку, нужно воспользоваться модулем «Метки» в административной части магазина. Метка, кроме простого текста, может содержать HTML-код.

**Важное замечание!** Если метка существует только в зоне А (административной части), обращение к ней в зоне С (в HTML- или PHP-файле витрины) приведет к пустому результату, т.к. в зоне С ее нет. Если метка с одним и тем же именем требуется в обеих зонах, необходимо добавлять ее сразу в обе зоны. Деление на зоны обусловлено тем, что в обеих зонах используется разное количество меток (в зоне А меток больше, чем в С), поэтому в зоне С загружаются свои метки, а в зоне А – свои (чтобы экономить используемую память сервера). Например, метка **lbl\_news** содержится как в зоне А, так и в зоне С, а метка **lbl\_optimization** содержится только в зоне А.

### 2.5.2 Тексты, формируемые модулем

Любой модуль может хранить различные тексты на разных языках. Например, модуль **products** хранит описание каждого продукта на разных языках. Информация о продукте содержится в таблице [proshop\\_products](#), а текстовые составляющие (название, краткое описание, полное описание) хранятся в таблице [proshop\\_products\\_texts](#). Если в магазине задано два языка, то в таблице [proshop\\_products\\_texts](#) будут содержаться две записи (по одной на каждый язык), относящиеся к одному и тому же продукту.

По описанному принципу работают все модули CMS, которые имеют отношение к хранению текстовой информации, зависимой от языка.

Извлечение текста для текущего языка должно происходить в методе класса, который использует модуль. В данный метод нужно передавать идентификатор текущего языка (хранится в **\$SHOP\_LANGUAGE['id']**), на который следует опереться в SQL-запросе по сохранению/извлечению текстовой информации.

Например, метод [Articles::GetAll](#) принимает параметр **\$lang\_id** (или метод [Articles::Update](#)).

## 2.6 Распределение прав администраторов

В Fast-Sales Pro реализован механизм учета прав администраторов. В магазине всегда есть один-единственный суперадминистратор, который имеет неограниченные права. Он может создавать менее привилегированных администраторов и наделять их определенными правами на доступ к тому или иному модулю и/или его действию. Для этого суперадмину нужно создать группу администраторов, наделить ее правами, а затем привязать к этой группе нужных администраторов.

За контроль и распределение прав администраторов отвечают следующие таблицы БД:

- [proshop\\_admins\\_allowed\\_actions](#) – содержит список действий (из таблицы [proshop\\_modules\\_admin\\_actions](#)), которые разрешено совершать заданным группам администраторов;
- [proshop\\_admins\\_groups](#) – содержит список групп администраторов, созданных суперадмином;
- [proshop\\_admins\\_restricted\\_modules](#) – содержит список модулей, доступ к которым запрещен для заданных групп администраторов;
- [proshop\\_modules\\_admin\\_actions](#) – список действий в рамках заданного модуля, которые может ограничивать суперадмин.

Из всех перечисленных таблиц сейчас для нас представляет интерес только одна – [proshop\\_modules\\_admin\\_actions](#). Если создатель модуля желает, чтобы суперадмин мог ограничивать не только доступ к нему, но и доступ к его действиям, нужно в данной таблице создать по одной записи на каждое ограничение.

Например, для модуля **news** существуют ограничения на:

- сохранение/изменение/добавление информации;
- изменение мета-тэгов;
- рассылка новостей;
- управление подписчиками.

Т.е. одно или несколько перечисленных действий суперадмин может запретить для той или иной группы администраторов.

В таблице [proshop\\_modules\\_admin\\_actions](#) имеется поле **acts**, в котором через точку с запятой перечислены те значения REQUEST-переменной **act** модуля с именем, заданным в поле **module**, которые подпадают под ограничение. Ограничение имеет свою



текстовую метку (поле **label**), которая отображается при выборе ограничений суперадмином.

Например, для модуля **news** есть ограничение с меткой **lbl\_managing** (управление). Поле **acts** данного ограничения содержит **add;add\_save;edit\_save;save\_changes**. Т.е. если суперадмин установит флажок напротив данного ограничения, все перечисленные действия модуля **news** для определенной группы администраторов будут запрещены.

При работе модулей админки (не считая [вспомогательных AJAX-модулей](#)) ограничения контролируются ядром автоматически с помощью функции [fn\\_check\\_admin\\_rights](#). Т.е. программисту при создании модуля не нужно беспокоиться о контроле прав. Единственное, что нужно сделать, если требуется возможность ограничения прав на модуль, - прописать ограничения для модуля в таблицу [proshop\\_modules\\_admin\\_actions](#). По умолчанию, новые создаваемые модули являются общедоступными (т.к. изначально для них нет записей в таблице [proshop\\_admins\\_restricted\\_modules](#)), поэтому после создания модуля при необходимости суперадмин снимает галочку доступности с модуля для определенной группы администраторов.

Как уже говорилось немного выше, ограничения не контролируются автоматическим образом для вспомогательных AJAX-модулей, речь о которых пойдет далее, поэтому в таких модулях необходимо применять функцию [fn\\_check\\_admin\\_rights](#).

Функцию [fn\\_check\\_admin\\_rights](#), как и любую другую функцию PHP, можно применять не только в PHP-коде, но и в Smarty-коде в условном операторе **if**.

## 2.7 Вспомогательные AJAX-модули

Помимо основных модулей в CMS зачастую необходимо применять код, выполняющий какую-либо работу по вызову из JavaScript по технологии AJAX. Как уже говорилось в самом начале руководства, в качестве JavaScript framework используется библиотека [jQuery](#). Поэтому для AJAX-взаимодействий шаблонов дизайна с PHP-кодом используются методы данной библиотеки.

Рассмотрим следующий пример (из уже существующего кода CMS). В CMS реализован модуль «Бестселлеры» (**bestsellers**), код управления которым имеется в админке. По технологии AJAX в данном коде осуществляются следующие действия:

- загрузка списка бестселлеров из БД;
- редактирование списка;

- поиск продуктов для добавления в список;
- добавление продукта в список.

Таким образом, практически весь код управления модулем **bestsellers** реализован с применением технологии AJAX.

Код типичного вспомогательного PHP-модуля для взаимодействия через AJAX состоит из следующих частей:

- блок лицензионного соглашения;
- код подключения [setup.php](#);
- код проверки того, что модуль запускается ядром магазина, а не обращением к PHP-файлу извне, а также проверки того, что модуль включен администратором;
- блок функций;
- код-развилка, выполняющий ту или иную функцию, определенную в блоке функций, в зависимости от действия модуля (модуль может выполнять разный код – в зависимости от того, что передано в `$_REQUEST['act']`).

Для примера см. код в файле **admin/modules/bestsellers/bestsellers\_js.php**.

Для админки код подключения [setup.php](#) в AJAX-модуле будет выглядеть следующим образом:

```
define('ROOT_DIR', '../..../');
define('AJAXED_ROOT_DIR', '../..../');
require AJAXED_ROOT_DIR.'setup.php';
```

Если модуль выполняет только одно-единственное действие, блок функций и код-развилку можно опустить.

Содержимое функции AJAX-модуля состоит из тех же самых элементов кода, что и [содержимое функции обычного модуля](#), за исключением того, что функция в AJAX модуле всегда должна в конце своей работы выдать ответ на запрос в виде echo (при этом отсутствует код возврата с помощью header('Location:...'), т.к. в AJAX-модулях он не имеет смысла).

Например, рассмотрим функцию **bestsellers\_add** из файла **admin/modules/bestsellers/bestsellers\_js.php**. Данная функция отвечает за добавление продукта в число бестселлеров. Тело функции:

```
function bestsellers_add() {
    global $LABEL;
    global $msgs;
    global $err;
```

```
if (fn_check_admin_rights('bestsellers', 'add')) {
    # input parameters
    $prod_id = $_REQUEST['prod_id'];
    $sold_qty = $_REQUEST['sold_qty'];
    # /input parameters

    if (empty($sold_qty)) {
        $err[] = $LABEL['err_empty_sold_qty'];
    } else {
        if (!Bestsellers::SetSoldQty($prod_id, $sold_qty)) {
            $err[] = $LABEL['err_cant_save_info'];
        } else {
            $msgs[] = $LABEL['msg_info_saved'];
        }
    }
} else {
    $err[] = $LABEL['err_no_rights'];
}

echo func_to_json('', $msgs, $err);
}
```

Здесь мы видим блок определения глобальных переменных, блок входящих параметров, код выполнения действия и код возврата результатов (echo в конце).

Блок входящих параметров и код выполнения действия обернут в проверку доступности данного действия для модуля (с помощью функции [fn\\_check\\_admin\\_rights](#)) – если модуль или действие **add** модуля **bestsellers** запрещено суперадминистратором для текущего администратора, на выходе будет передано сообщение об ошибке доступа, которое содержится в метке **err\_no\_rights**. В данном случае ответа скрипта формируется в JSON-формат с помощью функции [func\\_to\\_json](#).

Ответ будет принят в JavaScript-функции `AddBestseller`, которая послала запрос на добавление бестселлера, и которая определена в файле **templates/admin/bestsellers/bestsellers\_js.html**:

```
function AddBestseller(prod_id) {
    var sold_qty = $('#prod_' + prod_id + '_sold_qty').attr('value');

    $.getJSON(BASE_URL + 'admin/modules/bestsellers/bestsellers_js.php',
        {act: 'add',
        prod_id: prod_id,
        sold_qty: sold_qty},
        function(obj) {
            if (!obj.err) {
                LoadBestsellers(0);
            }
            ProcessMsgs(obj.msg, obj.err);
        });
}
```

```
    }  
    );  
  
    return false;  
}
```

Здесь мы видим, что после того, как отправлен запрос, jQuery-метод `getJSON` ждет JSON-ответ от скрипта. Как только поступит ответ, функция проверит поле `err` (текст ошибок) на пустоту и вызовет [ProcessMsgs](#), которая выдаст принятое сообщение на экран.

Рассмотрим также код функции `bestsellers_list`, которая просто в качестве ответа выводит заполненный шаблон:

```
function bestsellers_list() {  
    global $SHOP_LANGUAGE;  
    global $SMARTY;  
  
    # входные параметры  
    $count = isset($_REQUEST['count']) ? $_REQUEST['count'] : 0;  
    # /входные параметры  
  
    $req = new Request('bestsellers', 'list');  
  
    if (!empty($count)) {  
        $req->SetParamValue('count', $count);  
    } else {  
        $count = $req->GetParamValue('count');  
  
        if (empty($count)) {  
            $count = 10;  
            $req->SetParamValue('count', $count);  
        }  
    }  
  
    // Определяем общее кол-во бестселлеров  
    $total_count = Bestsellers::GetCount(0);  
  
    // Вычисляем кол-во шагов для combo-box  
    $steps = intval(($total_count + 9) / 10);  
  
    // Извлекаем список первых $count бестселлеров  
    $bestsellers = Bestsellers::GetAll(0, $SHOP_LANGUAGE['id'], $count);  
  
    // Корректируем текущее число в combo-box в случае удаления элементов из списка  
    $diff = $count - $total_count;  
  
    if ($diff >= 10) {  
        $count -= intval($diff / 10) * 10;  
    }  
}
```

```
$SMARTY->assign('steps',      $steps);
$SMARTY->assign('show_count', $count);
$SMARTY->assign('bestsellers', $bestsellers);

echo $SMARTY->fetch(TPLS_PATH.'bestsellers/bestsellers_list.html');
}
```

В конце своей работы функция формируем Smarty-переменные и возвращает содержимое заполненного шаблона с помощью `echo`.

JavaScript-функция **LoadBestsellers**, которая послала запрос и ждет ответа, выглядит следующим образом:

```
function LoadBestsellers(count) {
    ShowLoading('bestsellers-list');
    $('#bestsellers-list').load(BASE_URL + 'admin/modules/bestsellers/bestsellers_js.php',
        {act: 'list', count: count});
    return false;
}
```

Т.е. функция вызывает jQuery-метод **load** и устанавливает содержимое `div` с `id="bestsellers-list"` в возвращенный скриптом HTML-текст.

Таким образом работают AJAX-модули в административной части.

На витрине AJAX-модули работают аналогичным образом, но у них код подключения [setup.php](#) несколько отличается (например, возьмем код из `modules/account/orders_js.php`):

```
define('ROOT_DIR', '../..');
require ROOT_DIR.'setup.php';
```

## 2.8 Написание тестового модуля «Hello, world!»

### 2.8.1 Постановка задачи

В качестве тренировочной практики напишем простой модуль `hello_world`. Определим выполняемые модулем задачи:

- модуль должен выводить любое текстовое сообщение в центральном блоке на главной странице магазина и это же сообщение в левом или правом боковом блоке на остальных страницах;
- в админке мы можем настраивать выводимое на витрине сообщение;
- сообщение может содержать в себе любой HTML-код и должно быть зависимым от языка.

## 2.8.2 Создание таблицы модуля в БД

Перед программированием модуля для админки и витрины необходимо создать класс модуля в директории **includes/classes** для взаимодействия модуля с БД, но перед этим важно определиться с таблицами, которые будет использовать модуль для хранения текстовой информации.

Согласно постановке задачи, модуль должен хранить и выводить текстовое сообщение, зависимое от языка. Поэтому, для его функционирования достаточно создать всего одну таблицу – **proshop\_hello\_world\_texts** (см. [правила именования таблиц](#)).

В данной таблице будем хранить текстовое сообщение модуля для каждого языка, имеющегося в магазине, поэтому таблица должна состоять из трех полей: **id** (идентификатор записи), **message** (текст сообщения) и **lang\_id** (идентификатор языка).

Код создания таблицы на языке MySQL выглядит следующим образом:

```
CREATE TABLE `proshop_hello_world_texts` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `message` VARCHAR(254) NOT NULL,  
  `lang_id` INT NOT NULL,  
  UNIQUE (`lang_id`)  
) ENGINE = MYISAM;
```

После создания таблицы необходимо прописать ее в коде формирования глобальных переменных [\\$TABLES](#) и [\\$TABLES\\_RELS](#) в файле **includes/config.php**:

```
$TABLES = array(  
  ...  
  'hello_world_texts',  
  ...  
);
```

В массиве [\\$TABLES\\_RELS](#) необходимо прописать привязанность поля **lang\_id** к первичному ключу таблицы [proshop\\_languages](#):

```
$TABLES_RELS = array(  
  ...  
  'languages' => array(  
    'relations' => array(  
      ...  
      'hello_world_texts' => array('lang_id' => 'd'),  
      ...  
    ),  
  ),  
  ...  
);
```

Это означает, что при удалении языка с идентификатором, который совпадает с `lang_id`, автоматически будет удалена и запись из таблицы `proshop_hello_world_texts`, у которой значение поля `lang_id` равно идентификатору удаляемого языка.

### 2.8.3 Создание класса модуля для взаимодействия с БД

Назовем класс модуля **HelloWorld**. Соответственно, имя файла класса будет следующим: **HelloWorld.class.php** (см. [требования к именованию объектов файловой системы](#)).

Согласно постановке задачи, модуль `hello_world` должен хранить и извлекать текстовое сообщение из БД. Поэтому, класс модуля будет содержать всего два метода – **Update** и **Get**. Приведем программный код класса:

```
<?php
/*****\
+-----+
| Fast-Sales Pro |
| Copyright (c) 2007-2010 Fsales development group <support@fast-sales.ru> |
| All rights reserved. |
+-----+
| PLEASE READ THE FULL TEXT OF SOFTWARE LICENSE AGREEMENT IN THE "COPYRIGHT" |
| FILE PROVIDED WITH THIS DISTRIBUTION. THE AGREEMENT TEXT IS ALSO AVAILABLE |
| AT THE FOLLOWING URL: http://fast-sales.ru/fsales_pro_license.html |
| |
| THIS AGREEMENT EXPRESSES THE TERMS AND CONDITIONS ON WHICH YOU MAY USE |
| THIS SOFTWARE PROGRAM AND ASSOCIATED DOCUMENTATION THAT Fsales |
| development group (hereinafter referred to as "THE AUTHOR") IS FURNISHING |
| OR MAKING AVAILABLE TO YOU WITH THIS AGREEMENT (COLLECTIVELY, THE |
| "SOFTWARE"). PLEASE REVIEW THE TERMS AND CONDITIONS OF THIS |
| LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE |
| SOFTWARE. BY INSTALLING, COPYING OR OTHERWISE USING THE |
| SOFTWARE, YOU AND YOUR COMPANY (COLLECTIVELY, "YOU") ARE ACCEPTING |
| AND AGREEING TO THE TERMS OF THIS LICENSE AGREEMENT. IF YOU ARE |
| NOT WILLING TO BE BOUND BY THIS AGREEMENT, DO NOT INSTALL OR USE THE |
| SOFTWARE. VARIOUS COPYRIGHTS AND OTHER INTELLECTUAL PROPERTY |
| RIGHTS PROTECT THE SOFTWARE. THIS AGREEMENT IS A LICENSE AGREEMENT |
| THAT GIVES YOU LIMITED RIGHTS TO USE THE SOFTWARE AND NOT AN |
| AGREEMENT FOR SALE OR FOR TRANSFER OF TITLE. THE AUTHOR RETAINS ALL RIGHTS |
| NOT EXPRESSLY GRANTED BY THIS AGREEMENT. |
| |
| The Initial Developers of the Original Code are Fsales development group |
| Portions created by Fsales development group are Copyright (C) 2007-2009 |
| Fsales development group. All Rights Reserved. |
+-----+
\*****/

if (!defined('_FAST_SALES_ADMIN_STARTED_') && !defined('_FAST_SALES_STARTED_')) {
```

```
header('Location: /'); exit;
}

class HelloWorld {
/**
 *
 * Извлечение текстового сообщение из БД
 *
 * @param int $lang_id - идентификатор языка
 *
 * Функция возвращает строку
 *
 */
function Get($lang_id) {
    global $TABLES;
    global $DB;

    $lang_id = Security::ToInt($lang_id);

    $sql = 'SELECT `message` FROM '.$TABLES['hello_world_texts'].' WHERE lang_id = '.$lang_id;
    $row = $DB->GetRow($sql);

    return Security::ToStr($row['message'], 'html');
}

/**
 *
 * Сохранение текстового сообщения в БД
 *
 * @param string $message - текст сообщения
 * @param int $lang_id - идентификатор языка
 *
 * Функция возвращает true в случае успеха и false - в случае ошибки
 *
 */
function Update($message, $lang_id) {
    global $TABLES;
    global $DB;

    $message = Security::ToSqlStr($message, 'html');
    $lang_id = Security::ToInt($lang_id);

    $sql = 'INSERT INTO '.$TABLES['hello_world_texts'].' (`message`, `lang_id`)' .
        ' VALUES ("'.$message.'", '.$lang_id.') ON DUPLICATE KEY
            UPDATE message = "'.$message.'";

    if (!$DB->Query($sql)) {
        return false;
    }
}
```



```
        return true;
    }
}
?>
```

Файл `includes/classes/HelloWorld.class.php` будет автоматически подключен с помощью `__autoload` при обращении к любому методу класса `HelloWorld`.

## 2.8.4 Программирование модуля в админке

Создадим PHP-файл модуля в админке. PHP-файл модуля должен располагаться в директории `admin/modules/hello_world` и иметь имя `hello_world.php`, т.е. полный путь к PHP-файлу модуля будет следующим: `admin/modules/hello_world/hello_world.php`.

Согласно постановке задачи, модуль должен хранить текстовое сообщение с возможностью его изменения, поэтому PHP-файл этого модуля в админке должен содержать код сохранения текстового сообщения и код его чтения из базы данных. В связи с этим, при заходе на страницу модуля в админке администратор должен видеть текстовое поле сообщения (вместе с самим сообщением, сохраненным ранее) и кнопку сохранения, нажатие на которую производит действие (**act**), которое мы назовем `save_changes`. При заходе на страницу модуля в админке, действие должно быть пустым, т.е. адрес этой страницы будет иметь вид: `/admin/index.php?module=hello_world`.

Приведем код PHP-файла `admin/modules/hello_world/hello_world.php`:

```
<?php
/*****\
+-----+
| Fast-Sales Pro |
| Copyright (c) 2007-2010 Fsales development group <support@fast-sales.ru> |
| All rights reserved. |
+-----+
| PLEASE READ THE FULL TEXT OF SOFTWARE LICENSE AGREEMENT IN THE "COPYRIGHT" |
| FILE PROVIDED WITH THIS DISTRIBUTION. THE AGREEMENT TEXT IS ALSO AVAILABLE |
| AT THE FOLLOWING URL: http://fast-sales.ru/fsales_pro_license.html |
|
| THIS AGREEMENT EXPRESSES THE TERMS AND CONDITIONS ON WHICH YOU MAY USE |
| THIS SOFTWARE PROGRAM AND ASSOCIATED DOCUMENTATION THAT Fsales |
| development group (hereinafter referred to as "THE AUTHOR") IS FURNISHING |
| OR MAKING AVAILABLE TO YOU WITH THIS AGREEMENT (COLLECTIVELY, THE |
| "SOFTWARE"). PLEASE REVIEW THE TERMS AND CONDITIONS OF THIS |
| LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE |
| SOFTWARE. BY INSTALLING, COPYING OR OTHERWISE USING THE |
| SOFTWARE, YOU AND YOUR COMPANY (COLLECTIVELY, "YOU") ARE ACCEPTING |
| AND AGREEING TO THE TERMS OF THIS LICENSE AGREEMENT. IF YOU ARE |
| NOT WILLING TO BE BOUND BY THIS AGREEMENT, DO NOT INSTALL OR USE THE |
```

```
| SOFTWARE. VARIOUS COPYRIGHTS AND OTHER INTELLECTUAL PROPERTY |
| RIGHTS PROTECT THE SOFTWARE. THIS AGREEMENT IS A LICENSE AGREEMENT |
| THAT GIVES YOU LIMITED RIGHTS TO USE THE SOFTWARE AND NOT AN |
| AGREEMENT FOR SALE OR FOR TRANSFER OF TITLE. THE AUTHOR RETAINS ALL RIGHTS |
| NOT EXPRESSLY GRANTED BY THIS AGREEMENT. |
|
| The Initial Developers of the Original Code are Fsales development group |
| Portions created by Fsales development group are Copyright (C) 2007-2009 |
| Fsales development group. All Rights Reserved. |
+-----+
\*****/

#
# Код проверки
#
global $CONFIG;

if (!defined('_FAST_SALES_ADMIN_STARTED_') ||
    $CONFIG['modules']['hello_world']['avail'] != 'Y') {
    die(basename(__FILE__).' : Access denied');
}

#
# Блок функций
#
function hw_location_bar() {
    global $LOCATION;
    global $LABEL;

    $LOCATION->AppendItem('index.php?module=hello_world', $LABEL['lbl_hello_world']);
}

function hw_save_changes() {
    global $SHOP_LANGUAGE;
    global $LABEL;
    global $msgs;
    global $err;

    # входные параметры
    $message = isset($_REQUEST['message']) ? $_REQUEST['message'] : '';
    # /входные параметры

    if (!HelloWorld::Update($message, $SHOP_LANGUAGE['id'])) {
        $err[] = $LABEL['err_cant_save_changes'];
    } else {
        $msgs[] = $LABEL['msg_info_changes_saved'];
    }

    func_msgs_create($err, $msgs);
    header('Location: '.$_REQUEST['referer']); exit;
}
```

```
}

function hw_show() {
    global $SHOP_LANGUAGE;
    global $SMARTY;

    $SMARTY->assign('message', HelloWorld::Get($SHOP_LANGUAGE['id']));
}

#
# Общий код (формирование панели навигации)
#
hw_location_bar();

#
# Код-развилка
#
$act = isset($_REQUEST['act']) ? $_REQUEST['act'] : '';

if ($act == 'save_changes') {
    hw_save_changes();
} else {
    hw_show();
}

#
# Код сохранения сообщений
#
func_msgs_assign($err, $msgs);
?>
```

В данном коде функция **hw\_save\_changes** относится к числу [функций, модифицирующих данные в БД](#), а функция **hw\_show** – к [функциям, извлекающим данные из БД](#).

Следует также обратить внимание на метку с именем **lbl\_hello\_world**, которая используется в функции **hw\_location\_bar** – создать данную метку нужно заранее с помощью модуля **Метки** из административной части магазина. Причем, данная метка пригодится и на витрине, поэтому при создании метки следует указать обе зоны размещения. Текст этой метки может быть таким: «Привет, Мир!», а английский эквивалент – «Hello, World!».

Теперь мы должны создать HTML-шаблон со Smarty-кодом для данного модуля. Главный HTML-файл модуля должен находиться по следующему пути:

**templates/admin/hello\_world/hello\_world.html.**

Приведем Smarty-код шаблона модуля:

```
{capture name=hw}
<div align="center">
```

```
<form name="hw_form" action="index.php?module=hello_world&act=save_changes" method="post">
  <input type="hidden" name="referer" value="{$_smarty.server.REQUEST_URI}" />

  <table cellpadding="0" cellspacing="5">
  <tr>
    <td class="textLeft">{$LABEL.lbl_text}</td>
    <td width="15"></td>
    <td class="textLeft">
      <input type="text" class="text" name="message"
        value="{$_message|replace:' ': '&quot;'}" maxlength="254" />
    </td>
  </tr>
  <tr>
    <td colspan="3" align="center">
      <br />
      {include file="_system_/buttons/button.html" title=$LABEL.link_save
        js="document.forms.hw_form.submit(); return false;" apply=true}
    </td>
  </tr>
  </table>
</form>
</div>
{/capture}

{include file="_system_/block.html" title=$LABEL.lbl_hello_world width="100%"
  content=$_smarty.capture.hw}
```

Здесь приведена простейшая форма, с помощью которой мы сохраняем текстовое сообщение в базу данных, передав POST-запрос PHP-файлу модуля.

При создании HTML-шаблона (как и при создании PHP-файла) следует иметь в виду, что все текстовые метки, упоминаемые в нем, должны существовать в базе данных магазина (в таблице [proshop\\_labels](#)). Если среди меток, используемых в шаблоне, есть новые, их необходимо создать через админку с помощью модуля **Метки**.

Чтобы увидеть наш модуль в действии в админке, необходимо прописать модуль в таблице [proshop\\_modules](#). Выполним для этого следующий SQL-запрос:

```
INSERT INTO `proshop_modules` (
  `admin_menu_id`,
  `pos`,
  `name`,
  `label`,
  `avail`,
  `changeable`,
  `admin_autorun`,
  `customer_autorun`,
  `icon`
) VALUES (
  3,
  9,
  'hello_world',
```

```
'lbl_hello_world',  
'N',  
'Y',  
'N',  
'Y',  
NULL  
);
```

Таким образом, служебным именем модуля будет **hello\_world**, текстовой меткой (используемой для показа в меню админки) – **lbl\_hello\_world**, модуль будет создан выключенным (почему не наоборот – см. ниже), его сможет включать/выключать администратор. Модуль не является автозапускаемым в админке, но является таковым на витрине. Иконка модуля для меню админки не задана, поэтому будет использована иконка по умолчанию. Значения полей **admin\_menu\_id** и **pos** говорят о том, что ссылка на страницу модуля в админке будет показана на 9-ой позиции в блоке меню, который имеет идентификатор 3 в таблице [proshop\\_admin\\_menus](#).

**Важное замечание!** Новый модуль необходимо всегда создавать по умолчанию выключенным (т.е. со значением поля **avail** таблицы [proshop\\_modules](#), равным **N**), т.к. код проверки модуля на включенность в PHP-файле модуля не сработает, когда данные о составе модулей не обновлены в кэше (состав модулей хранится в **\$CONFIG['modules']**). Поэтому после создания модуля его должен включить суперадминистратор через раздел **Модули** - таким образом информация о составе модулей в кэше будет обновлена.

После включения модуля мы можем видеть ссылку на него в блоке меню админки (рис. 2).

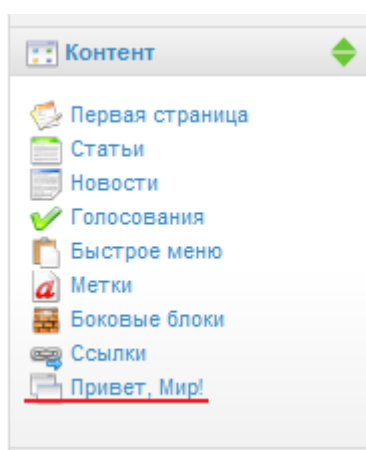


Рисунок 2. Ссылка на новый модуль в меню админки.

Проследовав по данной ссылке, мы перейдем на страницу **/admin/index.php?module=hello\_world** (рис. 3).

[Главная страница](#) :: Привет, Мир!

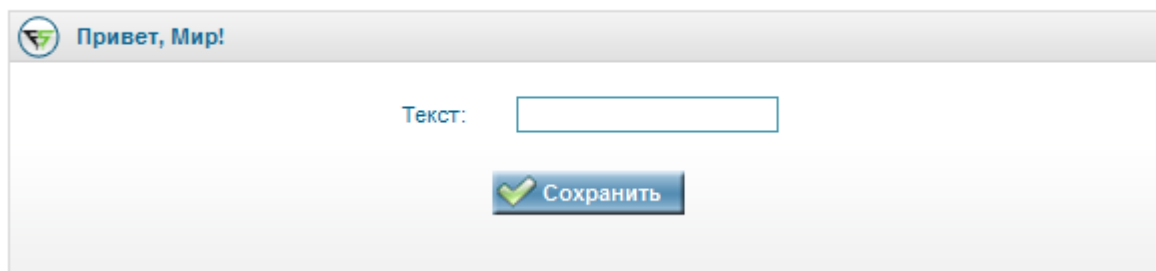


Рисунок 3. Страница модуля `hello_world` в админке.

Чтобы проверить, действительно ли сохраняется текст отдельно для каждого языка, необходимо ввести какой-либо текст (например, «Привет, Мир!») в текстовое поле, сохранить его, затем переключить язык на другой (например, английский) и ввести текст на этом языке (например, «Hello, World!»), сохранив его. Затем, переключая язык в админке, проверить, меняется ли текст сообщения в текстовом поле соответственно языку.

Таким образом, управление простым модулем `hello_world` через админку запрограммировано. Далее рассмотрим программирование модуля на витрине.

## 2.8.5 Программирование модуля на витрине

Создадим PHP-файл модуля на витрине. Путь к нему будет выглядеть следующим образом: `modules/hello_world/hello_world.php`.

У модуля не будет собственной отдельной страницы помимо бокового блока (слева или справа) и центрального блока на главной странице магазина, поэтому для этого модуля не нужно прописывать правила для адресов страниц в [.htaccess](#) и не нужно добавлять методы, формирующие адреса страниц модуля, в класс [Links](#).

Согласно поставленной задаче, PHP-файл модуля должен содержать в себе код для вывода текстового сообщения из БД. Причем, этот код должен быть извлечен из БД только в том случае, если пользователь находится на главной странице магазина, или если включен боковой блок модуля. Иначе, магазин при каждой загрузке страницы будет выполнять лишнюю работу и затормаживать работу всего сайта в целом, т.к. модуль `hello_world` является автозапускаемым на витрине (в таблице [proshop\\_modules](#) у данного модуля поле `customer_autorun` установлено в значение `Y`), что означает, что код его PHP-файла будет выполняться при загрузке любой страницы витрины (даже не относящейся к модулю).

Приведем программный код PHP-файла модуля:

```
global $CONFIG;
```

```
if (!defined('_FAST_SALES_STARTED_') || $CONFIG['modules']['hello_world']['avail'] != 'Y') {
    header('HTTP/1.1 404 Not Found');
    header('Status: 404 Not Found');
    die(basename(__FILE__).' : Access denied');
}

if (fn_cblock_enabled('hello_world_box') || empty($_REQUEST['module'])) {
    global $SHOP_LANGUAGE;
    global $SMARTY;

    $SMARTY->assign('hw_message_text', HelloWorld::Get($SHOP_LANGUAGE['id']));
}
```

Теперь нам необходимо модифицировать шаблон, отвечающий за формирование центрального блока главной страницы магазина. Файл данного шаблона находится в директории **templates/customer/[тема\_оформления]/homepage** и называется **homepage.html**. Открыв данный файл, мы обнаружим, что он формирует блок главной страницы и блок для модуля специальных предложений. Вставим Smarty-код в конец файла, чтобы блок модуля **hello\_world** выводился последним:

```
{if $CONFIG.modules.hello_world.avail eq 'Y' and $hw_message_text}
    {capture name=hw}
        <p>{$hw_message_text}</p>
    {/capture}

    {include file="_system_/center_block.html" title=$LABEL.lbl_hello_world
        content=$smarty.capture.hw}
{/if}
```

Далее нам необходимо создать шаблон, отвечающий за боковой блок. Шаблон будет располагаться в **templates/customer/[тема\_оформления]/hello\_world** и иметь имя **hello\_world\_box.html**. Код данного шаблона состоит всего из одной строчки:

```
<p>{$hw_message_text}</p>
```

Для бокового блока также необходимо создать HTML-файл в директории **templates/customer/[тема\_оформления]/\_system\_/blocks**. Назовем его **hello\_world\_box.html**. Содержимое данного файла будет следующим:

```
{if $CONFIG.modules.hello_world.avail eq 'Y' and $hw_message_text}
    {capture name=hw_box}
        {include file="hello_world/hello_world_box.html"}
    {/capture}

    {include file="_system_/side_block.html" title=$LABEL.lbl_hello_world
        content=$smarty.capture.hw_box}
{/if}
```

Таким образом, шаблон, находящийся в директории **templates/customer/[тема\_оформления]/\_system\_/blocks**, подключит другой шаблон,

который находится в директории **templates/customer/[тема\_оформления]/hello\_world**. Но, чтобы боковой блок был выведен магазином, необходимо сообщить CMS о том, что он существует. Для этого нужно создать в таблице [proshop\\_cblocks](#) запись о боковом блоке:

```
INSERT INTO `proshop_cblocks` (
  `name`,
  `label`,
  `column`,
  `column_fixed`,
  `weight`,
  `in_homepage`,
  `not_in_homepage`,
  `module`
) VALUES (
  'hello_world_box',
  'lbl_hello_world',
  'R',
  'N',
  '100',
  'N',
  'Y',
  'hello_world'
);
```

Данным запросом мы создали блок, именем файла которого в директории **templates/customer/[тема\_оформления]/\_system\_/blocks** является **hello\_world\_box.html**. Название блока хранится в метке **lbl\_hello\_world**. Данное название выводится только в модуле **Боковые блоки** в админке и предназначено для администратора (название самого блока на витрине задается меткой в шаблоне этого блока). Поле **column** со значением **R** говорит о том, что блок будет выведен в правой колонке. Поле **column\_fixed** со значением **N** говорит о том, что расположение блока (в левой или правой колонке) может менять администратор через модуль **Боковые блоки**. Поле **weight** отвечает за его расположение относительно других блоков той же колонки. Поле **in\_homepage** определяет, показывать ли блок на главной странице магазина (**Y** – показывать, **N** - нет). Поле **not\_in\_homepage** определяет, показывать ли блок на остальных страницах. Поле **module** отвечает за связь с модулем, который в нем прописан (в данном случае, блок принадлежит модулю **hello\_world**).

**Важное замечание!** После всех внесенных в CMS изменений необходимо воспользоваться модулем **Оптимизация** в админке, чтобы обновить кэш, иначе Вы не увидите внесенные изменения (также можно отключить кэширование, изменив значение константы [CACHE\\_ENABLED](#) из файла **includes/defs.php** на **false**).



Чтобы проверить работоспособность модуля, нужно зайти на главную страницу магазина – в ее центральной части должен располагаться блок с текстовым сообщением, заданным через админку (рис. 4).



**Рисунок 4.** Блок с текстовым сообщением модуля `hello_world` на витрине.

При этом на главной странице не должно быть бокового блока модуля `hello_world`. Если перейти на любую другую страницу, мы увидим боковой блок справа.

После отключения модуля через админку, все его блоки должны исчезнуть.

## **2.9 Создание модуля на примере модуля «Распродажа»**

*Раздел в процессе написания*

## 3. Взаимодействие с базой данных

Взаимодействие с базой данных в модулях Fast-Sales осуществляется путем вызова методов [классов](#) (в соответствии со [схемой работы](#)). Чтобы реализовать какое-либо взаимодействие с БД, необходимо воспользоваться существующим методом какого-либо класса или написать недостающий метод и/или класс.

### 3.1 Написание классов

Все классы CMS хранятся в директории **includes/classes**. Имя файла класса должно иметь следующий формат: **[имя\_класса].class.php**. Имя класса должно состоять из латинских букв нижнего и верхнего регистра (можно добавлять цифры) и начинаться с буквы верхнего регистра (например, Products.class.php).

При вызове статического метода класса либо при создании объекта класса файл класса будет подключен автоматически.

Для наибольшего быстродействия необходимо делать методы классов статическими (если для класса не требуется создавать объект).

PHP-файл класса должен состоять из следующих блоков (по порядку):

- блок лицензионного соглашения (на английском языке);
- код проверки того, что модуль запускается ядром магазина, а не обращением к PHP-файлу извне;
- блок функций (методов) класса.

Таким образом, PHP-файл класса в упрощенном виде выглядит так:

```
<?php
/* лицензионное соглашение */
// ...

/* код проверки */
if (!defined('_FAST_SALES_ADMIN_STARTED_') && !defined('_FAST_SALES_STARTED_')) {
    die(dirname(__FILE__) . ': Access denied');
}

/* блок функций */
function Method1() {
    // ...
}

function Method2() {
    // ...
}
```

?>

Рассмотрим каждую из этих частей по отдельности.

## **Блок лицензионного соглашения**

Блок совпадает с [указанным выше \(для файла модуля\)](#).

## **Код проверки**

```
if (!defined('_FAST_SALES_ADMIN_STARTED_') && !defined('_FAST_SALES_STARTED_')) {  
    die(basename(__FILE__).' : Access denied');  
}
```

В данном коде проверяются на существование константы

[FAST\\_SALES\\_ADMIN\\_STARTED](#) и [FAST\\_SALES\\_STARTED](#), которые устанавливаются в ядре.

## **Блок функций**

Название публичной (public) функции (метода) класса должно состоять только из латинских букв нижнего и верхнего регистра и начинаться с буквы верхнего регистра (например, GetAll).

Название приватной (private, protected) функции (метода) класса должно состоять только из латинских букв верхнего и нижнего регистра и начинаться со знака подчеркивания, после которого следует слово в нижнем регистре (например, \_getProductSqlFields).

## **3.2 Правильное удаление информации из БД**

Поскольку в CMS используется база данных MySQL с движком MyISAM, который не поддерживает ссылочную целостность, был придуман механизм для ручной поддержки ссылочной целостности.

Связи между таблицами БД отражены в глобальной переменной [\\$TABLES\\_RELS](#).

Чтобы удалить какую-либо сущность из БД с сохранением ссылочной целостности, необходимо вызвать глобальную функцию [fn\\_del\\_db\\_entity](#). Например, при удалении продукта мы вызываем функцию с соответствующими параметрами:

```
fn_del_db_entity('products', array($id));
```

При удалении сущности из БД в соответствующем методе класса необходимо вызвать именно функцию [fn\\_del\\_db\\_entity](#) вместо ручного удаления/обнуления связанных

записей. Исключениями могут быть случаи, при которых удалить сущности с помощью этой функции невозможно.

Если для каких-либо таблиц необходимо задать связи (первичных ключей со вторичными), нужно добавить эти связи в глобальную переменную [\\$TABLES\\_RELS](#). О том, как это сделать, написано в [описании](#) данной переменной.

## 4. Управление кэшированием

*Раздел в процессе написания*

## 5. Встраивание платежных систем

*Раздел в процессе написания*

## 6. Смена/модификация дизайна и его интеграция

В данном разделе описаны основные моменты по встраиванию/смене дизайна в CMS.

### 6.1 Введение

CMS Fast-Sales спроектирована таким образом, чтобы дизайн был полностью отделен от программной части. Это позволяет менять верстку имеющегося дизайна по своему усмотрению, а также встраивать любой другой дизайн, не прибегая к модификациям программного кода и не обладая знаниями в программировании (тем не менее, знать базовые основы программирования все же крайне желательно). Данная особенность достигается с помощью применения известного шаблонизатора [Smarty](#).

В корневой директории CMS имеется поддиректория [templates](#), отвечающая за хранение шаблонов дизайна витрины, административной части и шаблонов почтовых писем, отправляемых в качестве уведомлений пользователям магазина или администратору.

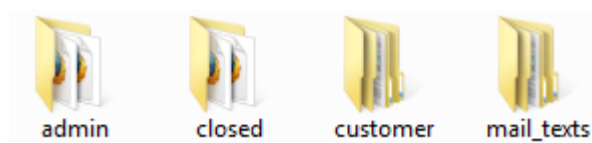


Рисунок 5. Поддиректории templates.

Директория [templates](#) содержит несколько вложенных поддиректорий (рис. 5):

- **admin** – директория для хранения шаблонов дизайна административной части;
- **customer** – директория для хранения шаблонов дизайна витрины магазина, которая также содержит поддиректории, отвечающие за хранение разных вариантов дизайна (Fast-Sales может содержать в себе несколько вариантов дизайна витрины, которые могут переключаться в настройках);
- **mail\_texts** – директория для хранения шаблонов почтовых писем на разных языках, поддерживаемых магазином;
- **closed** – директория для хранения шаблонов витрины, отвечающих за отображение специальной страницы, когда магазин закрыт на обслуживание.

Директории **admin**, **mail\_texts** и **closed** в данном контексте нам не интересны, поскольку дизайн админки и специальной страницы **closed** меняется очень редко, а про

тексты шаблонов почтовых уведомлений можно подробнее прочитать в [Руководстве пользователя Fast-Sales](#).

Директория **templates/customer** может содержать несколько поддиректорий дизайнов витрины, не зависящих друг от друга, и должна содержать в себе по крайней мере одну поддиректорию (у витрины должен быть хотя бы один скин), в которой структурировано расположены все файлы шаблонов дизайна, css-файлы и изображения. Например, на момент написания этой статьи, CMS Fast-Sales Pro содержала три встроенных дизайна (скина) витрины (рис. 6).

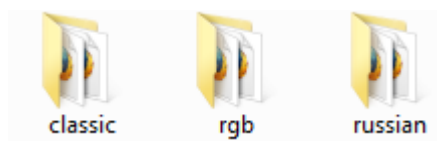


Рисунок 6. Поддиректории **templates/customer**.

Удалив одну из данных директорий, мы тем самым полностью удаляем один из вариантов дизайна витрины (вместе со всеми его цветовыми схемами).

Далее рассмотрим внутреннюю структуру директории скина витрины магазина на базе CMS Fast-Sales на примере дизайна **Classic** (директория этого скина - **templates/customer/classic**).

## **6.2 Структура директории скина витрины Fast-Sales**

Рассмотрим, какие поддиректории и файлы шаблонов входят в директорию скина (рис. 7).



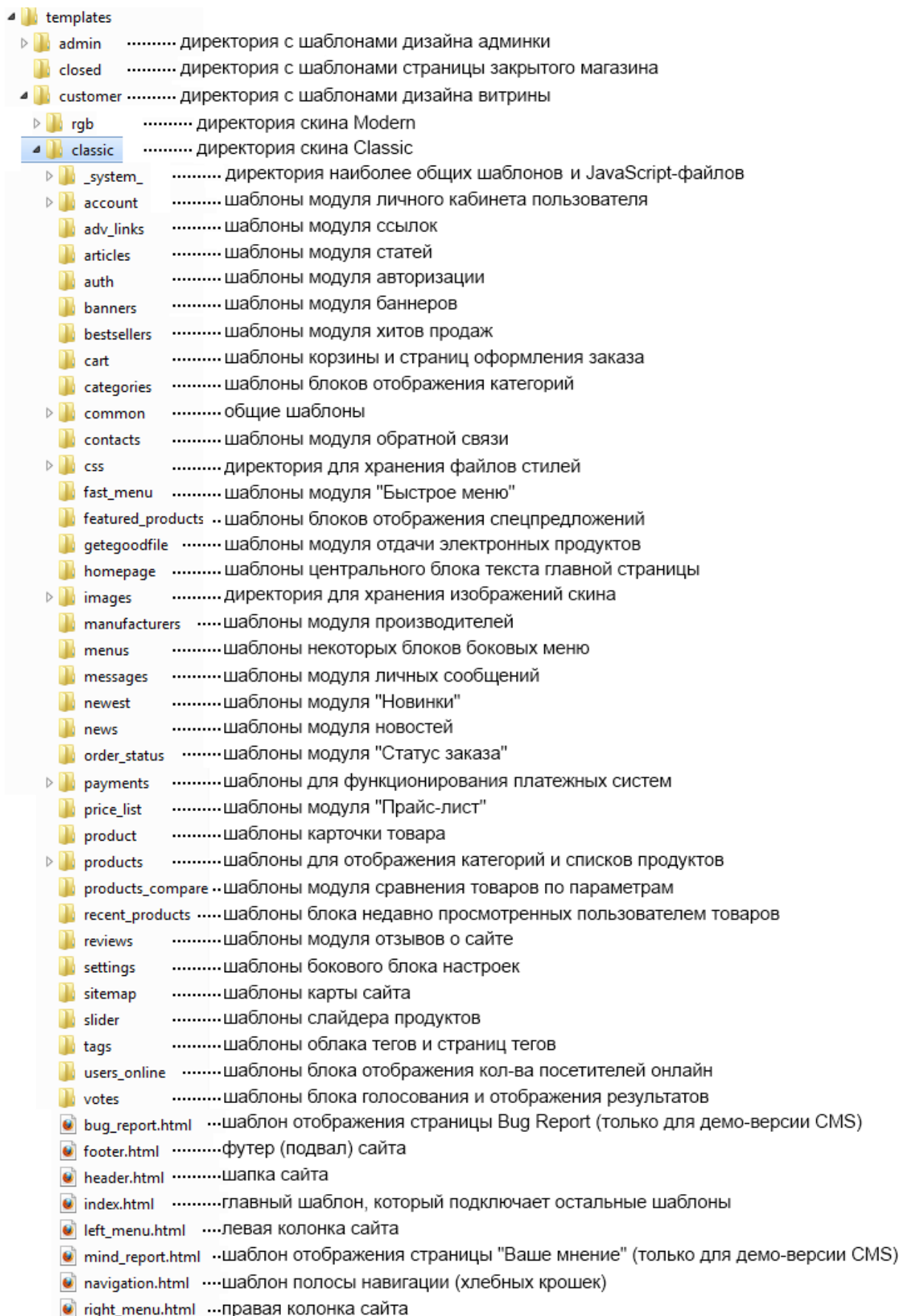


Рисунок 7. Структура директории скина витрины.

## 6.3 Структура дизайна витрины

Структуру дизайна витрины в общем виде можно представить, как показано на рисунке (рис. 8).

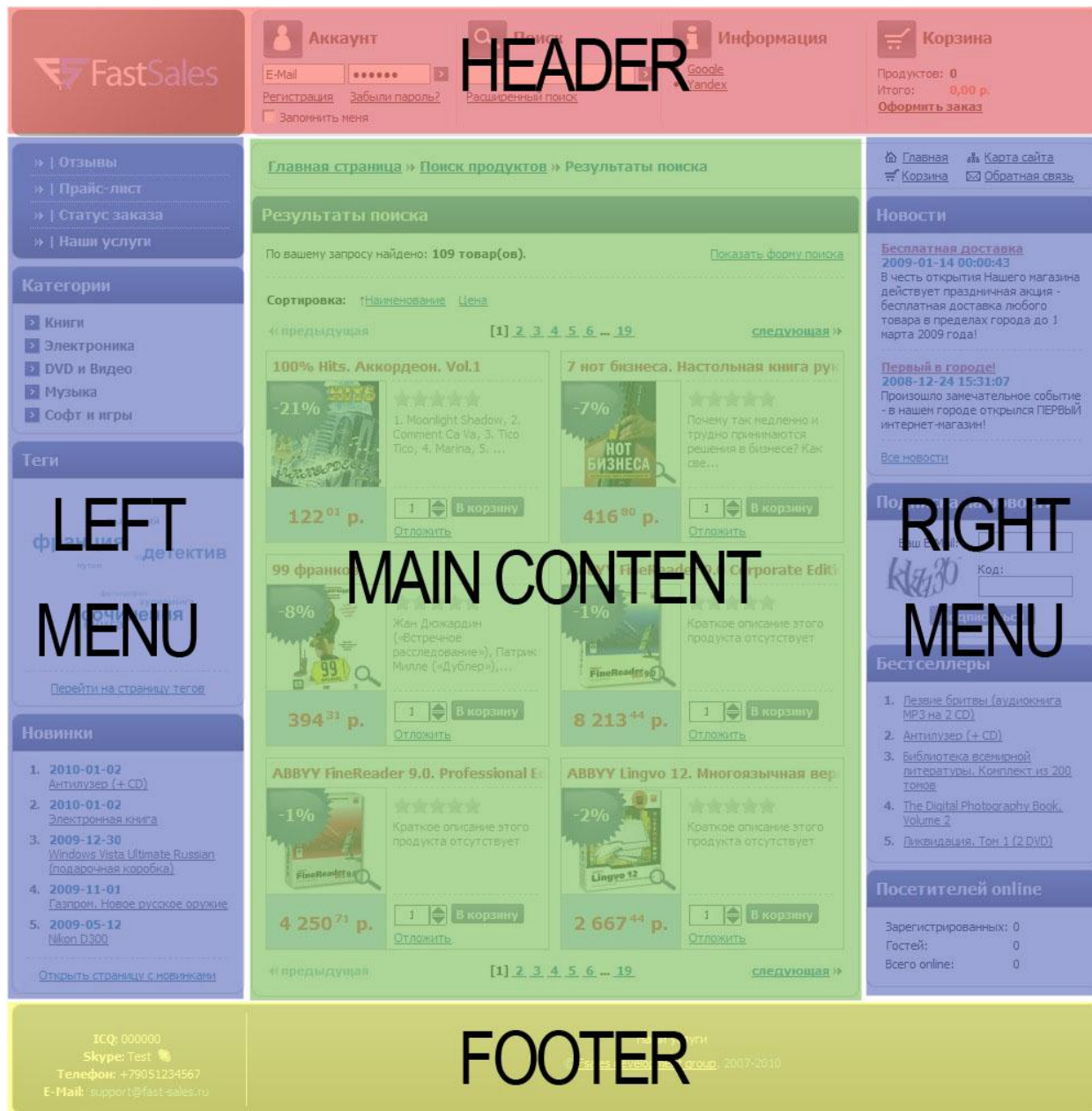


Рисунок 8. Структура дизайна витрины.

При отображении любой страницы витрины сайта обработка начинается с главного шаблона **index.html**. Этот шаблон должен включать остальные по мере необходимости:

- шапка сайта (header.html)
- левая колонка (left\_menu.html)
- правая колонка (right\_menu.html)

- подвал сайта (footer.html)

Шапка и подвал сайта в большинстве случаев всегда являются сквозными. Левая и/или правая колонки могут менять свое содержимое в зависимости от того, в каком модуле находится посетитель (это определяется настройками расположения боковых блоков в панели управления CMS).

Содержимое центральной части сайта определяется модулем, в котором в данный момент находится посетитель. При этом, для каждого модуля определена своя директория шаблонов, совпадающая с именем модуля (рис. 7). Например, для отображения текста главной страницы витрины используется модуль **homepage** – его главный шаблон хранится в директории **homepage** и называется **homepage.html**.

## **6.4 Принцип подключения шаблонов модулей Fast-Sales**

В Fast-Sales шаблоны модулей подключаются из **index.html** по следующему принципу: если посетитель просматривает определенный модуль, то имя текущего модуля хранится в глобальной переменной `$module`, поэтому, чтобы подключить шаблон определенного модуля, нужно выполнить в **index.html** следующую конструкцию (в той части верстки, которая отвечает за вывод центральной колонки):

```
{include file="`$module`/`$module`.html"}
```

Т.е. шаблоны модулей должны храниться в директориях по именам модулей, причем в директории шаблонов модуля обязательно должен быть главный шаблон, имя которого совпадает с именем модуля, например:

```
homepage/homepage.html
```

Остальные шаблоны модуля могут подключаться из его главного шаблона (с помощью конструкции `{include}`).

При подключении разных шаблонов с помощью конструкции `{include}` следует задавать их относительные пути (относительно корня директории скина). Например, если мы хотим подключить шаблон **\_system\_/social\_bookmarks.html** (социальные закладки) из шаблона **homepage/homepage.html**, нужно писать следующим образом:

```
{include file="_system_/social_bookmarks.html"}
```

## **6.5 Процесс добавления или смены дизайна витрины**

Во-первых, для смены дизайна у Вас уже должен быть сверстанный вариант дизайна, включающий в себя HTML-код, CSS-файлы и изображения.

Весь процесс добавления скина можно представить так, как показано на рисунке (рис. 9).

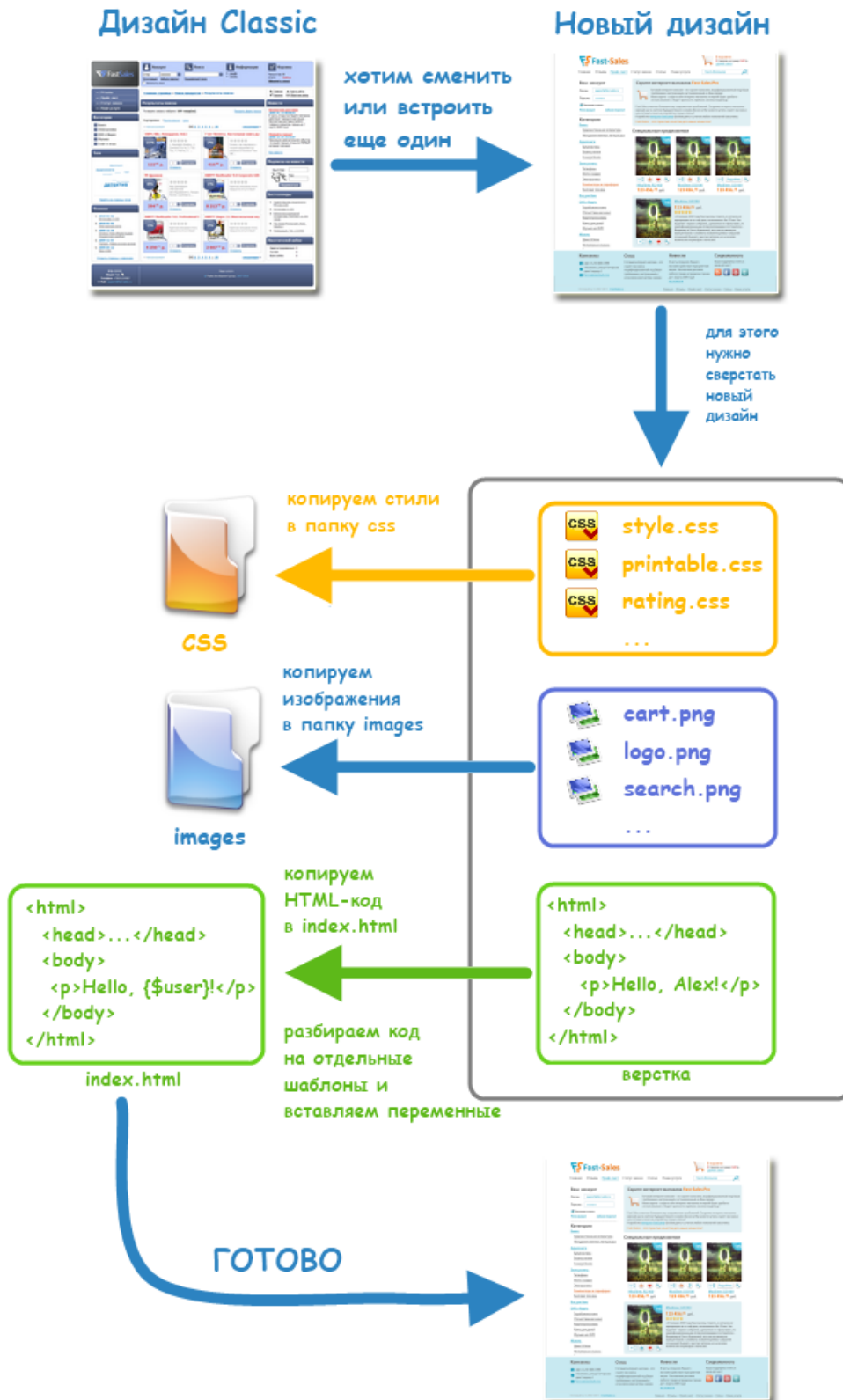


Рисунок 9. Процесс добавления/смены дизайна витрины Fast-Sales.

При этом желательно, чтобы макет дизайна соответствовал структуре, указанной на рис. 8. Макет может иметь как две боковые колонки, так и одну колонку (слева или справа), а может вообще не иметь колонок, но в интернет-магазинах так не принято.

Верстка, которую необходимо интегрировать в CMS, может быть сделана как на слоях, так и на таблицах, т.е. никаких специальных требований к верстке нет.

Чтобы новый дизайн отображался в настройках панели управления CMS (в разделе выбора скина витрины), нужно проделать следующие действия:

1. В базе данных магазина в таблицу [proshop\\_cthemes](#) добавить строчку, которая будет хранить служебное имя скина и его отображаемое название. Служебное имя скина – это название директории скина на витрине (например, для дизайна **Modern** – служебное имя **rgb**, а для дизайна **Classic** служебным именем является **classic**).
2. В таблицу [proshop\\_cthemes\\_colors](#) добавить определения цветовых схем, поддерживаемых данным скином (в этой таблице должна быть определена хотя бы одна цветовая схема).
3. В директории **templates/admin/images/customer\_themes** создать папку и назвать ее служебным именем скина, которое мы задали на шаге 1.
4. В папке, созданной на шаге 3, создаем графический файл **theme.png** с изображением скина размером **157x160** пикселей.
5. В папке, созданной на шаге 3, создаем папку **color\_schemes**.
6. В папке, созданной на шаге 5, для каждой цветовой схемы, определенной на шаге 2, создаем графический файл **[служебное\_имя\_цветовой\_схемы].png** с изображением скина в данной цветовой схеме. Размер изображения – **157x160** пикселей. Если скин имеет только одну цветовую схему, этот PNG-файл должен по изображению совпадать с файлом, созданным на шаге 4.
7. В таблице [proshop\\_cblocks](#) дублируем все строчки для нового скина (поле **ctheme** каждой продублированной строчки должно содержать служебное имя скина, заданное на шаге 1).

При интеграции нового скина нужно сохранять все наименования шаблонов, которые уже определены во встроенных скинах. Например, если в скине **Classic** имеется шаблон **templates/customer/classic/account/orders/order\_details.html**, то имя файла **order\_details.html** нужно оставить таким же (вместе с расширением html), т.к. использо-

вание этого файла может быть прописано в PHP-коде и его переименование вызовет некорректную работу некоторой части модуля.

В новом скине можно использовать библиотеку **jQuery**, которая вместе с плагинами доступна и находится в директории `includes/libs/jquery`. Для определения версии браузера можно использовать библиотеку **BrowserDetect** (`includes/libs/browserdetect/browserdetect.js`). Для борьбы с некорректным отображением прозрачных PNG-изображений в браузере Internet Explorer 6 можно использовать библиотеку **DD\_belatedPNG** (`includes/libs/belated/DD_belatedPNG_0.0.8a-min.js`).

### **6.6 Глобальные переменные CMS в шаблонах дизайна**

В шаблонах дизайна Fast-Sales можно использовать несколько глобальных переменных (`$error404`, `$module`, `$BASE_URL` и т.д.), которые доступны в любом шаблоне.

Полный перечень возможных переменных и PHP-констант, а также их описание, можно найти в [соответствующих разделах справочника](#).

### **6.7 Примеры кода и работы шаблонов Fast-Sales**

Примеры уже встроенных скинов витрины и их работающих шаблонов всегда можно увидеть в пробной версии CMS, скачав ее с [официального сайта Fast-Sales](#).

### **6.8 Нюансы и полезные советы**

Здесь собраны полезные советы и нюансы, которые нужно учитывать при встраивании нового дизайна Fast-Sales.

- Поскольку интерфейс Fast-Sales может быть многоязыковым, то в шаблонах вместо надписей используются текстовые метки. Например, вместо того, чтобы прямо в шаблоне написать «Добро пожаловать», нужно через панель управления создать метку с данным текстом и встроить ее вместо этого текста в шаблон следующим образом:

```
{ $LABEL.lbl_welcome }
```

где `lbl_welcome` – служебное название метки.

Подробнее про работу с метками можно прочитать в [Руководстве пользователя Fast-Sales](#).

- Если все же необходимо вставить голый текст в шаблон не с помощью метки, то шаблоны следует сохранять в кодировке UTF-8 (без сигнатуры).

- В главном шаблоне index.html между тегами <head></head> должны быть прописаны по крайней мере следующие теги:
  - `<base href="{BASE_URL}" />`
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />`
- При подключении шаблона с помощью Smarty-конструкции {include} путь к подключаемому шаблону должен содержать прямые слэши вместо обратных:
  - `{include file="homepage\homepage"}` – неправильно
  - `{include file="homepage/homepage"}` – правильно
- При вставке каких-либо значений в поля input необходимо заменять во вставляемых значениях двойные кавычки на их HTML-эквиваленты следующим образом (а значения атрибутов всех HTML-тегов обрамлять в двойные кавычки):  
`<input type="text" name="name" value="{name|replace:'":'&quot;}'" />`

## 7. Требования

В данном разделе описаны требования по оформлению кода, именованию различных элементов кода, а также объектов файловой системы, и прочие требования.

### 7.1 Требования к именованию элементов кода

#### 7.1.1 Имена глобальных переменных

Имена всех глобальных переменных, создаваемых в коде CMS, должны быть в ВЕРХНЕМ\_РЕГИСТРЕ (разрешается соединять различные слова знаком подчеркивания).

Имя индекса в любом массиве, определенном в `$_SESSION`, может состоять из букв нижнего регистра.

#### 7.1.2 Имена локальных переменных

Имена всех локальных переменных в коде CMS должны быть в нижнем регистре. Если имя переменной состоит из двух или более слов, их нужно соединять знаком подчеркивания: `$test_variable`.

#### 7.1.3 Имена констант

Имена всех констант, определяемых в коде CMS, должны быть в ВЕРХНЕМ\_РЕГИСТРЕ (разрешается соединять различные слова знаком подчеркивания). Имена констант могут начинаться со знака подчеркивания.

#### 7.1.4 Имена функций

Имена всех функций, определяемых в модулях магазина, а также в файле `includes/functions.php`, должны состоять из букв нижнего регистра. Слова в имени функции следует разделять знаком подчеркивания.

Функции, определяемые в файле `includes/functions.php`, должны иметь имена, начинающиеся с префикса `fn_`.

Функции, определяемые в пределах какого-либо модуля, должны иметь любой одинаковый префикс, по которому можно определить, что функция относится именно к определенному модулю.



Имена JavaScript-функций должны состоять из букв верхнего и нижнего регистра. Каждое слово в имени JS-функции должно начинаться с заглавной буквы. Если имя функции состоит из более чем одного слова, слова при этом не должны разделяться какими-либо знаками. Пример: MyJavascriptFunc.

## 7.1.5 Имена классов и их методов

Имя класса в CMS должно состоять из букв нижнего и верхнего регистра и может содержать цифры. Имя класса должно начинаться с заглавной буквы. Если имя класса состоит из нескольких слов, слова не должны разделяться друг от друга никакими символами. При этом каждое слово должно начинаться с заглавной буквы. Например: WishedProducts.

Имя метода внутри CMS должно состоять только из букв разного регистра и цифр (в некоторых случаях содержит и знак подчеркивания в начале). При этом если метод используется только внутри класса (только другими методами этого класса), его имя должно начинаться со знака подчеркивания и следующей буквы нижнего регистра. Пример: `_getShippingInfo`. В остальных случаях – с буквы верхнего регистра. Если имя метода содержит несколько слов, они не должны разделяться друг от друга никакими символами. Например: `SetFilename`.

## 7.1.6 Имена таблиц и полей БД

Все таблицы БД CMS должны иметь одинаковый префикс (по умолчанию – **proshop\_**). Имена таблиц должны состоять из букв нижнего регистра. Если имя таблицы состоит из нескольких слов, их нужно разделять знаком подчеркивания.

Если таблица принадлежит определенной группе таблиц (например, продукты), ее имя должно начинаться с названия данной группы, чтобы при сортировке всех таблиц по имени можно было видеть все таблицы одной группы рядом друг с другом (как в случае с просмотром списка таблиц CMS в phpMyAdmin). Например:

- `proshop_products`
- `proshop_products_categories`
- `proshop_products_details`
- ...

Если таблица предназначена для хранения языкозависимых текстов, ее имя должно оканчиваться на **\_texts** (сама таблица при этом должна обязательно иметь поле **lang\_id** типа **INT**).

Имена полей таблицы должны состоять из букв нижнего регистра (разрешается применять знак подчеркивания). Если имя поля состоит из нескольких слов, они должны разделяться знаком подчеркивания. Имя поля должно начинаться с буквы.

### 7.1.7 Имена текстовых меток интерфейса

Все текстовые метки интерфейса CMS (которые хранятся в таблице [proshop\\_labels](#)) должны иметь имена, состоящие из букв нижнего регистра и знака подчеркивания.

При этом имя метки должно иметь следующий префикс:

- **msg\_** - если метка содержит текст какого-либо сообщения об успешности операции;
- **err\_** - если метка содержит текст какого-либо сообщения об ошибке;
- **note\_** - если метка содержит текст какого-либо примечания, вставляемого возле какого-либо HTML-элемента;
- **tip\_** - если метка содержит текст всплывающей подсказки;
- **link\_** - если метка содержит какой-либо глагол;
- **lbl\_** - в остальных случаях.

## 7.2 Требования к именованию объектов файловой системы

Имена всех файлов и директорий CMS, за исключением имен файлов, расположенных в директориях **includes/classes** и **includes/libs**, должны состоять из букв нижнего регистра и знака подчеркивания.

Система Unix чувствительна к регистру имен файлов и директорий.

Файлы классов, расположенные в директории **includes/classes**, должны иметь имена в следующем формате: **ИмяКласса.class.php**.

Файлы и директории сторонних библиотек в директории **includes/libs** могут иметь любые имена.

Файлы Smarty-шаблонов в директории [templates](#) должны иметь расширение **.html**.

## 7.3 Требования к оформлению PHP-кода и кода шаблонов дизайна

Здесь описаны требования по оформлению кодов, которых желательно придерживаться при написании нового функционала CMS, чтобы выдерживать общий стиль.

### 7.3.1 PHP-код

Оформление PHP-кода базируется на соглашениях по программированию кода Java ([Code Conventions for the Java Programming Language](#)), но не полностью следует им.

Для примера того, как следует оформлять код в CMS, приведем код функции:

```
function product_discussion($prod_id, $from) {
    global $SHOP_LANGUAGE;
    global $CONFIG;
    global $SMARTY;
    global $LABEL;
    global $LINKS;
    global $msgs;
    global $err;

    # входящие параметры
    $author   = isset($_REQUEST['author'])   ? $_REQUEST['author']   : '';
    $email    = isset($_REQUEST['user_email']) ? $_REQUEST['user_email'] : '';
    $rate     = isset($_REQUEST['rate'])     ? $_REQUEST['rate']     : '';
    $comment  = isset($_REQUEST['comment'])  ? $_REQUEST['comment']  : '';
    $keysting = isset($_REQUEST['keysting']) ? $_REQUEST['keysting'] : '';
    # /входящие параметры

    $prod_discuss_req = new Request('c_product_discuss', 'add', $prod_id);

    if (empty($_REQUEST['again'])) {
        $prod_discuss_req->DelParamValue('discuss_author');
        $prod_discuss_req->DelParamValue('discuss_email');
        $prod_discuss_req->DelParamValue('discuss_rate');
        $prod_discuss_req->DelParamValue('discuss_comment');
    }

    $SMARTY->assign('discuss_author', $prod_discuss_req->GetParamValue('discuss_author'));
    $SMARTY->assign('discuss_email', $prod_discuss_req->GetParamValue('discuss_email'));
    $SMARTY->assign('discuss_rate', $prod_discuss_req->GetParamValue('discuss_rate'));
    $SMARTY->assign('discuss_comment', $prod_discuss_req->GetParamValue('discuss_comment'));

    // Добавление отзыва
    if (isset($_REQUEST['act']) && 'add_discuss' == $_REQUEST['act']) {
        if (empty($author)) {
```

```
$err[] = $LABEL['err_empty_value'].': '.$LABEL['lbl_author'];
}

if (!empty($email) && !func_check_email($email)) {
    $err[] = $LABEL['err_value_incorrect'].': '.$LABEL['lbl_your_email'];
}

if (empty($rate)) {
    $err[] = $LABEL['err_empty_value'].': '.$LABEL['lbl_select_rating'];
}

if (func_count_array($err) == 0) {
    if (!isset($_SESSION['captcha_keystring'])
        || $_SESSION['captcha_keystring'] != $keystring) {
        $err[] = $LABEL['err_picture_text_incorrect'];
    } else {
        if (ProdsDiscuss::Add($prod_id, $_SESSION['visitor_id'],
            isset($_SESSION['customer_info']['id']) ?
                $_SESSION['customer_info']['id'] : 0,
            $author, $email, $rate, $comment, $SHOP_LANGUAGE['id'])) {
            if (mb_strlen(trim($comment)) == 0) {
                $msgs[] = $LABEL['msg_review_added2'];
            } else {
                $msgs[] = $LABEL['msg_review_added'];

                $info = array(
                    'short_lang' => $SHOP_LANGUAGE['short'],
                    'from'       => $CONFIG['settings']['other_notify_email'],
                    'from_name'  => $CONFIG['settings']['shop_name'],
                    'to'         => $CONFIG['settings']['other_notify_email'],
                    'SHOP_NAME'  => $CONFIG['settings']['shop_name'],
                    'PRODUCT_CAPTION' => Products::IdToCaption($prod_id, $SHOP_LANGUAGE['id']),
                    'PRODUCT_LINK'   => $LINKS->Product($prod_id, false, $from),
                    'AUTHOR_NAME'    => $author,
                    'AUTHOR_EMAIL'   => !empty($email) ? $email : '-',
                    'AUTHOR_RATE'    => $rate,
                    'AUTHOR_COMMENT' => $comment,
                    'MODERATE_LINK'  => BASE_URL.'admin/index.php?module=discussions'
                );

                if (!empty($info['to'])) {
                    func_mail_admin_product_review_notify($info);
                }
            }
        } else {
            $err[] = $LABEL['err_cant_save_info'];
        }
    }
}
}
```

```
func_msgs_create($err, $msgs);
if (func_count_array($err) > 0) {
    $prod_discuss_req = new Request('c_product_discuss', 'add', $prod_id);

    $prod_discuss_req->SetParamValue('discuss_author', $author);
    $prod_discuss_req->SetParamValue('discuss_email', $email);
    $prod_discuss_req->SetParamValue('discuss_rate', $rate);
    $prod_discuss_req->SetParamValue('discuss_comment', $comment);

    header('Location: '.$LINKS->Product($prod_id, true, $from).'#reviews'); exit;
} else {
    header('Location: '.$LINKS->Product($prod_id, false, $from)); exit;
}
}

// Извлекаем список отзывов из БД
$discuss = ProdsDiscuss::GetAllApproved($prod_id, $SHOP_LANGUAGE['id']);

if (!empty($discuss)) {
    $SMARTY->assign('discuss', $discuss);
}

// Определим, оставлял ли пользователь уже отзыв о продукте
$SMARTY->assign('already_discussed',
    ProdsDiscuss::IsDiscussed($prod_id, $_SESSION['visitor_id'],
        isset($_SESSION['customer_info']['id']) ? $_SESSION['customer_info']['id'] : 0));

// Для генерирования изображения капчи
$SMARTY->assign('KCAPTCHA_IMG',
    KCAPTCHA_PATH.'index.php?'.session_name().'='.session_id().'&rnd='.mt_rand());
}
```

Как видим, основными моментами здесь являются следующие особенности:

- Сразу же после имени функции идет открывающая фигурная скобка. То же самое и с условными операторами.
- После открывающей фигурной скобки не должно быть пустых строк. Перед закрывающей фигурной скобкой также не должно быть пустых строк.
- Отступы от левого края организуются только табуляцией (при примере вставлены пробелы, но в PHP-файлах нужно вставлять табуляцию). При этом при просмотре текста в любом текстовом редакторе нужно устанавливать размер табуляции равным 4 пробелам (если текстовый редактор поддерживает такую возможность).
- В блоке объявления глобальных переменных в самом начале функции каждая глобальная переменная объявляется на отдельной строке.

- Операторы присваивания (а также знаки ? и : в условиях) должны находиться друг под другом (см. блок входящих параметров). В подобных случаях необходимо добавлять пробелы между операторами.
- Условные операторы должны отделяться пустой строкой сверху (и снизу) от предыдущего (и следующего) кода. Исключение – пустая строка в начале (сразу после открывающей фигурной скобки) или в конце функции (перед закрывающей фигурной скобкой) не нужна. Также пустые строки не нужно ставить в случае, если условный оператор расположен сразу после «родительского» условного оператора или в конце его.
- Код условного оператора должен обрамляться фигурными скобками даже если состоит всего из одной строки.
- При присваивании значений ключам массива также нужно соблюдать выравнивание ключей и значений друг под другом (см. на объявление массива **\$info** в примере выше).
- Комментарии (в т.ч. и многострочные) в коде нужно оформлять с помощью // и они должны быть на русском языке. При этом кодировка PHP-файла должна быть обязательно UTF-8 (без начальной сигнатуры).

### 7.3.2 HTML и Smarty-код

При оформлении кода Smarty-шаблонов нужно придерживаться следующих рекомендаций:

- Отступы слева должны быть организованы табуляцией.
- Пустые строки необходимо вставлять только в том случае, если они действительно необходимы для удобочитаемости кода.
- Код условного оператора должен быть написан с отступом слева от уровня {if} на одну табуляцию.
- Операторы {include file="..."} желательно обрамлять сверху и снизу пустыми строками (при условии, что оператор не стоит сразу за условием или перед его концом).
- Необходимо правильно оформлять вложенность HTML-тегов друг в друга – при необходимости делать отступ слева на одну табуляцию для соблюдения уровней вложенности тегов.

- Значения ВСЕХ элементов `<textarea>`, `<input type="text">` и `<input type="hidden">` нужно экранировать следующим образом:  
`<textarea>{$content|escape:'htmlall':'utf-8'}</textarea>` и `<input type="text" value="{ $value|replace:'":'&quot;}'" />` `<input type="hidden" value="{ $value|replace:'":'&quot;}'" />`.
- Во всех HTML-тегах ВСЕ значения атрибутов должны быть обрамлены в ДВОЙНЫЕ кавычки.
- Везде, где используется jQuery-функция **sortable**, нужно выносить тег **form** за пределы таблицы (во избежание неправильной сортировки), строки которой перемещаются пользователем мышью при сортировке.
- Если в файле шаблона присутствуют HTML-комментарии на русском языке, его необходимо сохранить в кодировке UTF-8 (без начальной сигнатуры), т.к. браузер IE6 не воспринимает JavaScript-код, содержащий русские комментарии, сохраненные не в той кодировке.
- В таблицах с несколькими строками фоновым цветом закрашиваются только нечетные по счету строки.

Приведем пример Smarty-кода с элементами HTML:

```
{if $CONFIG.modules.products.avail ne 'Y'}
  {capture name=product}
    <p>{$LABEL.lbl_module_disabled}</p>
  {/capture}

  {include file="_system_/center_block.html" title=$LABEL.lbl_product content=$smarty.capture.product}
{elseif $product}
  {include file="product/product_js.html"}

  {capture name=product}
    {if $prev_prod or $next_prod}
      {if $prev_prod}
        <div class="float-left">
           <a href="{ $LINKS->Product($prev_prod, false, $prod_from)}" title="{ $prev_prod.caption}">{$prev_prod.caption|truncate:35:"...":true}</a>
        </div>
      {/if}
      {if $next_prod}
        <div align="right">
          <a href="{ $LINKS->Product($next_prod, false, $prod_from)}" title="{ $next_prod.caption}">{$next_prod.caption|truncate:35:"...":true}</a> 
        </div>
      {/if}
    }
  {/capture}
```

```
</div>
{/if}
<div class="clearer"></div>
<div class="hr"></div>
{/if}

<div class="preview-container">
  {if $product.img}
    <div id="zoom-product-img">
      <a href="{ $PRODS_PICS}{ $product.img}" id="prod_img" class="fancy" rel="gal_pics" ti-
title="{ $product.caption}">{ $LABEL.link_enlarge}</a>
    </div>
  {/if}
  {strip}
    <div class="preview">
      {if $product.img}<a href="javascript:void();" onClick="$j('#prod_img').click(); return
false;">{/if}
      
      {if $product.img}</a>{/if}
    </div>
  {/strip}
  <div class="price">
    {if $CONFIG.settings.prices_for_only_registered eq 'on' &&
!$smarty.session.customer_info.id}
      {include file="_system/_helptip.html" id="product-`$product.id`-price"
text=$LABEL.tip_prices_for_only_registered label="<u>`$LABEL.lbl_price`</u>"}
    {else}
      {if $CONFIG.settings.prices_same_font_size eq 'on'}
        <span id="prod-price">{ $product.price}</span>
      {else}
        {strip}
          {if $SHOP_CURRENCY.left_side eq 'Y'}{ $SHOP_CURRENCY.short_name}{/if}
          <span id="prod-price-int">{ $product.price_int}</span>{if $prod-
uct.price_fract}<sup><span id="prod-price-fract">{ $product.price_fract}</span></sup>{/if}
          {if $SHOP_CURRENCY.left_side ne 'Y'}{ $SHOP_CURRENCY.short_name}{/if}
        {/strip}
      {/if}
    {/if}
  </div>
</div>

<h1>
  {if $product.egood eq 'Y'}
    
  {/if}
  { $product.caption}
```



```
</h1>
<div class="hr"></div>

<form method="post" id="product_form">
<table cellpadding="0" cellspacing="0" width="400">
<tr>
<td valign="top">
  {if $CONFIG.modules.discussions.avail eq 'Y'}
    <div class="vote-bar">
      {math equation="round(r * 4)" r=$product.rating|default:0 assign=rated_index}
      {section name=rating loop=21 start=1}
        <input name="star_total_{$product.id}" type="radio" class="star_total" {if $rated_index eq $smarty.section.rating.index}checked="checked"{/if} />
      {/section}
    </div>
  {/if}

  {if $CONFIG.settings.prices_for_only_registered ne 'on' or
  $smarty.session.customer_info.id}
    {if $product.prev_price and (not $product.options or $CONFIG.modules.prod_options.avail ne 'Y')}
      <div class="price-list">
        <span class="param">{$LABEL.lbl_old_price}:&nbsp;</span>
        <span class="value"><s>{$product.prev_price}</s></span>
      </div>
    {/if}
  {/if}

  <div class="price-list hidden" id="prod-discount-label">
    <span class="param">{$LABEL.lbl_discount}:&nbsp;</span>
    <span class="value" id="prod-discount">{$product.discount}</span>
  </div>

  <div class="price-list hidden" id="prod_sku_label">
    <span class="param">{$LABEL.lbl_sku}:&nbsp;</span>
    <span class="value" id="prod_sku">{$product.sku}</span>
  </div>

  {if $product.manuf_caption and $CONFIG.modules.manufacturers.avail eq 'Y'}
    <div class="link">
      {$LABEL.lbl_manufacturer}: <a href="{if $product.manuf_eng_name ne ''}{$LINKS->Manufacturer($product.manuf_eng_name)}{else}{$LINKS->Manufacturer($product.manuf_id)}{/if}">{$product.manuf_caption}</a>
    </div>
  {/if}

  {if $prod_pics}
    <div class="link">
      <a href="javascript:void(0);" onClick="$j.scrollTo('#prod_pics_anchor', 300);">{$LABEL.lbl_other_prod_pics}</a>
    </div>
  {/if}

```

```
</div>
{/if}

{if $product.options and $CONFIG.modules.prod_options.avail eq 'Y'}
  <table cellpadding="0" cellspacing="0" class="product-options">
    {foreach from=$product.options item=option}
      <tr>
        <td>{$option.caption}</td>
        <td>
          &nbsp;
          <select name="options[{$option.id}]" onChange="OptionChanged({$product.id});">
            {foreach from=$option.variants item=variant}
              <option value="{ $variant.id}" {if $product.def_vars_selected[$variant.id]}selected="selected"{/if}>
                { $variant.caption|replace: ' ': '&nbsp;' }
                {if $variant.price_mod}
                  ({$variant.price_mod})
                {/if}
              </option>
            {/foreach}
          </select>
        </td>
      </tr>
    {/foreach}
  </table>
  <p id="options_exception_detected" class="product-options hidden">
    <b>{$LABEL.err_invalid_combination}</b>
  </p>
{/if}

{if $product.tax_value}
  <p>{$LABEL.lbl_price_with_tax}</p>
{/if}
</td>
<td valign="top" class="right-part">
  <div class="link">
    <a href="javascript:void(0);" onClick="ShowSendToFriend(); return false;">{$LABEL.link_send_to_friend}</a>
  </div>
  <div class="link">
    {include file="products/wishlist_js.html"}
    <a href="javascript:void(0);" onClick="AddProdToWishlist({$product.id}, {$smarty.session.customer_info.id|default:0}); return false;">{$LABEL.link_prod_to_wishlist}</a>
  </div>
  {if $product.egood eq 'Y' and $product.is_free eq 'Y' and ($CONFIG.settings.disallow_free_egoods_for_anonym ne 'on' or $CONFIG.settings.disallow_free_egoods_for_anonym eq 'on' and $smarty.session.customer_info.id)}
    <div class="link">
      <a href="{ $LINKS->GetEgoodFree($curr_cat_id, $product.id)}">{$LABEL.link_free_download}</a>
    </div>
  {/if}
</td>
</tr>
</tbody>
</table>
</div>
{/if}
</div>
```

```

    </div>
  {/if}
  <div class="price-list hidden" id="prod_qty_label">
    <span class="param">{$LABEL.lbl_store_qty}:&nbsp;</span>
    <span class="value" id="prod_qty">{$product.qty}</span>
  </div>
  <p>
    <div id="add_to_cart_button" class="hidden">
      <table cellpadding="0" cellspacing="0">
        <tr>
          <td>
            {if $product.egood ne 'Y'}
              <td><input type="text" name="prod_qty" class="text spin" value="1" />&nbsp;</td>
            {/if}
            <td>{include file="_system_/button.html" js="AddToCart(`$product.id`,
`$curr_cat_id`);" title=$LABEL.link_add_to_cart}</td>
          </tr>
        </table>
      </div>
    </p>
  </td>
</tr>
</table>
</form>

{if $product.descr or $product.short_descr}
  <div id="description">
    <div class="hr"></div>
    {$product.descr|default:$product.short_descr}
  </div>
{/if}

{if $details}
  {include file="product/details.html"}
{/if}

{if $CONFIG.modules.recommended_products.avail eq 'Y' and $recommended_products}
  {include file="product/recommended.html"}
{/if}

{if $CONFIG.modules.related_products.avail eq 'Y' and $related_products}
  {include file="product/related.html"}
{/if}

{if $prod_pics}
  {include file="product/pictures.html"}
{/if}

{if $CONFIG.modules.discussions.avail eq 'Y'}
  {include file="product/discussions.html"}
{/if}

```

```
{include file="product/send_to_friend.html"}
{/capture}

{include file="_system_/center_block.html" title=$product.caption content=$smarty.capture.product id="product-details"}
{/if}
```

## 7.3.3 JavaScript-код

## 7.4 Прочие требования

Здесь описаны прочие требования, которые необходимо соблюдать для корректной работы CMS:

- каждая таблица БД должна содержать первичный ключ с именем **id** и свойством **auto\_increment**;
- для подсчета кол-ва элементов массива вместо функции **count** везде нужно использовать глобальную функцию [func\\_count\\_array](#). Даже в шаблонах Smarty;
- вместо функции **explode** следует использовать глобальную функцию `fn_explode`, т.к. стандартная функция **explode** неправильно работает с пустой строкой, возвращая в результате массив из одного пустого значения;
- перед использованием цикла **foreach** необходимо проверять, является ли переменная массивом, с помощью **is\_array**. При этом производить итерации, только если **is\_array** вернула истинное значение;
- любые строковые функции PHP, включая функции регулярных выражений, должны учитывать кодировку UTF-8, поэтому необходимо всегда вызывать эквивалентные строковые функции, начинающиеся с префикса **mb\_**;
- при вызове любой PHP-функции необходимо удостовериться (в документации к PHP), что функция присутствует в версии PHP, поддерживаемой магазином;
- в коде нельзя использовать опцию **include\_path** конфигурации, поскольку ее изменение не поддерживается некоторыми хостингами, а также ее наличие влияет на производительность. Вместо нее нужно прописывать в директивах **include/require** путь к подключаемому PHP-файлу относительно корня магазина, например: `include ROOT_DIR.'modules/products/products.php'`.

- в коде нельзя злоупотреблять **include/require** и тем более **include\_once/require\_once**, т.к. эти директивы весьма отрицательно влияют на быстродействие магазина;
- в SQL-запросах к базе нельзя оборачивать имена таблиц в косые кавычки (они уже обернуты в [\\$TABLES](#));
- все текстовые файлы магазина должны быть в Unix-формате, т.е. каждая строка в файле должна оканчиваться на `\n` (символ перевода строки), а не на `\r\n` (символы возврата каретки и перевода строки соответственно), как в Windows;
- нельзя обращаться к файлу, загруженному из браузера, напрямую (например, `fopen($_FILES['file']['tmp_name'])`), поскольку при включенной директиве **open\_basedir** PHP запретит к нему доступ. Вместо этого нужно совместно использовать PHP-функции **is\_uploaded\_file** и **move\_uploaded\_file**;
- в HTML-формах нельзя давать имя **encoding** какой-либо переменной (полю), т.к. могут быть проблемы с передачей такой формы с помощью Ajax в браузере Internet Explorer. Лучше давать такой переменной имя **enc**.

## 8. Справочник

### 8.1 Глобальные переменные в коде PHP

Любая глобальная переменная из представленных в данном разделе (кроме `$_SESSION`) перед использованием в любом месте кода CMS должна быть объявлена с помощью ключевого слова **global**.

#### **`$_SESSION['admin_info']`**

Данная переменная доступна в административной части (админке) магазина и является массивом с информацией о текущем администраторе, который работает в админке. Ключи данного массива совпадают с полями таблицы [proshop\\_users](#), например:

`$_SESSION['admin_info']['id']` – идентификатор текущего администратора

`$_SESSION['admin_info']['first_name']` – имя текущего администратора

#### **`$_SESSION['customer_info']`**

Данная переменная хранит информацию о покупателе, который вошел в свой аккаунт или совершил заказ без регистрации в течение данной сессии. Переменная является массивом и доступна только на витрине магазина. Ключи данного массива совпадают с полями таблицы [proshop\\_users](#), например:

`$_SESSION['customer_info']['id']` – идентификатор текущего зарегистрированного покупателя

`$_SESSION['first_name']` – имя текущего покупателя

`$_SESSION['last_name']` – фамилия текущего покупателя

Следует учесть, что идентификатор `$_SESSION['customer_info']['id']` не задан для тех пользователей, которые делают заказ без регистрации, т.к. для таких пользователей не создана запись в таблице [proshop\\_users](#) – они не могут иметь идентификатора из этой таблицы.

#### **`$_SESSION['visitor_id']`**

Данная переменная хранит идентификатор текущего посетителя из таблицы [proshop\\_visitors](#). Вся информацию о посетителе можно извлечь из таблицы, используя класс [Visitors](#).

## **\$BASE\_CURRENCY**

Данная переменная хранит информацию о базовой валюте магазина и доступна как в админке, так и на витрине. Переменная является массивом, ключи которого совпадают с полями таблицы [proshop\\_currencies](#), например:

`$BASE_CURRENCY['name']` – трехбуквенное обозначение базовой валюты (например, RUR)

`$BASE_CURRENCY['short_name']` – обозначение валюты возле значения цены (например, \$100)

## **\$CACHE**

Данная переменная является объектом класса [Cache](#) и используется для прозрачного взаимодействия с кэшем магазина (запись/извлечение кэшированной информации в БД, очистка кэша, актуализация кэша). Переменная доступна как в админке, так и на витрине. На витрине данный объект используется в основном для записи и извлечения кэшированной информации в БД (в зависимости от ее актуальности в памяти). В админке данный объект используется преимущественно для актуализации измененной информации в кэше магазина. Для более подробной информации об управлении кэшированием см. в [соответствующем разделе](#).

## **\$CBLOCKS**

Данная переменная доступна только на витрине и только при работе модуля главной страницы магазина (**homepage**) и хранит массив всех центральных блоков главной страницы. Массив представляет собой список всех записей из таблицы [proshop\\_cblocks](#), отсортированных по возрастанию значения поля **weight** и относящихся к центральной колонке главной страницы, например:

```
foreach ($CBLOCKS as $cblock) {  
    // $cblock['name'] – служебное имя блока  
    // $cblock['module'] – название модуля (из таблицы proshop\_modules), которому  
    // принадлежит блок  
    // ...  
}
```

## \$CONFIG

Данная переменная доступна как в админке, так и на витрине, и хранит в себе три массива:

- **\$CONFIG['settings']** – массив настроек магазина и их значений из таблицы [proshop\\_settings](#). Данные настройки могут изменяться администратором в админке. Для доступа к значению заданной настройки нужно задать ее название в качестве ключа: `$CONFIG['settings']['имя_настройки]`, например: `$CONFIG['settings']['shop_name']` - будет содержать название магазина.
- **\$CONFIG['options']** – массив опций (настроек) для различных модулей из таблицы [proshop\\_options](#). Для получения доступа к значению заданной опции необходимо задать ее имя в качестве ключа, например: `$CONFIG['options']['test_option']` – содержит значение опции с именем `test_option`.
- **\$CONFIG['modules']** – массив модулей из таблицы [proshop\\_modules](#). Используется преимущественно для определения включенности того или иного модуля, например: `$CONFIG['modules']['news']['avail']` – модуль новостей включен, если значение равно `Y`.

## \$CURRENCIES

Данная переменная доступна только на витрине и является массивом всех валют (из таблицы [proshop\\_currencies](#)), которые может выбирать пользователь на витрине для отображения различных ценовых значений. Например:

```
foreach ($CURRENCIES as $curr) {  
    // $curr['id'] – идентификатор валюты из таблицы proshop\_currencies  
    // $curr['name'] – международное трехбуквенное обозначение валюты  
}
```

## \$DB

Данная переменная является объектом класса [DB](#) и может использоваться как в админке, так и на витрине. Преимущественно данный объект используется только в классах и глобальных функциях. В соответствии с [архитектурой магазина](#), данный объект не должен использоваться в модулях магазина. Вместо этого модули должны использовать методы классов.



## **\$LABEL**

Данная переменная является массивом текстовых меток, которая доступна как в админке, так и на витрине. Причем, в админке в данном массиве содержатся только метки зоны админки, а на витрине – метки зоны витрины. На витрине недоступны метки админки и наоборот. Пример получения текста заданной метки на текущем языке: `$LABEL['lbl_news']` – текст метки **lbl\_news** на текущем языке, который определяется глобальной переменной [\\$SHOP\\_LANGUAGE](#).

## **\$LANGUAGES**

Данная переменная доступна как на витрине, так и в админке, и является массивом всех языков (из таблицы [proshop\\_languages](#)). Например:

```
foreach ($LANGUAGES as $lang) {  
    // $lang['id'] – идентификатор языка из таблицы proshop\_languages  
    // $lang['short'] – международное двухбуквенное обозначение языка  
}
```

## **\$LANGUAGES\_COUNT**

Данная переменная доступна как на витрине, так и в админке. Содержит размер глобального массива [\\$LANGUAGES](#).

## **\$BLOCKS**

Данная переменная доступна только на витрине и хранит массив всех блоков левой колонки для текущей страницы магазина. Массив представляет собой список всех записей из таблицы [proshop\\_cblocks](#), отсортированных по возрастанию значения поля **weight** и относящихся к левой колонке страницы, например:

```
foreach ($BLOCKS as $block) {  
    // $block['name'] – служебное имя блока  
    // $block['module'] – название модуля (из таблицы proshop\_modules), которому  
    // принадлежит блок  
    // ...  
}
```

## \$LINKS

Данная переменная является объектом класса [Links](#) и доступна как на витрине, так и в админке. С помощью данного объекта можно сформировать адрес какой-либо страницы витрины магазина. Например, формирование адреса страницы указанного продукта: `$LINKS->Product($prod_info)`.

## \$LOCATION

Данная переменная является объектом класса [Location](#) и доступна как в админке, так и на витрине. Например, формирование ссылки в строке навигации в админке:

```
$LOCATION->AppendItem('index.php?module=articles', $LABEL['lbl_articles']);
```

## \$META\_TAGS

Данная переменная предназначена для задания значений мета-тэгов текущей страницы (таких, как title, meta keys, meta description). Переменная является массивом и доступна только на витрине. Массив может содержать следующие ключи:

- **meta\_title** – значение тэга <title> для текущей страницы;
- **meta\_keys** – значение тэга <meta http-equiv="keywords" />;
- **meta\_desc** – значение тэга <meta http-equiv="description" />.

Например, если требуется задать заголовок текущей страницы модуля на витрине: `$META_TAGS['meta_title'] = $prod['caption'];`

## \$RBLOCKS

Данная переменная доступна только на витрине и хранит массив всех блоков правой колонки для текущей страницы магазина. Массив представляет собой список всех записей из таблицы [proshop\\_cblocks](#), отсортированных по возрастанию значения поля **weight** и относящихся к правой колонке страницы, например:

```
foreach ($RBLOCKS as $rblock) {  
    // $rblock['name'] – служебное имя блока  
    // $rblock['module'] – название модуля (из таблицы proshop\_modules), которому  
    // принадлежит блок  
    // ...  
}
```

## **\$SHOP\_CURRENCY**

Данная переменная доступна только на витрине и является массивом, описывающим текущую выбранную пользователем валюту для просмотра различных ценовых значений (например, цены товаров). Массив содержит ключи, совпадающие с полями таблицы [proshop\\_currencies](#). Например: `$SHOP_CURRENCY['name']` – трехбуквенное обозначение текущей выбранной валюты магазина.

## **\$SHOP\_LANGUAGE**

Данная переменная доступна как на витрине, так и в админке, и является массивом, описывающим текущий выбранный пользователем язык. Массив содержит ключи, совпадающие с полями таблицы [proshop\\_languages](#). Например: `$SHOP_LANGUAGE['title']` – название текущего выбранного языка.

## **\$SMARTY**

Данная переменная доступна как на витрине, так и в админке, и является объектом класса **Smarty**. Документация по использованию шаблонизатора Smarty находится на [сайте проекта Smarty](#).

Например, присвоение какого-либо значения переменной в шаблоне:

```
$SMARTY->assign('my_var', 'my_value');
```

## **\$TABLES**

Данная переменная является массивом таблиц БД и доступна как на витрине, так и в админке. Используется преимущественно [классами](#).

Массив состоит из ключей, каждый из которых является именем таблицы БД без префикса. Например, `$TABLES['products']` содержит строку `proshop_products`, т.е. полное название таблицы (с префиксом `proshop_`). Таким образом, к таблицам удобнее обращаться через данный глобальный массив, не обращая внимания на префикс таблицы, который может быть изменен в файле `includes/config.php` (если на то будет причина).

## \$TABLES\_RELS

Глобальный массив, доступный как в админке, так и на витрине магазина. Содержит связи между таблицами для контроля ссылочной целостности БД при удалении различных записей. Используется глобальной функцией [fn\\_del\\_db\\_entity](#).

Программный контроль ссылочной целостности введен в PHP-код из-за того, что в CMS используется база данных MySQL с движком MyISAM, который не поддерживает ссылочную целостность, но работает быстрее, чем движок InnoDB.

При создании новой таблицы, которая имеет связи с другими таблицами, необходимо прописать эти связи в данном глобальном массиве.

Например:

```
$TABLES_RELS = array(
    'access_ip' => array( // имя таблицы с первичным ключом
        'relations' => array(
            'access_ip_texts' => array('access_id' => 'd'), // имя таблицы с
            // вторичным ключом ('d' - удалить при удалении первичного ключа, 'u' - установить в 0
            // при удалении первичного ключа)
        ),
    ),
    // ...
);
```

Для указанного примера при удалении записи из таблицы **proshop\_access\_ip** будут также удалены связанные записи из таблицы **proshop\_access\_ip\_texts** (записи, значения поля **access\_id** которых совпали с удаляемым **id** из таблицы **proshop\_access\_ip**).

Еще один пример:

```
'categories' => array(
    'relations' => array(
        'categories' => array('cat' => 'd'),
        'categories_params' => array('cat_id' => 'd'),
        'categories_texts' => array('category_id' => 'd'),
        'orders_products' => array('cat_id' => 'u'),
        'products' => array('cat' => 'd'),
        'products_categories' => array('cat_id' => 'd'),
        'products_positions' => array('cat_id' => 'd'),
        'stats_cats_views' => array('cat_id' => 'd'),
    ),
    'files' => array(
```

```
'img' => array(CATS_PICS) // путь к директории файла, который будет
удален при удалении записи
    )
),
// ...
```

В данном примере описан ключ **files**, в котором прописывается путь к директории (который хранится в константе) файла, который будет удален при удалении записи из таблицы **proshop\_categories**. Предполагается, что имя файла прописано в поле **img** таблицы. В массиве допускается задавать несколько путей через запятую. См. для примеров содержимое файла **includes/config.php**.

## **\$TABS**

Данная переменная доступна только в админке и является объектом класса [Tabs](#). Используется преимущественно в Smarty-шаблонах админки для контроля отображения вкладок (табов) модулей.

## **\$msgs**

Данная переменная доступна как в админке, так и на витрине, и является массивом сообщений об успешности операции, которые необходимо показать пользователю в специальном окне, выдаваемом в браузере.

Например, если мы хотим показать пользователю сообщение, хранящееся в метке `msg_operation_success`: `$msgs[] = $LABEL['msg_operation_success'];`

## **\$err**

Данная переменная доступна как в админке, так и на витрине, и является массивом сообщений об ошибках, которые необходимо показать пользователю в специальном окне, выдаваемом в браузере.

Например, если мы хотим показать пользователю сообщение, хранящееся в метке `err_cant_save_info`: `$err[] = $LABEL['err_cant_save_info'];`

## **8.2 Глобальные переменные в коде Smarty**

### **\$BANNERS\_PATH**

Доступна на витрине и в админке. См. описание константы [BANNERS\\_PATH](#).

### **\$BASE\_CURRENCY**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$BASE\\_CURRENCY](#) для PHP.

### **\$BASE\_URL**

Доступна на витрине и в админке. См. описание константы [BASE\\_URL](#).

### **\$CATS\_PICS**

Доступна на витрине и в админке. Содержит полный URL к директории хранения изображений категорий с косой чертой на конце. Пример использования: ``

### **\$CONFIG**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$CONFIG](#) для PHP.

### **\$CURRENCIES**

Доступна только на витрине. См. описание глобальной переменной [\\$CURRENCIES](#) для PHP.

### **\$CURRENCIES\_COUNT**

Доступна только на витрине. Содержит кол-во элементов в массиве [\\$CURRENCIES](#).

### **\$CURRENT\_SECTION**

Доступна в админке. Содержит название текущей секции, в которой находится пользователь (для корректной работы быстрого меню админки сверху в шапке дизайна). Например: **catalog** – администратор находится в модуле, который связан с каталогом товаров; **users** – администратор находится в модуле, связанном с управлением пользователями. Полный список значений данной переменной можно найти в шаблоне `templates\admin\index.html`.

## **\$DEMO\_VERSION**

Доступна на витрине и в админке. Содержит значение true, если CMS запущена в демо-режиме. См. описание константы [DEMO\\_VERSION](#).

## **\$error404**

Доступна только на витрине. Содержит значение true, если текущая страница, которую просматривает пользователь, не найдена и требуется вывести сообщение с заголовком 404 Not Found.

## **\$FLAGS\_PICS**

Доступна на витрине и в админке. Содержит полный URL к директории хранения изображений флагов разных стран и территорий с косой чертой на конце, которая прописана в константе [FLAGS\\_PICS](#).

## **\$IMGS\_DIR**

Доступна на витрине и в админке. Содержит полный URL к директории хранения изображений дизайна с косой чертой на конце. Причем в админке содержит путь к директории изображений дизайна админки, а на витрине – путь к директории изображений дизайна витрины. Пример использования: ``

## **\$is\_checkout**

Доступна только на витрине. Содержит true, если в данный момент посетитель находится на странице корзины (оформляет заказ).

## **\$LABEL**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$LABEL](#) для PHP.

## **\$LANGUAGES**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$LANGUAGES](#) для PHP.

## **\$LANGUAGES\_COUNT**

Доступна на витрине и в админке. Содержит кол-во элементов в массиве [\\$LANGUAGES](#).

## **\$LIBS\_PATH**

Доступна на витрине и в админке. Содержит URL к директории хранения сторонних библиотек (**includes/libs**), используемых CMS.

## **\$LINKS**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$LINKS](#) для PHP.

## **\$LOCATION**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$LOCATION](#) для PHP.

## **\$MANUF\_PICS**

Доступна на витрине и в админке. Содержит полный URL к директории хранения изображений производителей с косой чертой на конце. Пример использования: ``

## **\$META\_TAGS**

Доступна только на витрине. См. описание глобальной переменной [\\$META\\_TAGS](#) для PHP.

## **\$module**

Доступна на витрине и в админке. Содержит название текущего работающего модуля. Если равна пустому значению, то считается, что текущий модуль – главная страница витрины или админки.

## **\$NEWS\_PICS**

Доступна на витрине и в админке. Содержит полный URL к директории хранения превью-изображений новостей с косой чертой на конце. ``



## **\$PRODS\_PICS**

Доступна на витрине и в админке. Содержит полный URL к директории хранения изображений продуктов с косой чертой на конце. Если на изображения продуктов наложен водяной знак с помощью модуля **watermarks**, возвращает URL к директории хранения изображений продуктов с водяным знаком с косой чертой на конце. Пример использования: ``

## **\$ROOT\_DIR**

Доступна на витрине и в админке. Аналог переменной [\\$BASE\\_URL](#).

## **\$SHOP\_CURRENCY**

Доступна только на витрине. См. описание глобальной переменной [\\$SHOP\\_CURRENCY](#) для PHP.

## **\$SHOP\_LANGUAGE**

Доступна на витрине и в админке. См. описание глобальной переменной [\\$SHOP\\_LANGUAGE](#) для PHP.

## **\$TABS**

Доступна только в админке. См. описание глобальной переменной [\\$TABS](#) для PHP.

## **\$TPLS\_PATH**

Доступна на витрине и в админке. Содержит URL директории, в которой хранятся шаблоны дизайна. Причем, в админке содержит URL директории шаблонов дизайна админки, а на витрине – URL директории шаблонов дизайна витрины.

## **8.3 Константы**

### **\_DEBUG\_**

Задается в файле **includes/defs.php** и определяет активность режима отладки. Если значение равно true, режим отладки включен. Иначе – режим отладки выключен.

Режим отладки – это режим, при котором все возникающие в CMS сообщения об ошибках в коде PHP (в т.ч. сообщения типа **notice**) или в шаблонах Smarty выводятся на

экран в браузер вместо их сокрытия и записи в директорию логов и таблицу [proshop\\_errors](#).

Режим отладки помогает обнаруживать возникающие в CMS ошибки при разработке какого-либо модуля или при устранении неисправностей.

При реальной эксплуатации CMS режим отладки **должен быть выключен**.

### **FAST\_SALES\_ADMIN\_STARTED**

Если данная константа установлена и равна true, это значит, что текущий код выполняется в административной части. Данная константа используется для проверки того, что текущий работающий в админке код предварен работой ядра магазина. Если это не так, код выполняться не должен из соображений безопасности и правильного функционирования работающего модуля или класса.

### **FAST\_SALES\_STARTED**

Если данная константа установлена и равна true, это значит, что текущий код выполняется на витрине. Данная константа используется для проверки того, что текущий работающий на витрине код предварен работой ядра магазина. Если это не так, код выполняться не должен из соображений безопасности и правильного функционирования работающего модуля или класса.

### **BACKUP\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения резервных копий магазина с косой чертой на конце. Используется модулем резервного копирования. См. описание директории [backups](#).

### **BANNERS\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения баннеров магазина с косой чертой на конце. Используется модулем баннеров. См. описание директории [uploads](#).

### **BASE\_URL**

Доступна на витрине и в админке. Содержит URL-адрес главной страницы витрины магазина.

## **BASE\_URL\_REW**

Доступна на витрине и в админке. Содержит часть URL-адреса главной страницы витрины магазина, идущую за именем хоста, например: `http://pro.fast-sales.ru/shop/` - здесь `/shop/` является такой частью.

## **CACHE\_ENABLED**

Доступна на витрине и в админке. Определяет, включено ли кэширование информации в магазине (использовать ли механизм кэша внутри методов класса [Cache](#)). Принимает значения `true/false`. Задается в файле `includes/defs.php`.

## **CATS\_PICS**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения изображений категорий магазина с косой чертой на конце. См. описание директории [images](#).

## **CHECK\_REQUIREMENTS**

Доступна на витрине и в админке. Определяет, включена ли проверка совместимости настроек сервера с CMS при каждой загрузке магазина. Принимает значения `true/false`. Задается в файле `includes/defs.php`. Рекомендуется выключать эту проверку при реальной эксплуатации магазина для его более быстрой работы.

## **COLOR\_SCHEMES\_ENABLED**

Доступна на витрине и в админке. Определяет, можно ли переключать цветовые схемы стандартного оформления витрины магазина в админке (и/или в демо-версии). Принимает значения `true/false`. Задается в файле `includes/defs.php`. Если в CMS необходимо встроить дизайн заказчика, данную директиву необходимо выключить.

## **CURRENT\_TIMESTAMP**

Содержит текущую метку времени Unix (которая может быть возвращена с помощью `time()`). При работе модулей магазина необходимо использовать именно эту константу для более быстрой работы CMS вместо вызова `time()`.

## DB\_HOST

Доступна на витрине и в админке. Содержит имя сервера БД. Задается в файле `includes/defs.php`.

## DB\_NAME

Доступна на витрине и в админке. Содержит имя БД. Задается в файле `includes/defs.php`.

## DB\_NAME\_PRECISION

Доступна на витрине и в админке. Определяет, уточнять ли имя БД при выполнении запросов к таблицам БД. Принимает значения true/false. Если значение данной константы равно true, запрос `'SELECT * FROM '.$TABLES['products']` будет интерпретирован как `SELECT * FROM `db_name`.proshop_products``, где `db_name` – имя БД из константы [DB\\_NAME](#). Если же значение данной константы равно false, запрос будет сформирован в более простом виде: `SELECT * FROM `proshop_products``.

Данная константа предназначена для поддержки совместимости с некоторыми серверами БД, в которых необходимо уточнять имя БД при обращении к таблице.

## DB\_PASS

Доступна на витрине и в админке. Содержит пароль пользователя БД. Задается в файле `includes/defs.php`.

## DB\_USER

Доступна на витрине и в админке. Содержит имя пользователя БД. Задается в файле `includes/defs.php`.

## DEMO\_VERSION

Доступна на витрине и в админке. Используется только разработчиком CMS (на сайте [fast-sales.ru](http://fast-sales.ru)). Включает/отключает режим демо-версии. Задается в файле `includes/defs.php`. Должна всегда принимать значение false. При включенном режиме демо-версии префиксы таблиц изменяются, поэтому если демо-режим включить без предварительного создания копий таблиц с другим префиксом (определяется в `includes/config.php`), CMS при работе выдаст ошибку доступа к таблицам БД.

## **EGOODS\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов цифровых продуктов магазина с косой чертой на конце. См. описание директории [uploads](#).

## **ENABLE\_ERROR\_LOG\_ROTATION**

Доступна на витрине и в админке. Включает/отключает режим ротации логов сообщений об ошибках в магазине, которые хранятся в директории [logs](#) и в таблице [proshop\\_errors](#). Режим ротации – режим, при котором при достижении заданного максимального кол-ва сообщений в логах самые старые сообщения удаляются при появлении новых. Максимальное кол-во сообщений задается константой [ERRORS\\_LOG\\_LIMIT](#). Задается в файле `includes/defs.php`.

## **ERRORS\_LOG\_FILE**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к файлу логов сообщений об ошибках CMS. Задается в файле `includes/defs.php`. См. описание директории [logs](#).

## **ERRORS\_LOG\_LIMIT**

Доступна на витрине и в админке. Максимальное кол-во сообщений в логе ошибок CMS, при которой включается ротация лога, если значение константы [ENABLE\\_ERROR\\_LOG\\_ROTATION](#) равно true. Задается в файле `includes/defs.php`.

## **EXPORT\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов экспорта продуктов, созданных с помощью модуля экспорта, с косой чертой на конце. См. описание директории [uploads](#).

## **FAULT\_LOGIN\_ATTEMPTS**

Доступна на витрине и в админке. Содержит максимальное кол-во неудачных попыток входа в день для включения капчи при входе покупателя в свой аккаунт. Задается в файле `includes/defs.php`.

## **FCKE\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов HTML-редактора FCKEditor, с косой чертой на конце. См. описание директории [includes](#).

## **FLAGS\_PICS**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения иконок флагов разных стран и территорий, с косой чертой на конце. См. описание директории [images](#).

## **FONTS\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов TTF-шрифтов, с косой чертой на конце. См. описание директории [includes](#). Файлы TTF-шрифтов используются модулем **watermarks**.

## **FS\_ONLY\_TABLES**

При объявлении данной константы (с присвоенным ей значением true) перед подключением файла **includes/config.php**, в нем не выполняется инициализация механизма кэша и не загружаются настройки магазина с помощью класса [Config](#). Такой режим может использоваться в тех случаях, когда нужно загрузить только константы, и глобальные переменные [\\$TABLES](#) и [\\$TABLES\\_RELS](#), определяемые в **includes/config.php**, без инициализации кэширования и настроек магазина.

## **IMGS\_DIR**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов изображений дизайна витрины (или админки – в зависимости от того, где данная константа используется), с косой чертой на конце. См. описание директории [templates](#).

## КCAPTCHA\_PATH

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов библиотеки **kcaptcha**, с косой чертой на конце. См. описание директории [includes](#).

## LIBS\_PATH

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения сторонних библиотек, используемых CMS, с косой чертой на конце. См. описание директории [includes](#).

## MAIL\_TEXTS\_PATH

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории **templates/mail\_texts** (с косой чертой на конце). Если требуется обратиться к любому файлу данной директории, необходимо использовать именно эту константу в качестве начального пути.

## MANUF\_PICS

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов изображений производителей (для модуля **manufacturers**), с косой чертой на конце. См. описание директории [images](#).

## MEMCACHE\_ALLOWED

Определяет, использовать ли memcache при кэшировании информации с помощью класса [Cache](#). Задается в **includes/defs.php** и может принимать значения true/false. Значение данной константы игнорируется, если значение константы [CACHE\\_ENABLED](#) равно false.

Не следует устанавливать значение данной константы в true, если точно не известно, имеется ли на сервере memcache, иначе при загрузке любой страницы магазина будет задержка во времени на таймаут соединения с memcache.

## MEMCACHE\_FLUSH\_ON\_RESTART

Определяет, сбросить ли кэш в memcache при следующей загрузке любой страницы магазина. Задается в **includes/defs.php** и может принимать значения true/false. В нормальной ситуации всегда должна быть равна false.

## MEMCACHE\_HOST

Определяет сервер, на котором расположен memcache. Задается в **includes/defs.php**. Значение по умолчанию – **localhost**.

## MEMCACHE\_PORT

Определяет порт сервера memcache. Задается в **includes/defs.php**. Значение по умолчанию – **11211**.

## NEWS\_PICS

Доступна на витрине и в админке (объявлена в **includes/config.php**). Содержит относительный путь (относительно корня магазина) к директории хранения файлов превью-изображений новостей, с косой чертой на конце. См. описание директории [images](#).

## NO\_PICTURE

Данная константа объявлена в **includes/config.php** и содержит в себе имя файла изображения по умолчанию, которое отображается на витрине вместо отсутствующих изображений товара/категории/производителя и т.п. При этом предполагается, что данное изображение располагается в директории, путь к которой прописан в константе [WM\\_PRODS\\_PICS](#), [NOWM\\_PRODS\\_PICS](#), [CATS\\_PICS](#) или [MANUF\\_PICS](#).

## NOWM\_PRODS\_PICS

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов изображений продуктов без водяного знака, с косой чертой на конце. См. описание директории [images](#).

## ROOT\_DIR

Доступна на витрине и в админке. Содержит относительный путь к корневой директории магазина в текущем модуле, с косой чертой на конце. Если требуется получить



доступ к файлу/директории CMS из любого модуля или класса, необходимо в начале прописать `ROOT_DIR`, например: `fopen(ROOT_DIR.'templates/admin/index.html', 'r')`. Другое пример: `include ROOT_DIR.'modules/account/account.php'`.

### **SHOP\_MODIF**

Доступна на витрине и в админке. Определяется в `includes/config.php`. Содержит короткое название модификации CMS. Может принимать значения: **Lite** или **Pro**. Для Fast-Sales Pro должна всегда быть **Pro** и не должна изменяться программистом. Предназначена преимущественно для правильной установки обновлений CMS с помощью модуля `updates`.

### **SHOP\_VERSION**

Доступна на витрине и в админке. Определяется в `includes/config.php`. Содержит версию текущей CMS. Например: 2.2.7. Данное значение не должно изменяться программистом и предназначено преимущественно для правильной установки обновлений CMS с помощью модуля `updates`.

### **TMP\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения временных файлов CMS, с косой чертой на конце. См. описание директории [uploads](#).

### **TPLS\_PATH**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов шаблонов дизайна магазина, с косой чертой на конце. Причем, если данная константа используется в админке, в ней содержится путь к шаблонам дизайна админки, а если на витрине – путь к шаблонам дизайна витрины (текущей темы оформления). См. описание директории [templates](#).

### **WM\_PRODS\_PICS**

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов изображений продуктов с водяным знаком (проставленным модулем `watermarks`), с косой чертой на конце. См. описание директории [images](#).

## WMARK\_PNG

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к PNG-файлу изображения водяного знака для модуля **watermarks**. См. описание директории [images](#).

## WMARK\_PNG\_THUMB

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к PNG-файлу уменьшенного варианта изображения водяного знака для модуля **watermarks**. См. описание директории [images](#).

## WMARKS\_PICS

Доступна на витрине и в админке. Содержит относительный путь (относительно корня магазина) к директории хранения файлов изображений водяного знака (нормального и уменьшенного вариантов) для модуля **watermarks**, с косой чертой на конце. См. описание директории [images](#).

## 8.4 Функции (PHP)

Все функции, определенные в **includes/functions.php**, доступны как в любом PHP-коде, так и в любом Smarty-коде.

### fn\_301\_redirect

Назначение:

- Корректная переадресация на указанный URL (код 301 - Moved Permanently).

Объявление:

- `function fn_301_redirect($url)`

Параметры:

- **string \$url**. URL-адрес, на который необходимо выполнить переход.

Возвращаемое значение:

- Функция ничего не возвращает.

### fn\_404\_not\_found

Назначение:

- Подготовка страницы для выдачи ошибки 404 на витрине. Данная функция выдает заголовок статуса 404, пишет в навигационной строке и в теге `<title>` магазина, что страница не найдена, и устанавливает флаг для Smarty о том, что нужно показать страницу ошибки 404. Дан-

ную функцию следует вызвать тогда, когда нужно показать страницу, но не удается это сделать, т.к. по заданным идентификаторам не найдена информация в БД.

Объявление:

- `function fn_404_not_found()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **fn\_add\_to\_cart**

Назначение:

- Добавление указанного товара в корзину покупателя.

Объявление:

- `function fn_add_to_cart($prod_id, $cat_id, $prod_qty, $options, $pos = 0)`

Параметры:

- **int \$prod\_id**. Идентификатор добавляемого продукта из таблицы [proshop\\_products](#).
- **int \$cat\_id**. Идентификатор категории (из таблицы [proshop\\_categories](#)), из которой добавляется продукт.
- **int \$prod\_qty**. Кол-во добавляемых экземпляров продукта.
- **array \$options**. Массив идентификаторов вариантов опций продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$pos**. Позиция добавляемого товара в корзине. Если равна 0, товар добавляется в конец списка продуктов корзины. Требуется для правильной сортировки товаров корзины при их обновлении/удалении.

Возвращаемое значение:

- Функция ничего не возвращает. Об успешности операции можно судить, проверив наличие сообщений об ошибках в глобальном массиве [\\$err](#).

## **fn\_cblock\_enabled**

Назначение:

- Определение, доступен ли в данный момент пользователю на витрине указанный блок (из таблицы [proshop\\_cblocks](#)).

Объявление:

- `function fn_cblock_enabled($name)`

Параметры:

- **string \$name**. Название блока из таблицы [proshop\\_cblocks](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## fn\_check\_admin\_rights

Назначение:

- Проверка, имеет ли право текущий администратор выполнять указанное действие в указанном модуле. Используется только в админке для разграничения прав администраторов. Более подробная информация содержится в [соответствующем разделе](#).

Объявление:

- `function fn_check_admin_rights($module, $act = "")`

Параметры:

- **string \$module**. Название модуля (из таблицы [proshop\\_modules](#)).
- **string \$act**. Действие в указанном модуле (из таблицы [proshop\\_modules\\_admin\\_actions](#)). Если данный параметр пуст, функция проверяет доступ к любому действию модуля (ко всему модулю в целом).

Возвращаемое значение:

- Тип **bool**. Возвращается значение true, если текущий администратор имеет права совершать заданное действие в заданном модуле. Иначе – возвращается false.

## fn\_check\_phone\_number

Назначение:

- Проверка заданного номера телефона на соответствие формату `+[код_страны][код_города][номер]`.

Объявление:

- `function fn_check_phone_number($num)`

Параметры:

- **string \$num**. Строка с номером телефона.

Возвращаемое значение:

- Тип **bool**. Возвращается true, если номер телефона задан в корректном формате. Иначе – false.

## fn\_check\_user\_fields

Назначение:

- Проверка полей профайла покупателя на правильную заполненность в соответствии с таблицей [proshop\\_users\\_fields](#). Используется модулем аккаунта покупателя и модулем корзины при совершении заказа для контроля правильности заполнения необходимой информации о покупателе. Если какие-либо необходимые поля не заполнены, функция записывает сообщения об ошибках в глобальный массив [\\$err](#).

Объявление:

- `function fn_check_user_fields($checkout, $user_info, $use_shipping_address)`

Параметры:

- **bool \$checkout.** Если данный параметр равен true, проверяются поля при совершении заказа покупателем. Если же значение параметра – false, проверяются поля при регистрации покупателя.
- **array \$user\_info.** Массив с полями информации о покупателе. Должен содержать ключи, совпадающие с некоторыми полями таблицы [proshop\\_users](#) – с какими именно, см. код функции.
- **bool \$use\_shipping\_address.** Если данный параметр равен true, функция проверит правильность заполнения полей, названия которых начинаются с префикса **ship\_**, т.е. полей, относящихся к информации об отдельном адресе доставки заказа.

Возвращаемое значение:

- Функция ничего не возвращает. Если функция найдет ошибки заполнения полей, она запишет их в глобальный массив [\\$err](#).

## fn\_del\_db\_entity

Назначение:

- Каскадное удаление указанной сущности из БД (с соблюдением всех связей между таблицами). Функция использует глобальный массив связей между таблицами БД [\\$TABLES\\_RELS](#).

Объявление:

- `function fn_del_db_entity($main_table, $id = null)`

Параметры:

- **string \$main\_table.** Название таблицы без префикса. Например, если требуется задать таблицу **proshop\_products**, название без префикса **proshop\_** будет **products**.
- **array \$id.** Массив идентификаторов записей в заданной таблице, которые необходимо удалить из таблицы с сохранением ссылочной целостности.

Возвращаемое значение:

- Функция ничего не возвращает.

## fn\_filter\_city

Назначение:

- Фильтрация названия города (вырезание префикса «г.» и удаление пробелов в начале и в конце названия).

Объявление:

- `function fn_filter_city($city)`

Параметры:

- **string \$city.** Название города.

Возвращаемое значение:

- Тип **string**. Отфильтрованное название города.

## fn\_explode

Назначение:

- Замена стандартной PHP-функции **explode**. Более правильный вариант.

Объявление:

- `function fn_explode($delimiter, $string)`

Параметры:

- **string \$delimiter**. См. аналогичный параметр стандартной PHP-функции **explode**.
- **string \$string**. См. аналогичный параметр стандартной PHP-функции **explode**.

Возвращаемое значение:

- Тип **array**. Возвращает массив подстрок строки `$string`, разделенных разделителем `$delimiter`. Если параметр `$string` является пустой строкой, функция возвращает пустой массив.

## fn\_get\_countries

Назначение:

- Извлечение списка видимых стран из БД и подготовка этого списка для вывода в шаблоне. После того, как вызвана данная функция, можно использовать переменную **\$countries** в шаблоне Smarty.

Объявление:

- `function fn_get_countries($user_info = "")`

Параметры:

- **array \$user\_info**. Массив с информацией о зарегистрированном пользователе. Ключи данного массива должны совпадать с полями таблицы [proshop\\_users](#). Данный массив нужен для того, чтобы отобразить в списке стран страну пользователя даже в том случае, если она не видима (неактивна, выключена). Если данный параметр **\$user\_info** не задан, в списке стран будут присутствовать только те страны, которые видимы (включены для отображения в магазине).

Возвращаемое значение:

- Функция ничего не возвращает, но вместо этого записывает массив стран в переменную **\$countries**, которую можно использовать в Smarty-шаблоне. Данная переменная является массивом, каждый элемент которого является массивом, состоящим из ключей, совпадающих с полями таблицы [proshop\\_countries](#).

## fn\_img\_check

Назначение:

- Проверка размера файла указанного изображения продукта, а также его ширины/высоты, перед загрузкой в магазин через админку. Если размеры изображения выходят за рамки

ограничений, установленных в настройках администратором, функция записывает сообщения об ошибках в глобальный массив [\\$err](#) и возвращает false.

Объявление:

- `function fn_img_check($filename)`

Параметры:

- **string \$filename**. Путь к файлу изображения.

Возвращаемое значение:

- Тип **bool**. True, если изображение не выходит за рамки, указанные в настройках. Иначе – возвращается false и записываются сообщения об ошибках в глобальный массив [\\$err](#).

## fn\_incline\_word

Назначение:

- Склонение слова для правильного его отображения вместе с числом (например, 15 товаров, 2 товара, 1 товар).

Объявление:

- `function fn_incline_word($num, $words, $type = 'noun')`

Параметры:

- **int \$num**. Число, для которого необходимо преобразовать слово.
- **array \$words**. Массив из трех слов, который содержит склоненные формы слова.
- **string \$type**. Тип слова. В данный момент поддерживается только один тип – **noun** (существительное).

Возвращаемое значение:

- Тип **string**. Слово - один из трех вариантов слов, указанных в массиве **\$words**, в зависимости от параметра **\$num**.

Пример:

- `echo fn_incline_word(1, array('товар', 'товара', 'товаров')); // отобразит 1 товар`  
`echo fn_incline_word(2, array('товар', 'товара', 'товаров')); // отобразит 2 товара`  
`echo fn_incline_word(7, array('товар', 'товара', 'товаров')); // отобразит 7 товаров`  
`echo fn_incline_word(22, array('товар', 'товара', 'товаров')); // отобразит 22 товара`

## fn\_is\_urlname\_correct

Назначение:

- Проверка заданного имени в URL (для modRewrite) любого объекта (продукта, категории, производителя и т.п.) на корректность. Перед выполнением данной функции рекомендуется воспользоваться функцией [func\\_correct\\_engname](#) для корректировки имени в URL.

Объявление:

- `function fn_is_urlname_correct($name)`

Параметры:

- **string \$name**. Проверяемое имя в URL.

Возвращаемое значение:

- Тип **bool**. Возвращается **true**, если имя корректно или не задано (пустая строка). Иначе – возвращается **false**.

## fn\_json2array

Назначение:

- Конвертация JSON-строки в массив-аналог (представление JSON в виде массива).

Объявление:

- `function fn_json2array($json)`

Параметры:

- **string \$json**. Текстовая JSON-строка.

Возвращаемое значение:

- Тип **array**. JSON в виде массива.

## fn\_make\_http\_post\_request

Назначение:

- Отправка POST-запроса по указанному URL. Функция использует библиотеку CURL.

Объявление:

- `function fn_make_http_post_request($url, $post, $ssl = false, $headers = "", $uagent = "")`

Параметры:

- **string \$url**. Полный путь к удаленному скрипту, POST-запрос которому необходимо отправить.
- **array \$post**. Переменные и их значения для отправки методом POST. Массив должен быть в следующем формате: `array('name1' => 'value1', 'name2' => 'value2')`.
- **bool \$ssl**. Если значение данного параметра равно `true`, предполагается, что адрес `$url` задан по протоколу SSL (HTTPS), и функция при этом должна игнорировать проверку действительности сертификата безопасности.
- **array \$headers**. Массив дополнительных заголовков запроса, которые следует передать при формировании POST-запроса.
- **string \$uagent**. Строка User Agent, которую можно передать удаленному скрипту.

Возвращаемое значение:

- Тип **string/bool**. `False` – в случае ошибки. Иначе – содержимое страницы, которое выдал удаленный скрипт на заданный запрос.

## fn\_month2str

вНазначение:



- Конвертирование номера месяца (от 1 до 12) в его краткое символьное представление на текущем языке.

Объявление:

- `function fn_month2str($month)`

Параметры:

- **int \$month**. Номер месяца от 1 до 12.

Возвращаемое значение:

- Тип **string**. Строковое представление указанного месяца. Берется из текстовых меток, определенных в глобальном массиве [\\$LABEL](#).

## fn\_msgs\_array\_to\_str

Назначение:

- Конвертирование массива сообщений (об успешности операции или об ошибках) в сплошную строку, закодированную алгоритмом Base 64, для ее использования в JavaScript-функции [ProcessMsgs](#).

Объявление:

- `function fn_msgs_array_to_str($msgs)`

Параметры:

- **array \$msgs**. Массив сообщений.

Возвращаемое значение:

- Тип **string**. Строка сообщений в кодировке Base 64.

## fn\_price\_round

Назначение:

- Округление заданного значения цены до целого числа, если соответствующая настройка включена администратором магазина.

Объявление:

- `function fn_price_round($price)`

Параметры:

- **float \$price**. Значение цены.

Возвращаемое значение:

- Тип **float**. Округленное до целого значение цены, если включена настройка округления. Иначе – неизменное значение цены.

## fn\_set\_products\_list

Назначение:

- Сохранение массива идентификаторов продуктов в сессии для указанного списка продуктов на витрине. Данное сохранение необходимо для навигации на следующий/предыдущий продукт при нахождении пользователя на странице продукта.

Объявление:

- `function fn_set_products_list($type, $ids = Array(), $pager = NULL, $rules = Array(), $products_count = 0)`

Параметры:

- **string \$type**. Тип списка продуктов (например, **bestsellers**, **newest**, и т.п.) – в зависимости от того, при формировании какого списка данная функция вызывается. Если данный параметр пуст, он считается равным строке **catalog**.
- **array \$ids**. Массив идентификаторов продуктов списка – в том порядке, в котором продукты отображались в списке. Если данный массив задан, остальные параметры функции (идущие после данного параметра) игнорируются.
- **object \$pager**. Объект класса `Pager`. Используется для формирования списка идентификаторов продуктов в соответствии с текущим списком, формируемым указанными правилами **\$rules** (для функции [Products::Search](#)), если параметр **\$ids** не задан (массив пуст). Данный параметр используется совместно с параметрами **\$rules** и **\$products\_count**.
- **array \$rules**. Правило поиска продуктов для функции [Products::Search](#) для формирования списка идентификаторов продуктов. Данный параметр используется совместно с параметрами **\$pager** и **\$products\_count**, если параметр **\$ids** не задан (массив пуст).
- **int \$products\_count**. Данный параметр используется совместно с параметрами **\$pager** и **\$rules**, если параметр **\$ids** не задан (массив пуст). Данный параметр содержит кол-во всех продуктов в списке и необходим для функции [setPageSize](#) класса [Pager](#).

Возвращаемое значение:

- Функция ничего не возвращает. Вместо этого она записывает список идентификаторов продуктов в сессию: `$_SESSION['products_list'][$type]`, где **\$type** – параметр данной функции. Данный массив идентификаторов используется на странице карточки продукта для определения следующего/предыдущего продуктов в навигации.

### fn\_smarty\_compile\_tpls

Назначение:

- Компилирование шаблонов дизайна Smarty, находящихся в указанной директории. Указанная директория должна совпадать с **SMARTY->template\_dir**. Функцией принимаются во внимание только файлы с расширением **html**.

Объявление:

- `function fn_smarty_compile_tpls($path)`

Параметры:

- **string \$path**. Путь к директории шаблонов дизайна, которые необходимо откомпилировать.

Возвращаемое значение:

- Функция ничего не возвращает.

## fn\_translit

Назначение:

- Транслитерация русского текста на латинский вариант (также пробелы заменяются дефисами).

Объявление:

- `function fn_translit($str)`

Параметры:

- **string \$str**. Строка русского текста в кодировке UTF-8.

Возвращаемое значение:

- Тип **string**. Латинский эквивалент строки **\$str**.

## fn\_user\_profile\_fields

Назначение:

- Подготовка специальных переменных Smarty для использования в шаблонах формирования форм регистрации и редактирования данных пользователя. Используется перед тем, как нужно вывести такую форму, чтобы сформировать набор обязательных/необязательных полей, которые должны быть заполнены пользователем.

Объявление:

- `function fn_user_profile_fields($checkout)`

Параметры:

- **bool \$checkout**. Если данный параметр равен **true**, то нужно сформировать форму при оформлении заказа. Если **false** – сформировать форму для обычной регистрации пользователя или для редактирования его данных.

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_check\_email

Назначение:

- Проверка корректности указанного адреса электронной почты.

Объявление:

- `function func_check_email($email)`

Параметры:

- **string \$email**. Адрес электронной почты для проверки.

Возвращаемое значение:

- Тип **bool**. Если адрес корректен, возвращается **true**. Иначе – **false**.

## func\_check\_install\_dir

Назначение:

- Проверка наличия в корне CMS директории инсталляции **install**. Проверка осуществляется в админке. Если директория обнаруживается, функция вызывает [func\\_msgs\\_append](#), чтобы добавить сообщение об ошибке, которое увидит администратор.

Объявление:

- `function func_check_install_dir()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_check\_ip

Назначение:

- Проверка IP-адреса (IPv4) на корректность. Используется в модуле админки **access**.

Объявление:

- `function func_check_ip($ip)`

Параметры:

- **string \$ip**. Значение IP-адреса.

Возвращаемое значение:

- Тип **bool**. Возвращается true, если IP-адрес корректен. Иначе – false.

## func\_clear\_dir

Назначение:

- Удаление всех файлов и директорий в указанной директории. Сама директория не удаляется.

Объявление:

- `function func_clear_dir($dir, $excluded = array(), $recursively = false)`

Параметры:

- **string \$dir**. Путь к директории, которую необходимо очистить.
- **array \$excluded**. Массив имен файлов (и/или директорий), которые необходимо проигнорировать при очистке.
- **bool \$recursively**. Если значение данного параметра равно true, функция удаляет все файлы, включая вложенные директории. Если false – удаляются только файлы в пределах указанной директории **\$dir**, файлы вложенных директорий и сами вложенные директории при этом не затрагиваются.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## func\_convert\_sum

Назначение:

- Конвертирование указанной суммы из одной валюты магазина в другую.

Объявление:

- `function func_convert_sum($sum, $to_currency, $enable_format = true, $enable_name = true, $from_currency = ")`

Параметры:

- **float \$sum**. Сумма, которую необходимо конвертировать из валюты **\$from\_currency** в валюту **\$to\_currency**.
- **array \$to\_currency**. Массив информации о валюте (из таблицы [proshop\\_currencies](#)), в которую необходимо конвертировать сумму **\$sum**. Чаще всего в качестве этого значения передается [\\$SHOP\\_CURRENCY](#).
- **bool \$enable\_format**. Если значение данного параметра равно true, итоговое значение сконвертированной суммы будет отформатировано в соответствии с настройками отображения валюты **\$to\_currency**.
- **bool \$enable\_name**. Если значение данного параметра равно true, к итоговому значению сконвертированной суммы будет добавлено обозначение валюты **\$to\_currency**.
- **array \$from\_currency**. Массив информации о валюте (из таблицы [proshop\\_currencies](#)), в которой задана сумма **\$sum**. Если данный массив не задан, предполагается, что сумма **\$sum** задана в базовой валюте (аналогично, если бы в качестве **\$from\_currency** мы передали [\\$BASE\\_CURRENCY](#)).

Возвращаемое значение:

- Тип **string/float**. Тип возвращаемого значения зависит от параметров **\$enable\_format** и **\$enable\_name**. Если хотя бы один из них равен true, возвращается строка. Иначе – число с плавающей точкой.

Пример:

- `$price = func_convert_sum($product['price'], $SHOP_CURRENCY);`  
`// $price содержит, к примеру, строку 1 423,34 p.`
- `$price = func_convert_sum($product['price'], $SHOP_CURRENCY, true, false);`  
`// $price содержит, к примеру, строку 1 423,34`

## func\_correct\_engname

Назначение:

- Корректировка заданного имени в URL (для modRewrite) любого объекта (например, продукт или статья) перед его помещением в БД. Из имени убираются HTML-тэги, оно преобразуется в нижний регистр, а пробелы заменяются дефисами.

Объявление:

- `function func_correct_engname($eng_name)`

Параметры:

- **string \$eng\_name**. Заданное имя в URL.

Возвращаемое значение:

- Тип **string**. Отфильтрованное имя в URL.

## **func\_correct\_number**

Назначение:

- Коррекция (и проверка на корректность) указанного числа (целого или с плавающей точкой) перед его помещением в БД.

Объявление:

- `function func_correct_number($num, $is_int = false)`

Параметры:

- **string \$num**. Число, значение которого необходимо проверить и откорректировать.
- **bool \$is\_int**. Если значение данного параметра равно true, число **\$num** воспринимается как целочисленное. Иначе – как число с плавающей точкой.

Возвращаемое значение:

- Тип **float/int/string**. Откорректированное значение числа или пустая строка, если число задано в некорректном формате или параметр **\$num** не является числом.

## **func\_count\_array**

Назначение:

- Обертка над стандартной PHP-функцией **count**. Для подсчета кол-ва элементов массива необходимо использовать именно эту обертку, поскольку стандартная функция **count** возвращает 1 для булевых переменных (в т.ч. имеющих значение **false**). Т.е. если данной обертке будет передан не массив, она вернет 0.

Объявление:

- `function func_count_array($array)`

Параметры:

- **array \$array**. Массив, кол-во элементов которого необходимо определить.

Возвращаемое значение:

- Тип **int**. Размер массива.

## **func\_course\_correct**

Назначение:

- Проверка и корректировка значения курса валюты перед помещением его в БД.

Объявление:

- `function func_course_correct($course)`

Параметры:

- **string \$course**. Строка со значением курса валюты.

Возвращаемое значение:

- Тип **float/bool**. Возвращается false, если заданное значение в неверном формате. Иначе – откорректированное значение курса.

## func\_file\_get\_contents

Назначение:

- Извлечение содержимого заданного текстового файла.

Объявление:

- `function func_file_get_contents($filepath)`

Параметры:

- **string \$filepath**. Путь к текстовому файлу, содержимое которого необходимо извлечь.

Возвращаемое значение:

- Тип **string/bool**. False, если не удалось извлечь содержимое файла. Иначе – текст файла.

## func\_file\_put\_contents

Назначение:

- Сохранение текста в заданный текстовый файл. Если файл не существует, он будет создан. Если файл уже существует, он будет перезаписан.

Объявление:

- `function func_file_put_contents($filepath, $content)`

Параметры:

- **string \$filepath**. Путь к текстовому файлу, в который необходимо записать текст.
- **string \$content**. Текстовое содержимое для записи в файл.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## func\_filesize

Назначение:

- Определение размера указанного файла в байтах.

Объявление:

- `function func_filesize($filename)`

Параметры:

- **string \$filename**. Путь к файлу, размер которого необходимо определить.

Возвращаемое значение:

- Тип **int**. Размер файла в байтах.

## func\_get\_discounts

Назначение:

- Извлечение всех возможных скидок указанного типа из БД и запись полученного массива в Smarty-переменную с указанным именем. Данная функция используется в админке.

Объявление:

- `function func_get_discounts($elem_type, $assign_name = 'discounts', $display_value = true)`

Параметры:

- **string \$elem\_type**. Тип скидки (из таблицы [proshop\\_discounts\\_types](#)).
- **string \$assign\_name**. Имя Smarty-переменной, которой будет присвоено полученное в результате работы функции значение массива скидок.
- **bool \$display\_value**. Если значение данного параметра равно true, к названию скидки добавляется ее величина (в скобках).

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_get\_month\_len

Назначение:

- Определение кол-ва дней в указанном месяце указанного года.

Объявление:

- `function func_get_month_len($month, $year)`

Параметры:

- **int \$month**. Номер месяца (от 1 до 12).
- **int \$year**. Номер года (4 цифры).

Возвращаемое значение:

- Тип **int**. Кол-во дней в указанном месяце указанного года.

## func\_get\_taxes

Назначение:

- Извлечение всех возможных налогов на продукты из БД и запись полученного массива в Smarty-переменную **\$taxes**. Данная функция используется в админке.

Объявление:

- `function func_get_taxes()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.



## func\_img\_resize

Назначение:

- Масштабирование изображения формата JPEG, GIF или PNG с последующим его сохранением (или выдачей в браузер) в формате JPEG указанного качества.

Объявление:

- `function func_img_resize($src, $dest, $width, $height, $rgb = 0xFFFFFF, $quality = 100, $limit_width = false)`

Параметры:

- **string \$src**. Путь к файлу исходного изображения.
- **string \$dest**. Путь к файлу результирующего изображения. Если данный параметр равен `null`, изображение выдается в браузер.
- **int \$width**. Новая ширина изображения в пикс.
- **int \$height**. Новая высота изображения в пикс.
- **int \$rgb**. Значение фонового цвета у свободного пространства нового изображения.
- **int \$quality**. Степень качества результирующего JPEG-изображения (по шкале от 1 до 100). 1 – самое плохое, но самое мелкое по размеру файла. 100 – самое качественное, но самое большое по размеру файла.
- **bool \$limit\_width**. Если значение данного параметра равно `true`, а значение параметра **\$width** превышает значение ширины изображения **\$src**, размеры изображения изменены не будут.

Возвращаемое значение:

- Если параметр **\$dest** задан, возвращается `true/false` – в зависимости от успешности операции. Иначе – ничего не возвращается.

## func\_mail\_admin\_cancel\_order\_notify

Назначение:

- Отправка письма-оповещения администратору о том, что заказ отменен покупателем.

Объявление:

- `function func_mail_admin_cancel_order_notify($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;

- **CUSTOMER\_FIRST\_NAME** – имя покупателя;
- **CUSTOMER\_LAST\_NAME** – фамилия покупателя;
- **CUSTOMER\_ADDRESS** – строка адреса (улицы) покупателя;
- **CUSTOMER\_ZIP\_CODE** – почтовый индекс покупателя;
- **CUSTOMER\_CITY** – город покупателя;
- **CUSTOMER\_COUNTRY** – страна покупателя;
- **CUSTOMER\_STATE** – область в стране покупателя;
- **CUSTOMER\_PHONE** – номер(а) телефона покупателя;
- **CUSTOMER\_MOBILE\_PHONE** – номер мобильного телефона покупателя;
- **CUSTOMER\_FAX** – номер факса покупателя;
- **CUSTOMER\_WEBSITE** – адрес сайта покупателя;
- **CUSTOMER\_COMPANY** – название компании (организации) покупателя;
- **CUSTOMER\_INN** – ИНН организации покупателя;
- **CUSTOMER\_KPP** – КПП организации покупателя;
- **CUSTOMER\_BANK\_NAME** – наименование банка покупателя;
- **CUSTOMER\_SHIP\_ADDRESS** – строка отдельного адреса (улицы) доставки заказа покупателю;
- **CUSTOMER\_SHIP\_ZIP\_CODE** – отдельный почтовый индекс для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_CITY** – отдельный город для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_COUNTRY** – отдельная страна для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_STATE** – отдельная область в стране покупателя для доставки заказа;
- **ORDER\_ID** – номер заказа;
- **PREV\_ORDER\_STATUS** – предыдущий статус заказа;
- **ORDER\_SHIP\_COST** – стоимость доставки заказа;
- **ORDER\_TOTAL\_SUM** – сумма заказа с учетом стоимости доставки;
- **ORDER\_QUANTITY** – кол-во заказанных продуктов;
- **ORDER\_SHIP\_METHOD** – метод доставки заказа покупателю;
- **ORDER\_PAY\_METHOD** – метод оплаты заказа;
- **ORDER\_PRODUCTS** – массив с информацией о заказанных продуктах. Каждый его элемент должен являться массивом и содержать следующие ключи:
  - **CATEGORY** – название категории продукта;
  - **CAPTION** – название продукта;
  - **SKU** – артикул продукта;
  - **PRICE** – цена одного экземпляра продукта;
  - **QUANTITY** – кол-во заказанных экземпляров продукта;
  - **EGOOD** – принимает значение **Y**, если товар является цифровым (скачиваемым). Иначе должен иметь значение **N**.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_admin\_cancel\_order\_prods\_notify

Назначение:

- Отправка письма-оповещения администратору о том, что некоторые продукты в заказе отменены покупателем.

Объявление:

- `function func_mail_admin_cancel_order_prods_notify($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **CUSTOMER\_FIRST\_NAME** – имя покупателя;
  - **CUSTOMER\_LAST\_NAME** – фамилия покупателя;
  - **CUSTOMER\_ADDRESS** – строка адреса (улицы) покупателя;
  - **CUSTOMER\_ZIP\_CODE** – почтовый индекс покупателя;
  - **CUSTOMER\_CITY** – город покупателя;
  - **CUSTOMER\_COUNTRY** – страна покупателя;
  - **CUSTOMER\_STATE** – область в стране покупателя;
  - **CUSTOMER\_PHONE** – номер(а) телефона покупателя;
  - **CUSTOMER\_MOBILE\_PHONE** – номер мобильного телефона покупателя;
  - **CUSTOMER\_FAX** – номер факса покупателя;
  - **CUSTOMER\_WEBSITE** – адрес сайта покупателя;
  - **CUSTOMER\_COMPANY** – название компании (организации) покупателя;
  - **CUSTOMER\_INN** – ИНН организации покупателя;
  - **CUSTOMER\_KPP** – КПП организации покупателя;
  - **CUSTOMER\_BANK\_NAME** – наименование банка покупателя;
  - **CUSTOMER\_SHIP\_ADDRESS** – строка отдельного адреса (улицы) доставки заказа покупателю;
  - **CUSTOMER\_SHIP\_ZIP\_CODE** – отдельный почтовый индекс для доставки заказа покупателю;
  - **CUSTOMER\_SHIP\_CITY** – отдельный город для доставки заказа покупателю;
  - **CUSTOMER\_SHIP\_COUNTRY** – отдельная страна для доставки заказа покупателю;

- **CUSTOMER\_SHIP\_STATE** – отдельная область в стране покупателя для доставки заказа;
- **ORDER\_ID** – номер заказа;
- **ORDER\_STATUS** – текущий статус заказа;
- **ORDER\_SHIP\_COST** – стоимость доставки заказа;
- **ORDER\_TOTAL\_SUM** – сумма заказа с учетом стоимости доставки;
- **ORDER\_QUANTITY** – кол-во заказанных продуктов;
- **ORDER\_SHIP\_METHOD** – метод доставки заказа покупателю;
- **ORDER\_PAY\_METHOD** – метод оплаты заказа;
- **ORDER\_PRODUCTS** – массив с информацией о заказанных продуктах. Каждый его элемент должен являться массивом и содержать следующие ключи:
  - **CATEGORY** – название категории продукта;
  - **CAPTION** – название продукта;
  - **SKU** – артикул продукта;
  - **PRICE** – цена одного экземпляра продукта;
  - **QUANTITY** – кол-во заказанных экземпляров продукта;
  - **EGOOD** – принимает значение **Y**, если товар является цифровым (скачиваемым). Иначе должен иметь значение **N**;
  - **CANCELLED** – принимает значение **Y**, если заказ данного товара отменен.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_admin\_order\_notify

Назначение:

- Отправка письма-оповещения администратору о поступлении нового заказа.

Объявление:

- function func\_mail\_admin\_order\_notify(\$info)

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **CUSTOMER\_FIRST\_NAME** – имя покупателя;
  - **CUSTOMER\_LAST\_NAME** – фамилия покупателя;

- **CUSTOMER\_ADDRESS** – строка адреса (улицы) покупателя;
- **CUSTOMER\_ZIP\_CODE** – почтовый индекс покупателя;
- **CUSTOMER\_CITY** – город покупателя;
- **CUSTOMER\_COUNTRY** – страна покупателя;
- **CUSTOMER\_STATE** – область в стране покупателя;
- **CUSTOMER\_PHONE** – номер(а) телефона покупателя;
- **CUSTOMER\_MOBILE\_PHONE** – номер мобильного телефона покупателя;
- **CUSTOMER\_FAX** – номер факса покупателя;
- **CUSTOMER\_WEBSITE** – адрес сайта покупателя;
- **CUSTOMER\_COMPANY** – название компании (организации) покупателя;
- **CUSTOMER\_INN** – ИНН организации покупателя;
- **CUSTOMER\_KPP** – КПП организации покупателя;
- **CUSTOMER\_BANK\_NAME** – наименование банка покупателя;
- **CUSTOMER\_SHIP\_ADDRESS** – строка отдельного адреса (улицы) доставки заказа покупателю;
- **CUSTOMER\_SHIP\_ZIP\_CODE** – отдельный почтовый индекс для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_CITY** – отдельный город для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_COUNTRY** – отдельная страна для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_STATE** – отдельная область в стране покупателя для доставки заказа;
- **ORDER\_ID** – номер заказа;
- **ORDER\_SHIP\_COST** – стоимость доставки заказа;
- **ORDER\_TOTAL\_SUM** – сумма заказа с учетом стоимости доставки;
- **ORDER\_QUANTITY** – кол-во заказанных продуктов;
- **ORDER\_SHIP\_METHOD** – метод доставки заказа покупателю;
- **ORDER\_PAY\_METHOD** – метод оплаты заказа;
- **ORDER\_PRODUCTS** – массив с информацией о заказанных продуктах. Каждый его элемент должен являться массивом и содержать следующие ключи:
  - **CATEGORY** – название категории продукта;
  - **CAPTION** – название продукта;
  - **SKU** – артикул продукта;
  - **PRICE** – цена одного экземпляра продукта;
  - **QUANTITY** – кол-во заказанных экземпляров продукта;
  - **EGOOD** – принимает значение **Y**, если товар является цифровым (скачиваемым). Иначе должен иметь значение **N**.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_admin\_product\_review\_notify

Назначение:

- Отправка письма-оповещения администратору о поступлении нового отзыва о товаре.

Объявление:

- `function func_mail_admin_product_review_notify($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двухбуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **PRODUCT\_CAPTION** – название продукта, отзыв о котором был оставлен;
  - **PRODUCT\_LINK** – ссылка на страницу карточки продукта, отзыв о котором был оставлен;
  - **AUTHOR\_NAME** – имя автора отзыва;
  - **AUTHOR\_EMAIL** – e-mail автора отзыва;
  - **AUTHOR\_RATE** – оценка продукта автором (по шкале от 1 до 5);
  - **AUTHOR\_COMMENT** – комментарий к оценке;
  - **MODERATE\_LINK** – ссылка на раздел модерирования отзывов о продуктах в админке.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_admin\_shop\_review\_notify

Назначение:

- Отправка письма-оповещения администратору о поступлении нового отзыва о магазине (в гостевой книге).

Объявление:

- `function func_mail_admin_shop_review_notify($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двухбуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;

- **from\_name** – имя отправителя;
- **to** – e-mail адрес получателя;
- **SHOP\_NAME** – название магазина;
- **AUTHOR\_NAME** – имя автора отзыва;
- **AUTHOR\_EMAIL** – e-mail автора отзыва;
- **AUTHOR\_COMMENT** – текст отзыва;
- **MODERATE\_LINK** – ссылка на раздел модерирования отзывов о магазине в админке.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_confirm\_news

Назначение:

- Отправка письма подписчику на новости со ссылкой на подтверждение подписки.

Объявление:

- `function func_mail_confirm_news($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **CONFIRM\_URL** – ссылка на подтверждение подписки на новости.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_customer\_free\_egood\_link

Назначение:

- Отправка пользователю письма со ссылкой для скачивания файла бесплатного цифрового продукта.

Объявление:

- `function func_mail_customer_free_egood_link($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **PRODUCT\_CAPTION** – название бесплатного цифрового продукта;
  - **EGOOD\_LINK** – ссылка на скачивание файла продукта;
  - **EGOOD\_LINK\_TTL** – время действия ссылки в часах;
  - **EGOOD\_LINK\_SPEED** – максимально разрешенная скорость скачивания файла в Кбайт/с.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_fail\_order

Назначение:

- Отправка письма покупателю с деталями заказа, который не удалось обработать в автоматическом режиме платежной системой.

Объявление:

- `function func_mail_fail_order($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **CUSTOMER\_TITLE** – обращение к покупателю (такое, как «уважаемый» или «уважаемая»);
  - **CUSTOMER\_FIRST\_NAME** – имя покупателя;
  - **CUSTOMER\_LAST\_NAME** – фамилия покупателя;
  - **CUSTOMER\_ADDRESS** – строка адреса (улицы) покупателя;



- **CUSTOMER\_ZIP\_CODE** – почтовый индекс покупателя;
- **CUSTOMER\_CITY** – город покупателя;
- **CUSTOMER\_COUNTRY** – страна покупателя;
- **CUSTOMER\_STATE** – область в стране покупателя;
- **CUSTOMER\_PHONE** – номер(а) телефона покупателя;
- **CUSTOMER\_MOBILE\_PHONE** – номер мобильного телефона покупателя;
- **CUSTOMER\_FAX** – номер факса покупателя;
- **CUSTOMER\_WEBSITE** – адрес сайта покупателя;
- **CUSTOMER\_COMPANY** – название компании (организации) покупателя;
- **CUSTOMER\_INN** – ИНН организации покупателя;
- **CUSTOMER\_KPP** – КПП организации покупателя;
- **CUSTOMER\_BANK\_NAME** – наименование банка покупателя;
- **CUSTOMER\_SHIP\_ADDRESS** – строка отдельного адреса (улицы) доставки заказа покупателю;
- **CUSTOMER\_SHIP\_ZIP\_CODE** – отдельный почтовый индекс для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_CITY** – отдельный город для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_COUNTRY** – отдельная страна для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_STATE** – отдельная область в стране покупателя для доставки заказа;
- **ORDER\_ID** – номер заказа;
- **ORDER\_STATUS** – статус заказа;
- **ORDER\_SHIP\_COST** – стоимость доставки заказа;
- **ORDER\_TOTAL\_SUM** – сумма заказа с учетом стоимости доставки;
- **ORDER\_QUANTITY** – кол-во заказанных продуктов;
- **ORDER\_SHIP\_METHOD** – метод доставки заказа покупателю;
- **ORDER\_PAY\_METHOD** – метод оплаты заказа;
- **ORDER\_PRODUCTS** – массив с информацией о заказанных продуктах. Каждый его элемент должен являться массивом и содержать следующие ключи:
  - **CATEGORY** – название категории продукта;
  - **CAPTION** – название продукта;
  - **SKU** – артикул продукта;
  - **PRICE** – цена одного экземпляра продукта;
  - **QUANTITY** – кол-во заказанных экземпляров продукта;
  - **EGOOD** – принимает значение **Y**, если товар является цифровым (скачиваемым). Иначе должен иметь значение **N**.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_news

Назначение:

- Отправка письма с текстом новости подписчику.

Объявление:

- `function func_mail_news($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двухбуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **FACT\_CAPTION** – заголовок новости;
  - **FACT\_TEXT** – текст новости;
  - **UNSUBSCRIBE\_URL** – ссылка для отписки от рассылки новостей.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_newstatus\_order

Назначение:

- Отправка письма-уведомления покупателю о том, что статус его заказа изменился.

Объявление:

- `function func_mail_newstatus_order($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двухбуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **CUSTOMER\_TITLE** – обращение к покупателю (такое, как «уважаемый» или «уважаемая»);

- **CUSTOMER\_FIRST\_NAME** – имя покупателя;
- **CUSTOMER\_LAST\_NAME** – фамилия покупателя;
- **CUSTOMER\_ADDRESS** – строка адреса (улицы) покупателя;
- **CUSTOMER\_ZIP\_CODE** – почтовый индекс покупателя;
- **CUSTOMER\_CITY** – город покупателя;
- **CUSTOMER\_COUNTRY** – страна покупателя;
- **CUSTOMER\_STATE** – область в стране покупателя;
- **CUSTOMER\_PHONE** – номер(а) телефона покупателя;
- **CUSTOMER\_MOBILE\_PHONE** – номер мобильного телефона покупателя;
- **CUSTOMER\_FAX** – номер факса покупателя;
- **CUSTOMER\_WEBSITE** – адрес сайта покупателя;
- **CUSTOMER\_COMPANY** – название компании (организации) покупателя;
- **CUSTOMER\_INN** – ИНН организации покупателя;
- **CUSTOMER\_KPP** – КПП организации покупателя;
- **CUSTOMER\_BANK\_NAME** – наименование банка покупателя;
- **CUSTOMER\_SHIP\_ADDRESS** – строка отдельного адреса (улицы) доставки заказа покупателю;
- **CUSTOMER\_SHIP\_ZIP\_CODE** – отдельный почтовый индекс для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_CITY** – отдельный город для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_COUNTRY** – отдельная страна для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_STATE** – отдельная область в стране покупателя для доставки заказа;
- **ORDER\_ID** – номер заказа;
- **ORDER\_STATUS** – текущий статус заказа;
- **ORDER\_SHIP\_COST** – стоимость доставки заказа;
- **ORDER\_TOTAL\_SUM** – сумма заказа с учетом стоимости доставки;
- **ORDER\_QUANTITY** – кол-во заказанных продуктов;
- **ORDER\_SHIP\_METHOD** – метод доставки заказа покупателю;
- **ORDER\_PAY\_METHOD** – метод оплаты заказа;
- **ORDER\_PRODUCTS** – массив с информацией о заказанных продуктах. Каждый его элемент должен являться массивом и содержать следующие ключи:
  - **CATEGORY** – название категории продукта;
  - **CAPTION** – название продукта;
  - **SKU** – артикул продукта;
  - **PRICE** – цена одного экземпляра продукта;
  - **QUANTITY** – кол-во заказанных экземпляров продукта;
  - **EGOOD** – принимает значение **Y**, если товар является цифровым (скачиваемым). Иначе должен иметь значение **N**;

- **DOWNLOAD** – если **EGOOD** равен **Y**, то значением данного параметра должна быть ссылка на скачивание файла цифрового продукта, если статус заказа был изменен на «Оплачен» или «Завершен».
- **EGOODS\_EXIST** – должен быть равен **true**, если в заказе есть цифровые продукты;
- **EGOODS\_LINKS\_TTL** – время действия ссылок на скачивание файлов цифровых продуктов, в часах.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается **true**. Иначе - возвращается **false** и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_partner\_notify

Назначение:

- Отправка письма-оповещения партнеру магазина (по партнерской программе) о том, что на его лицевой счет зачислена определенная сумма.

Объявление:

- `function func_mail_partner_notify($info)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **SUM** – размер партнерского вознаграждения.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается **true**. Иначе - возвращается **false** и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_pwd\_recover

Назначение:

- Отправка пользователю письма со ссылкой на страницу замены пароля, либо письма с новым установленным паролем.

Объявление:

- `function func_mail_pwd_recover($info, $getlink = false)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **CUSTOMER\_EMAIL** – e-mail пользователя, пароль которого изменяется;
  - **CUSTOMER\_NEW\_PASSWORD** – новый пароль пользователя. Данный параметр игнорируется, если значение параметра **\$getlink** равно **true**;
  - **RECOVER\_LINK** – ссылка на секретную страницу изменения пароля пользователя. Данный параметр игнорируется, если значение параметра **\$getlink** равно **false**.
- **bool \$getlink**. Если значение данного параметра равно **true**, в письме отсылается только ссылка на страницу изменения пароля пользователя. Иначе – отсылается новый пароль пользователя (после его установки самим пользователем на секретной странице).

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается **true**. Иначе - возвращается **false** и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_success\_order

Назначение:

- Отправка письма-уведомления покупателю о том, что его заказ успешно принят или о том, что состав его заказа был изменен.

Объявление:

- `function func_mail_success_order($info, $changed = false)`

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **CUSTOMER\_TITLE** – обращение к покупателю (такое, как «уважаемый» или «уважаемая»);

- **CUSTOMER\_FIRST\_NAME** – имя покупателя;
- **CUSTOMER\_LAST\_NAME** – фамилия покупателя;
- **CUSTOMER\_ADDRESS** – строка адреса (улицы) покупателя;
- **CUSTOMER\_ZIP\_CODE** – почтовый индекс покупателя;
- **CUSTOMER\_CITY** – город покупателя;
- **CUSTOMER\_COUNTRY** – страна покупателя;
- **CUSTOMER\_STATE** – область в стране покупателя;
- **CUSTOMER\_PHONE** – номер(а) телефона покупателя;
- **CUSTOMER\_MOBILE\_PHONE** – номер мобильного телефона покупателя;
- **CUSTOMER\_FAX** – номер факса покупателя;
- **CUSTOMER\_WEBSITE** – адрес сайта покупателя;
- **CUSTOMER\_COMPANY** – название компании (организации) покупателя;
- **CUSTOMER\_INN** – ИНН организации покупателя;
- **CUSTOMER\_KPP** – КПП организации покупателя;
- **CUSTOMER\_BANK\_NAME** – наименование банка покупателя;
- **CUSTOMER\_SHIP\_ADDRESS** – строка отдельного адреса (улицы) доставки заказа покупателю;
- **CUSTOMER\_SHIP\_ZIP\_CODE** – отдельный почтовый индекс для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_CITY** – отдельный город для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_COUNTRY** – отдельная страна для доставки заказа покупателю;
- **CUSTOMER\_SHIP\_STATE** – отдельная область в стране покупателя для доставки заказа;
- **ORDER\_ID** – номер заказа;
- **ORDER\_STATUS** – статус заказа;
- **ORDER\_SHIP\_COST** – стоимость доставки заказа;
- **ORDER\_TOTAL\_SUM** – сумма заказа с учетом стоимости доставки;
- **ORDER\_QUANTITY** – кол-во заказанных продуктов;
- **ORDER\_SHIP\_METHOD** – метод доставки заказа покупателю;
- **ORDER\_PAY\_METHOD** – метод оплаты заказа;
- **ORDER\_PAY\_DETAILS** – реквизиты для оплаты заказа;
- **SBERBANK\_RECEIPT\_URL** – ссылка на скачивание квитанции Сбербанка, если метод оплаты – Сбербанк;
- **ORDER\_PRODUCTS** – массив с информацией о заказанных продуктах. Каждый его элемент должен являться массивом и содержать следующие ключи:
  - **CATEGORY** – название категории продукта;
  - **CAPTION** – название продукта;
  - **SKU** – артикул продукта;
  - **PRICE** – цена одного экземпляра продукта;
  - **QUANTITY** – кол-во заказанных экземпляров продукта;

- **EGOOD** – принимает значение **Y**, если товар является цифровым (скачиваемым). Иначе должен иметь значение **N**.
- **bool \$changed**. Если данный параметр равен true, покупателю отсылается письмо с новым составом заказа. Иначе – письмо-уведомление о том, что его заказ успешно принят.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_user\_activated

Назначение:

- Отправка письма-уведомления зарегистрированному покупателю о том, что его аккаунт активирован.

Объявление:

- function func\_mail\_user\_activated(\$info)

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:
  - **short\_lang** – международное двубуквенное обозначение языка, на котором необходимо отправить письмо;
  - **from** – e-mail адрес отправителя;
  - **from\_name** – имя отправителя;
  - **to** – e-mail адрес получателя;
  - **SHOP\_NAME** – название магазина;
  - **SHOP\_EMAIL** – e-mail администратора магазина;
  - **CUSTOMER\_EMAIL** – e-mail зарегистрированного покупателя.

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_mail\_user\_registered

Назначение:

- Отправка письма-уведомления покупателю о том, что его аккаунт зарегистрирован в магазине.

Объявление:

- function func\_mail\_user\_registered(\$info)

Параметры:

- **array \$info**. Массив с информацией, которую необходимо отразить в письме. Должен содержать следующие ключи:

- **short\_lang** – международное двухбуквенное обозначение языка, на котором необходимо отправить письмо;
- **from** – e-mail адрес отправителя;
- **from\_name** – имя отправителя;
- **to** – e-mail адрес получателя;
- **SHOP\_NAME** – название магазина;
- **SHOP\_EMAIL** – e-mail администратора магазина;
- **CUSTOMER\_EMAIL** – e-mail зарегистрированного покупателя;
- **CUSTOMER\_PASSWORD** – пароль зарегистрированного покупателя. Вместо пароля можно отправить строку «Указанный при регистрации»;
- **ACTIVATION\_LINK** – ссылка для активации зарегистрированного аккаунта покупателя (если включена активация при регистрации).

Возвращаемое значение:

- Тип **bool**. Если функция отправки e-mail сработала успешно, возвращается true. Иначе - возвращается false и в глобальный массив [\\$err](#) записывается информация об ошибке.

## func\_msgs\_append

Назначение:

- Добавление сообщения об ошибке или сообщения об успешности операции в сессию для его отображения в браузере с помощью [func\\_msgs\\_process](#).

Объявление:

- `function func_msgs_append($err, $msg, $at_begin = false)`

Параметры:

- **string \$err**. Текст сообщения об ошибке.
- **string \$msg**. Текст сообщения об успешности операции.
- **bool \$at\_begin**. Если значение данного параметра равно **true**, текст сообщения будет добавлен в начало соответствующего массива.

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_msgs\_assign

Назначение:

- Конвертация глобальных массивов [\\$err](#) и [\\$msgs](#) в строки и запись их значений в соответствующие глобальные переменные **\$err** и **\$msgs** в Smarty для отображения сообщений об ошибках (или об успешности операции) в браузере.

Объявление:

- `function func_msgs_assign($err, $msgs)`

Параметры:

- **array \$err**. Значение глобального массива [\\$err](#).



- **array \$msgs**. Значение глобального массива [\\$msgs](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_msgs\_create

Назначение:

- Сохранение сообщений об ошибках (или об успешности операции) в сессии для последующего их вывода в браузер с помощью [func\\_msgs\\_process](#).

Объявление:

- `function func_msgs_create($err, $msgs)`

Параметры:

- **array \$err**. Значение глобального массива [\\$err](#).
- **array \$msgs**. Значение глобального массива [\\$msgs](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_msgs\_process

Назначение:

- Вывод сообщений об ошибках (или об успешности операции), подготовленных с помощью [func\\_msgs\\_create](#) или [func\\_msgs\\_append](#), в браузер с помощью [func\\_msgs\\_assign](#).

Объявление:

- `function func_msgs_process()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_price\_correct

Назначение:

- Коррекция (и проверка на корректность) указанной суммы перед ее помещением в БД.

Объявление:

- `function func_price_correct($price)`

Параметры:

- **string \$price**. Сумма, значение которой необходимо проверить и откорректировать.

Возвращаемое значение:

- Тип **float/bool**. Откорректированное значение суммы или `false`, если сумма задана в некорректном формате или параметр **\$price** не является числом.

## func\_save\_watermarked\_png

Назначение:

- Наложение водяного знака в формате PNG на заданное изображение и сохранение результата в виде файла (или выдача JPEG-результата в браузер).

Объявление:

- `function func_save_watermarked_png($src, $dest, $watermark, $alpha, $png_type)`

Параметры:

- **string \$src**. Путь к исходному изображению, на которое необходимо наложить водяной знак.
- **string \$dest**. Путь к результирующему изображению. Если данный параметр равен **null**, результат не сохраняется на диске, а выдается в браузер.
- **string \$watermark**. Путь к PNG-файлу изображения водяного знака.
- **int \$alpha**. Уровень прозрачности водяного знака (чем больше, тем прозрачнее). Принимает значения по шкале от 0 до 100.
- **string \$png\_type**. Тип PNG-изображения водяного знака. Может принимать значения **png8** или **png24**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции. Функция ничего не возвращает, если параметр **\$dest** не задан (равен **null**).

## func\_save\_watermarked\_text

Назначение:

- Наложение заданного текста поверх заданного изображения по диагонали в виде водяного знака.

Объявление:

- `function func_save_watermarked_text($src, $dest, $text, $text_color, $alpha)`

Параметры:

- **string \$src**. Путь к исходному изображению, на которое необходимо наложить водяной знак.
- **string \$dest**. Путь к результирующему изображению. Если данный параметр равен **null**, результат не сохраняется на диске, а выдается в браузер.
- **string \$text**. Строка текста, которую необходимо наложить на изображение в качестве водяного знака.
- **int \$text\_color**. RGB-значение цвета накладываемого текста.
- **int \$alpha**. Уровень прозрачности накладываемого текста (чем больше, тем прозрачнее). Принимает значения по шкале от 0 до 100.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции. Функция ничего не возвращает, если параметр **\$dest** не задан (равен **null**).

## func\_send\_mail

Назначение:

- Отправка E-Mail указанному адресату от указанного отправителя. Письмо отправляется в кодировке UTF-8 и в формате HTML. Для отправки используется библиотека **PHPMailer**.

Объявление:

- `function func_send_mail($language, $from, $from_name, $to, $subject, $body, &$error_text)`

Параметры:

- **string \$language**. Международное двухбуквенное обозначение языка, на котором будут возвращены сообщения об ошибках в переменной **\$error\_text**, если таковые возникнут при отправке.
- **string \$from**. E-Mail адрес отправителя.
- **string \$from\_name**. Имя отправителя.
- **string \$to**. E-Mail адрес получателя.
- **string \$subject**. Тема письма в кодировке UTF-8.
- **string \$body**. Тело письма в кодировке UTF-8. Может содержать HTML.
- **string \$error\_text**. Переменная, в которую будут записаны сообщения об ошибках, если таковые возникнут при отправке.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции. Если при отправке возникла ошибка, она будет зафиксирована в переменной **\$error\_text**.

## func\_shop\_err\_handler

Назначение:

- Функция-обработчик ошибок, вызываемая при возникновении какой-либо ошибки в коде PHP или в коде Smarty. Данная функция сохраняет текст ошибки в лог ошибок с помощью класса [ErrorLogger](#). Функция вызывается автоматически с помощью **set\_error\_handler**, поэтому нет необходимости вызывать ее вручную. Следует иметь в виду, что данная функция не отлавливает ошибки в ядре магазина и не отлавливает **fatal errors**.

Объявление:

- `function func_shop_err_handler($errno, $errormsg, $filename, $linenum)`

Параметры:

- **int \$errno**. Номер возникшей ошибки.
- **string \$errormsg**. Текст ошибки.
- **string \$filename**. Имя файла (полный путь к нему), в котором возникла ошибка.
- **string \$linenum**. Номер строки в файле, на которой возникла ошибка.

Возвращаемое значение:

- Функция ничего не возвращает.

## func\_stripslashes

Назначение:

- Рекурсивный вызов **stripslashes** для указанной переменной-массива. Переменная не обязательно должна быть массивом. В этом случае происходит обычный вызов **stripslashes**.

Объявление:

- `function func_stripslashes(&$el)`

Параметры:

- **string/array \$el**. Переменная, которую необходимо обработать с помощью **stripslashes**. Если является массивом, функция обрабатывает его рекурсивно.

Возвращаемое значение:

- Функция ничего не возвращает. Вместо этого результат записывается в ту же самую переменную **\$el**, переданную в функцию в качестве параметра.

## func\_to\_json

Назначение:

- Конвертирование массивов сообщений об ошибках (или об успешности операции), а также любой дополнительной информации, в JSON-представление для использования этой сконвертированной информации в JavaScript-коде при вызове jQuery-функций (таких, как `getJSON` и т.п.). В качестве примера см. раздел про [AJAX-модули](#).

Объявление:

- `function func_to_json($html, $msgs = "", $err = "")`

Параметры:

- **string \$html**. Любая информация, которая возвращается в поле с названием **html** в JSON-представлении.
- **array \$msgs**. Значение глобального массива [\\$msgs](#).
- **array \$err**. Значение глобального массива [\\$err](#).
- Функция может принимать больше аргументов. В этом случае имена последующих аргументов трактуются следующим образом: **arg4**, **arg5**, **arg6** и т.д.

Возвращаемое значение:

- Тип **string**. JSON-представление информации, полученной в аргументах функции.

## func\_wrap\_quotes

Назначение:

- Добавление к указанной строке одиночных кавычек в начале и в конце. Функция применяется в методах [классов](#) CMS для «оковывивания» строковых значений перед их помещением в SQL-запрос. Данная функция может не применяться (вместо этого можно вручную проставлять кавычки, где это необходимо) и оставлена из исторических соображений (т.к. в некоторых местах кода используется).

Объявление:

- `function func_wrap_quotes($str)`

Параметры:

- **string \$str**. Строка, которую необходимо взять в кавычки.

Возвращаемое значение:

- Тип **string**. «Оковыченная» строка **\$str**.

## prod\_options\_variants\_price

Назначение:

- Изменение информации о продукте в соответствии с комбинацией опций по умолчанию. Используется только на витрине. Функция принимает в качестве параметра массив с базовой информацией о продукте из таблицы [proshop\\_products](#) и изменяет эту информацию в соответствии с опциями продукта (если они заданы и если модуль опций продукта включен и включена соответствующая настройка в магазине (показ варианта продукта по умолчанию в списке продуктов)) – цена, артикул, вес и т.п.

Объявление:

- `function prod_options_variants_price($product)`

Параметры:

- **array \$product**. Массив с базовой информацией о продукте из таблицы [proshop\\_products](#). Ключи данного массива должны совпадать с полями этой таблицы.

Возвращаемое значение:

- Тип **array**. Измененный (или не измененный) массив **\$product**.

## prods\_process

Назначение:

- Формирование списка продуктов для его вывода в Smarty-шаблон списка на витрине. Данная функция используется на витрине для формирования цен, величин налогов и скидок продуктов в списке, а также их отображаемых по умолчанию вариантов (если модуль опций продуктов включен). При формировании любого списка продуктов на витрине необходимо делать это именно через данную функцию.

Объявление:

- `function prods_process($products, $assign = 'products')`

Параметры:

- **array \$products**. Массив продуктов, полученный с помощью [Products::Search](#).
- **string \$assign**. Имя Smarty-переменной, которая будет содержать подготовленный для вывода список продуктов.

Возвращаемое значение:

- Тип **array**. Измененный (или не измененный) массив **\$products**.

## 8.5 Функции (JavaScript)

JavaScript-функции, представленные в данном разделе, располагаются в файлах `templates/admin/_system_/js/utils.js`, `templates/admin/_system_/confirm.html` и/или `templates/customer/[тема_оформления]/_system_/js/utils.js`. Каждая из функций может как присутствовать, так и отсутствовать для той или иной темы оформления.

### BannerClicked

Назначение:

- Фиксация нажатия пользователем на баннер – передача специальному скрипту информации о нажатии для ее записи в БД.

Зона применения:

- Витрина.

Объявление:

- `function BannerClicked(id)`

Параметры:

- `int id`. Идентификатор баннера из таблицы [proshop\\_banners](#).

Возвращаемое значение:

- Функция ничего не возвращает.

### ClearMsgs

Назначение:

- Очистка окна сообщений об ошибках (или об успешности операции).

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function ClearMsgs()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип `bool`. Всегда `false`.

### ConfirmDialog

Назначение:

- Вывод окна подтверждения операции.

Зона применения:

- Витрина.

Объявление:

- `function ConfirmDialog(title, text, on_yes_button, on_no_button)`

Параметры:

- **string title.** Заголовок окна подтверждения.
- **string text.** Текст окна подтверждения. Может содержать HTML-код.
- **function on\_yes\_button.** JavaScript-функция обратного вызова, срабатывающая при нажатии кнопки «Да» окна подтверждения.
- **function on\_no\_button.** JavaScript-функция обратного вызова, срабатывающая при нажатии кнопки «Нет» окна подтверждения.

Возвращаемое значение:

- Функция ничего не возвращает.

## ConfirmSubmitForm

Назначение:

- Вывод окна подтверждения передачи (сабмита) указанной формы и сабмит формы при положительном ответе пользователя.

Зона применения:

- Админка.

Объявление:

- `function ConfirmSubmitForm(form_name, _confirm)`

Параметры:

- **string form\_name.** Имя формы, которую следует передать (сделать ее сабмит).
- **string \_confirm.** Текст сообщения, который необходимо вывести пользователю в браузер в окне подтверждения.

Возвращаемое значение:

- Тип **bool**. Всегда `false`.

## CorrectUrlName

Назначение:

- Корректировка имени в URL (для `modRewrite`). JavaScript-аналог PHP-функции [func\\_correct\\_engname](#).

Зона применения:

- Админка.

Объявление:

- `function CorrectUrlName(url_name)`

Параметры:

- **string url\_name.** Имя в URL (для `modRewrite`).

Возвращаемое значение:

- Тип **string**. Откорректированное значение `url_name`.

## dec2hex

Назначение:

- Конвертирование десятичного целого числа в его шестнадцатеричное представление.

Зона применения:

- Витрина.

Объявление:

- `function dec2hex(n)`

Параметры:

- **int n**. Десятичное целое число.

Возвращаемое значение:

- Тип **string**. Нех-представление заданного числа.

## getRGB

Назначение:

- Получение шестнадцатеричного представления значения цвета по его строковому значению из таблицы стилей CSS (например, из строки `rgb(255,255,255)` или `#FFFFFF`).

Зона применения:

- Витрина.

Объявление:

- `function getRGB(color)`

Параметры:

- **string color**. Значение цвета из CSS.

Возвращаемое значение:

- Тип **string**. Нех-представление заданного цвета (например, `FEFEFE`).

## GetSelect

Назначение:

- Определение значения текущего выбранного элемента в `select`-боксе на HTML-странице.

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function GetSelect(id)`

Параметры:

- **int id**. Идентификатор тэга `<select>`, текущее выбранное значение которого необходимо определить.

Возвращаемое значение:

- Тип **string**. Значение выбранного элемента или `null`, если ничего не выбрано.



## HideFastAuxPanel

Назначение:

- Скрытие плавающей панели быстрого доступа, отображенной с помощью функции [ShowFastAuxPanel](#).

Зона применения:

- Админка.

Объявление:

- `function HideFastAuxPanel()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## isCookieEnabled

Назначение:

- Определение, включены ли cookie в браузере.

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function isCookieEnabled()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## ProcessMsgs

Назначение:

- Отображение окна сообщений об ошибках (или об успешности операции). Если заданы (не пусты) одновременно два входных параметра функции, отображаются только сообщения об ошибках (сообщения об успешности операции не отображаются).

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function ProcessMsgs(msg_txt, err_txt)`

Параметры:

- **string msg\_txt.** Текст сообщений об успешности операции, закодированный алгоритмом Base 64.
- **string err\_txt.** Текст сообщений об ошибках, закодированный алгоритмом Base 64.

Возвращаемое значение:

- Тип **bool**. Всегда false.

## ReloadCaptcha

Назначение:

- Перезагрузка всех изображений капчи на текущей странице магазина. Функция ищет на странице все тэги **img** с атрибутом **class**, равным **captcha**, и заменяет атрибут **src** на другое изображения. Перезагрузка изображений капчи используется в тех случаях, когда пользователь не может прочитать комбинацию букв (и/или цифр) и нажимает на кнопку перезагрузки изображения.

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function ReloadCaptcha()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## SelectAllToDel

Назначение:

- Пометка всех галочек с атрибутом **rel**, равным **rel\_for\_del**, на текущей странице (если параметром **rel** не задано иное значение этого атрибута). Функция ищет все тэги **input type=checkbox** на текущей странице, имеющие атрибут **rel**, равный **rel\_for\_del** (или равный параметру **rel**, если он задан), и отмечает найденные галочки.

Зона применения:

- Админка.

Объявление:

- `function SelectAllToDel(rel)`

Параметры:

- **string rel.** Необязательный параметр. Если данный параметр не задан, то считается, что его значение равно **rel\_for\_del**.

Возвращаемое значение:

- Функция ничего не возвращает.

## SelectAllToVis

Назначение:

- Пометка всех галочек с атрибутом **rel**, равным **rel\_for\_vis**, на текущей странице. Функция ищет все теги **input type=checkbox** на текущей странице, имеющие атрибут **rel**, равный **rel\_for\_vis**, и отмечает найденные галочки.

Зона применения:

- Админка.

Объявление:

- `function SelectAllToVis()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## set\_active\_tab

Назначение:

- Установка в памяти текущей активной вкладки для указанного модуля.

Зона применения:

- Админка.

Объявление:

- `function set_active_tab(module, tab_num, sync)`

Параметры:

- **string module**. Служебное название модуля, в котором необходимо запомнить номер текущей активной вкладки в пределах текущей сессии.
- **int tab\_num**. Номер вкладки (начиная от 1).
- **bool sync**. Если данный параметр равен **true**, функция вернет управление только после того, как запрос на сохранение отправится на сервер, а сервер в свою очередь ответит. Иначе – функция вернет управление сразу после отправки запроса, не дожидаясь ответа сервера.

Возвращаемое значение:

- Функция ничего не возвращает.

## ShopHideSelectCat

Назначение:

- Скрытие окна выбора категории (в админке), отображенного с помощью функции [ShopSelectCategory](#).

Зона применения:

- Админка.

Объявление:

- `function ShopHideSelectCat()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## ShopSelectCat

Назначение:

- Данная функция предназначена для внутреннего использования – используется в Smarty-шаблоне вывода списка категорий для окна выбора категории. Является функцией обратного вызова, вызываемой в том случае, когда пользователь нажимает на ссылку выбора одной из категорий в окне выбора категории.

Зона применения:

- Админка.

Объявление:

- `function ShopSelectCat(cat_id, cat_caption)`

Параметры:

- **int cat\_id**. Идентификатор выбранной категории.
- **string cat\_caption**. Название выбранной категории.

Возвращаемое значение:

- Функция ничего не возвращает.

## ShopSelectCategory

Назначение:

- Вывод окна выбора категории (в админке).

Зона применения:

- Админка.

Объявление:

- `function ShopSelectCategory(title, cat_id, exclude, allow_select_all, allow_select_root, callback)`

Параметры:

- **string title**. Заголовок окна выбора категории.
- **int cat\_id**. Идентификатор выбранной в данный момент категории (для разворачивания дерева категорий).
- **int exclude**. Идентификатор категории, которую нужно исключить из дерева категорий при его выводе. Если этот параметр равен 0, ни одна категория не исключается.
- **bool allow\_select\_all**. Если данный параметр равен **true**, в окне выбора категорий будет отображена кнопка «Выбрать все».
- **bool allow\_select\_root**. Если данный параметр равен **true**, в окне выбора категорий будет отображена кнопка «Выбрать корневую».

- **function callback.** JavaScript-функция обратного вызова, вызываемая после того, как пользователь выберет категорию. Прототип данной функции выглядит след. образом:
  - **function(cat\_id, cat\_caption),** где:
    - **int cat\_id.** Идентификатор выбранной категории.
    - **string cat\_caption.** Название выбранной категории.

Возвращаемое значение:

- Функция ничего не возвращает.

## ShopShowSubcats

Назначение:

- Данная функция предназначена для внутреннего использования - используется функцией [ShopSelectCategory](#) и не должна вызываться сама по себе (программистом). Предназначена для вывода списка подкатегорий указанной категории в окне выбора категорий.

Зона применения:

- Админка.

Объявление:

- `function ShopShowSubcats(cat_id, selected_id, exclude, level)`

Параметры:

- **int cat\_id.** Идентификатор категории, список подкатегорий которой необходимо отобразить в окне выбора категории.
- **int selected\_id.** Идентификатор категории, которую нужно выделить в списке.
- **int exclude.** Идентификатор категории, которую не следует включать в выводимый список подкатегорий.
- **int level.** Уровень вложенности.

Возвращаемое значение:

- Функция ничего не возвращает.

## ShowBlockUI

Назначение:

- Блокировка интерфейса пользователя при выполнении какой-либо операции и вывод в окно браузера сообщения «Подождите, операция выполняется...». Заблокированный интерфейс можно разблокировать вызовом **\$j.unblockUI()** или путем перезагрузки страницы.

Зона применения:

- Админка.

Объявление:

- `function ShowBlockUI()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## ShowFastAuxPanel

Назначение:

- Отображение полупрозрачной плавающей панели быстрого доступа в админке (внизу страницы).

Зона применения:

- Админка.

Объявление:

- `function ShowFastAuxPanel(html)`

Параметры:

- **string html**. HTML-код, который необходимо отобразить на панели (например, код какой-нибудь кнопки).

Возвращаемое значение:

- Функция ничего не возвращает.

## ShowLoading

Назначение:

- Отображение анимированного (GIF) индикатора выполнения длительной операции (обычно, загрузки страницы или файла) в указанном месте.

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function ShowLoading(id, label)` – для админки
- `function ShowLoading(id, alt, label, centered)` – для витрины

Параметры:

- **int id**. Идентификатор HTML-элемента (обычно это тэг DIV), в котором следует отобразить индикатор выполнения работы.
- **string alt**. Альтернативный текст, отображаемый вместо изображения индикатора в том случае, если отображение картинок не включено в браузере. Этот параметр задавать не обязательно.
- **string label**. Подпись справа от изображения индикатора (например, «Пожалуйста, подождите»). Этот параметр задавать не обязательно.
- **bool centered**. Если значение данного параметра равно true, индикатор будет помещен в DIV, который центрирует свое содержимое. Это необходимо в тех случаях, когда требуется, чтобы индикатор отображался по центру какого-либо блока или страницы.

Возвращаемое значение:

- Функция ничего не возвращает.

## SubmitAction

Назначение:

- Вывод окна подтверждения выполнения заданного JavaScript-кода при положительном ответе пользователя.

Зона применения:

- Админка.

Объявление:

- `function SubmitAction(action, _confirm)`

Параметры:

- **function action**. JavaScript-функция, которую необходимо выполнить. См. пример.
- **string \_confirm**. Текст сообщения, который необходимо вывести пользователю в браузер в окне подтверждения.

Возвращаемое значение:

- Тип **bool**. Всегда `false`.

Пример:

- `<a href="javascript:void(0);" onClick="SubmitAction(function() { location.href = 'index.php?module=banners'; }, 'Вы уверены?'); return false;">Перейти</a>`

## SubmitForm

Назначение:

- Передача формы с подтверждением (или без подтверждения) со стороны пользователя. Данная функция принимает во внимание настройки админки. Если в настройках подтверждения передачи форм отключены, функция сразу будет делать сабмит формы без вывода окна подтверждения.

Зона применения:

- Админка.

Объявление:

- `function SubmitForm(form_name, _confirm)`

Параметры:

- **string form\_name**. Имя формы, которую следует передать (сделать ее сабмит).
- **string \_confirm**. Текст сообщения, который необходимо вывести пользователю в браузер в окне подтверждения.

Возвращаемое значение:

- Тип **bool**. Всегда `false`.

## Translit

Назначение:

- Транслитерация заданного русского текста на английский вариант.

Зона применения:

- Админка.

Объявление:

- `function Translit(text)`

Параметры:

- **string text**. Текст на русском языке, который нужно транслитерировать.

Возвращаемое значение:

- Тип **string**. Транслитерированный вариант заданного текста.

## trim

Назначение:

- Удаление ведущих и конечных пробелов из заданной строки.

Зона применения:

- Админка.
- Витрина.

Объявление:

- `function trim(s)`

Параметры:

- **string s**. Строка, из которой необходимо убрать окружающие ее пробелы.

Возвращаемое значение:

- Тип **string**. Обработанная строка.

## UnSelectAllToDel

Назначение:

- Снятие пометок со всех галочек с атрибутом **rel**, равным **rel\_for\_del**, на текущей странице (если параметром **rel** не задано иное значение этого атрибута). Функция ищет все теги **input type=checkbox** на текущей странице, имеющие атрибут **rel**, равный **rel\_for\_del** (или равный параметру **rel**, если он задан), и снимает найденные галочки.

Зона применения:

- Админка.

Объявление:

- `function UnSelectAllToDel(rel)`

Параметры:

- **string rel**. Необязательный параметр. Если данный параметр не задан, то считается, что его значение равно **rel\_for\_del**.

Возвращаемое значение:

- Функция ничего не возвращает.



## UnSelectAllToVis

Назначение:

- Снятие пометок со всех галочек с атрибутом **rel**, равным **rel\_for\_vis**, на текущей странице. Функция ищет все тэги **input type=checkbox** на текущей странице, имеющие атрибут **rel**, равный **rel\_for\_vis**, и снимает найденные галочки.

Зона применения:

- Админка.

Объявление:

- `function UnSelectAllToVis()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## 8.6 Классы и их методы

Методы некоторых (большинства) классов должны вызываться, как статические (т.е. с нотацией `::`). Для остальных классов существуют глобальные объекты в виде глобальных переменных (например, глобальная переменная `$CACHE`) – методы таких классов должны вызываться с нотацией `->` (например, методы класса `Cache`).

### Класс Access

Предназначен для управления черным списком IP-адресов из модуля **access**.

#### **Access** : : **AddIP**

Назначение:

- Добавление IP-адреса в черный список и описания причины его блокировки.

Объявление:

- `function AddIP($info, $lang_id)`

Параметры:

- **array \$info**. Массив информации об IP-адресе. Ключами массива являются следующие элементы:
  - **ip** – IP-адрес (макс. 15 символов);
  - **reason** – причина блокировки (макс. 254 символа);
  - **blocked** – флаг блокировки (**Y** – заблокирован, **N** - разблокирован).
- **int \$lang\_id**. Идентификатор языка, на котором указывается причина блокировки.

Возвращаемое значение:

- Тип **bool**. Возвращает true/false в зависимости от успешности операции.

## **Access::DeleteIP**

Назначение:

- Удаление IP-адреса по его идентификатору из черного списка.

Объявление:

- `function DeleteIP($id)`

Параметры:

- **int \$id**. Идентификатор удаляемого IP-адреса.

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Access::GetAllIP**

Назначение:

- Извлечение черного списка IP-адресов из БД.

Объявление:

- `function GetAllIP($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, на котором требуется прочитать причину блокировки каждого из извлеченных IP-адресов.

Возвращаемое значение:

- Тип **array/bool**. False в случае ошибки, иначе - массив, каждый элемент которого содержит следующие ключи:
  - все поля таблицы [proshop\\_access\\_ip](#);
  - поле **reason** таблицы [proshop\\_access\\_ip\\_texts](#);

Пример:

- ```
$rows = Access::GetAllIP($SHOP_LANGUAGE['id']);
foreach ($rows as $row) {
    // $row['id'] – идентификатор IP-адреса
    // $row['ip'] – IP-адрес
    // ...
}
```

## **Access::GetIPReason**

Назначение:

- Извлечение причины блокировки для указанного IP-адреса на указанном языке.

Объявление:

- `function GetIPReason($ip, $lang_id)`

Параметры:

- **string \$ip**. IP-адрес.
- **int \$lang\_id**. Идентификатор языка, для которого требуется извлечь причину блокировки.

Возвращаемое значение:

- Тип **string/bool**. Текстовое описание причины блокировки или false в случае ошибки.

## **Access : : IPTold**

Назначение:

- Извлечение идентификатора IP-адреса из черного списка по самому адресу.

Объявление:

- function IPTold(\$ip)

Параметры:

- **string \$ip**. IP-адрес.

Возвращаемое значение:

- Тип **int/bool**. Идентификатор IP-адреса или false в случае ошибки.

## **Access : : IsRestrictedByIP**

Назначение:

- Определение, заблокирован ли указанный IP-адрес.

Объявление:

- function IsRestrictedByIP(\$ip)

Параметры:

- **string \$ip**. IP-адрес.

Возвращаемое значение:

- Тип **bool**. True, если IP-адрес заблокирован. Иначе – false.

## **Access : : UpdateIP**

Назначение:

- Обновление информации об IP-адресе в черном списке.

Объявление:

- function UpdateIP(\$id, \$lang\_id, \$info)

Параметры:

- **int \$id**. Идентификатор IP-адреса.
- **int \$lang\_id**. Идентификатор языка, для которого требуется обновить причину блокировки.
- **array \$info**. Массив информации об IP-адресе. Ключами массива являются следующие элементы:
  - **reason** – причина блокировки (макс. 254 символа);
  - **blocked** – флаг блокировки (**Y** – заблокирован, **N** - разблокирован).

Возвращаемое значение:

- Тип **bool**. True в случае успеха, false – в случае ошибки.

## Класс AdminGroups

Предназначен для управления группами администраторов. Используется модулем **users**.

### **AdminGroups::Add**

Назначение:

- Добавление группы администраторов в БД.

Объявление:

- `function Add($name, $dmodules, $actions)`

Параметры:

- **string \$name**. Наименование группы администраторов. Не зависит от языка. Макс. длина – 64 символа.
- **array \$dmodules**. Массив идентификаторов запрещенных для данной группы модулей (идентификаторы модулей хранятся в таблице [proshop\\_modules](#)).
- **array \$actions**. Массив идентификаторов разрешенных действий для данной группы. Действия модулей хранятся в таблице [proshop\\_modules\\_admin\\_actions](#).

Возвращаемое значение:

- Тип **int/bool**. Идентификатор новой группы администраторов в случае успеха, `false` – в случае ошибки.

### **AdminGroups::Delete**

Назначение:

- Удаление группы администраторов из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор группы администраторов.

Возвращаемое значение:

- Тип **bool**. Всегда `true`.

### **AdminGroups::GetAll**

Назначение:

- Извлечение информации о каждой группе администраторов из БД.

Объявление:

- `function GetAll()`

Параметры:

- функция не имеет параметров;

Возвращаемое значение:

- Тип **array**. Массив групп администраторов (отсортированный по наименованию группы), каждый элемент которого содержит все поля таблицы [proshop\\_admins\\_groups](#).

Пример:

- ```
$rows = AdminGroups::GetAll();
foreach ($rows as $row) {
    // $row['id'] – идентификатор группы
    // $row['name'] – наименование группы
}
```

## **AdminGroups::GetOne**

Назначение:

- Извлечение информации об указанной группе администраторов;

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор группы.

Возвращаемое значение:

- Тип **array**. Массив со следующими ключами:
  - **name** – наименование группы администраторов;
  - **dmodules** – массив идентификаторов запрещенных модулей (список модулей хранится в таблице [proshop\\_modules](#));
  - **eactions** – массив идентификаторов разрешенных действий (действия модулей хранятся в таблице [proshop\\_modules\\_admin\\_actions](#)).

## **AdminGroups::Update**

Назначение:

- Обновление информации о группе администраторов в БД.

Объявление:

- `function Update($id, $name, $dmodules, $eactions)`

Параметры:

- **int \$id**. Идентификатор группы.
- **string \$name**. Наименование группы. Макс. 64 символа.
- **array \$dmodules**. Массив идентификаторов запрещенных для данной группы модулей (идентификаторы модулей хранятся в таблице [proshop\\_modules](#)).
- **array \$eactions**. Массив идентификаторов разрешенных действий для данной группы. Действия модулей хранятся в таблице [proshop\\_modules\\_admin\\_actions](#).

Возвращаемое значение:

- Тип **bool**. True/false в зависимости от успешности операции.

## Класс AdminMenus

Предназначен для работы меню в административной части CMS.

### **AdminMenus::GetAll**

Назначение:

- Формирование списка блоков меню, которые присутствуют в админке.

Объявление:

- `function GetAll()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив блоков меню админки с соответствующими именами (см. пример), которые хранятся в таблице [proshop\\_admin\\_menus](#).

Пример:

- ```
$rows = AdminMenus::GetAll();
foreach ($rows as $row) {
    // $row['name'] – служебное имя блока
    // $row['pos'] – позиция блока относительно остальных
    // $row['label'] – название текстовой метки блока из таблицы proshop\_labels
}
```

## Класс AdvLinks

Предназначен для управления ссылками с помощью модуля **adv\_links**.

### **AdvLinks::AddRoot**

Назначение:

- Добавление базовой информации о ссылке в БД.

Объявление:

- `function AddRoot($url, $new_window)`

Параметры:

- **string \$url**. URL-адрес ссылки. Макс. 254 символа.
- **char \$new\_window**. Флаг, определяющий, будет ли страница ссылки при нажатии открыта в новом окне (Y – будет открыта в новом окне или в новой вкладке, N – будет открыта в том же окне).

Возвращаемое значение:

- Тип **int/bool**. `False` – в случае ошибки. Иначе – идентификатор созданной ссылки.

## **AdvLinks::AddText**

Назначение:

- Добавление названия указанной ссылки для указанного языка.

Объявление:

- `function AddText($id, $lang_id, $title)`

Параметры:

- **int \$id**. Идентификатор ссылки, для которой добавляется языкозависимое название на указанном языке.
- **int \$lang\_id**. Идентификатор языка, для которого добавляется название ссылки.
- **string \$title**. Название ссылки. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **AdvLinks::Delete**

Назначение:

- Удаление указанной ссылки из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор ссылки.

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **AdvLinks::GetAll**

Назначение:

- Извлечение всех ссылок из БД.

Объявление:

- `GetAll($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого будут извлечены языкозависимые названия ссылок.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом со следующими ключами:
  - **id** – идентификатор ссылки;
  - **url** – адрес ссылки;
  - **pos** – позиция ссылки относительно других ссылок;
  - **new\_window** – флаг открываемости страницы ссылки в новом окне (значение может быть **Y** или **N**);

- **title** – название ссылки.

## **AdvLinks::GetOne**

Назначение:

- Извлечение информации о ссылке из БД.

Объявление:

- function GetOne(\$id, \$lang\_id)

Параметры:

- **int \$id**. Идентификатор ссылки.
- **int \$lang\_id**. Идентификатор языка, для которого будет извлечено языкозависимое название ссылки.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив со следующими ключами:
  - **id** – идентификатор ссылки;
  - **url** – адрес ссылки;
  - **pos** – позиция ссылки относительно других ссылок;
  - **new\_window** – флаг открываемости страницы ссылки в новом окне (значение может быть **Y** или **N**);
  - **title** – название ссылки.

## **AdvLinks::SetPos**

Назначение:

- Установка позиции указанной ссылки относительно других ссылок.

Объявление:

- function SetPos(\$link\_id, \$pos)

Параметры:

- **int \$link\_id**. Идентификатор ссылки.
- **int \$pos**. Позиция ссылки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **AdvLinks::Update**

Назначение:

- Обновление информации в БД об указанной ссылке.

Объявление:

- function Update(\$id, \$lang\_id, \$info)

Параметры:

- **int \$id**. Идентификатор обновляемой ссылки.



- **int \$lang\_id**. Идентификатор языка, для которого будет установлено название ссылки в массиве **\$info**.
- **array \$info**. Массив информации о ссылке, содержащий в себе следующие ключи:
  - **url** – адрес ссылки. Макс. 254 символа.
  - **title** – название ссылки для указанного языка. Макс. 254 символа.
  - **new\_window** - флаг открываемости страницы ссылки в новом окне (значение может быть **Y** или **N**);

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс **AcquiredProducts**

Предназначен для хранения/извлечения списков продуктов, которые были куплены зарегистрированными посетителями (для «истории» приобретений покупателя).

### **AcquiredProducts::Add**

Назначение:

- Добавление указанного продукта в список купленных продуктов указанного зарегистрированного покупателя (в таблицу [proshop\\_products\\_acquired](#)).

Объявление:

- function Add(\$prod\_id, \$options, \$user\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **array \$options**. Комбинация вариантов опций продукта **\$prod\_id**, являющаяся массивом, состоящим из идентификаторов вариантов опций (из таблицы [proshop\\_products\\_options\\_vars](#)).
- **int \$user\_id**. Идентификатор покупателя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **AcquiredProducts::Delete**

Назначение:

- Удаление указанного продукта из таблицы [proshop\\_products\\_acquired](#) для указанного покупателя.

Объявление:

- function Delete(\$user\_id, \$prod\_id)

Параметры:

- **int \$user\_id**. Идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#).
- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **AcquiredProducts::DeleteByUser**

Назначение:

- Удаление всех продуктов из таблицы [proshop\\_products\\_acquired](#) для указанного зарегистрированного покупателя.

Объявление:

- function DeleteByUser(\$user\_id)

Параметры:

- **int \$user\_id**. Идентификатор покупателя из таблицы [proshop\\_users](#), список купленных продуктов которого необходимо удалить.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **AcquiredProducts::GetAll**

Назначение:

- Извлечение списка купленных продуктов указанным зарегистрированным покупателем из таблицы [proshop\\_products\\_acquired](#).

Объявление:

- function GetAll(\$user\_id)

Параметры:

- **int \$user\_id**. Идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - **user\_id** – идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#);
  - **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#);
  - **options** – массив идентификаторов вариантов опций продукта **prod\_id** (варианты опций продуктов хранятся в таблице [proshop\\_products\\_options\\_vars](#)).

## **Класс Articles**

Предназначен для управления статьями и информационными страницами магазина. Данный класс используется модулем **articles**.

### **Articles::Add**

Назначение:

- Добавление HTML-страницы в БД.

Объявление:

- `function Add($art_info)`

Параметры:

- **array \$art\_info**. Массив информации о странице. Должен содержать следующие ключи:
  - **cat\_id** – идентификатор категории (рубрики) статей (из таблицы [proshop\\_articles\\_categories](#)), в которую необходимо поместить данную статью;
  - **url\_name** – имя в URL страницы (для modRewrite). Макс. длина – 254 символа;
  - **show\_in\_list** – флаг принадлежности к статьям (показывать ли страницу в списке статей на витрине) – принимает значение **Y** или **N**;
  - числовой ключ, обозначающий идентификатор языка, для которого создаются название и содержимое страницы (см. пример ниже).

Возвращаемое значение:

- Тип **int/bool**. Идентификатор статьи или `false` в случае ошибки.

Пример:

- ```
// ...
$info = array(
    'url_name' => 'myarticle',
    'show_in_list' => 'Y'
);
foreach ($LANGUAGES as $lang) {
    $info[$lang['id']]['caption'] = $caption;
    $info[$lang['id']]['content'] = $content;
}
Articles::Add($info);
// ...
```

## **Articles::CategoryAdd**

Назначение:

- Добавление базовой информации о категории статей в БД.

Объявление:

- `function CategoryAdd($cat_id, $url_name)`

Параметры:

- **int \$cat\_id**. Идентификатор родительской категории из таблицы [proshop\\_articles\\_categories](#), в которую необходимо поместить создаваемую категорию.
- **string \$url\_name**. Служебное имя категории (для modRewrite). Макс. 254 символа.

Возвращаемое значение:

- Тип **int/bool**. Идентификатор созданной категории статей (из таблицы [proshop\\_articles\\_categories](#)) или `false` в случае ошибки.

## **Articles::CategoryDelete**

Назначение:

- Удаление указанной категории статей.

Объявление:

- `function CategoryDelete($id)`

Параметры:

- **int \$id**. Идентификатор категории статей из таблицы [proshop\\_articles\\_categories](#).

Возвращаемое значение:

- Тип **bool**. Всегда **true**.

## **Articles::CategoryGetAll**

Назначение:

- Извлечение списка категорий статей из БД в порядке, заданным полем **pos** таблицы [proshop\\_articles\\_categories](#).

Объявление:

- `function CategoryGetAll($cat_id, $lang_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории (из таблицы [proshop\\_articles\\_categories](#)), список подкатегорий которой необходимо извлечь. Если данный параметр равен 0, извлекается корневой список категорий.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия категорий. Данный параметр может быть равен 0 (в этом случае названия не извлекаются).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - все поля таблицы [proshop\\_articles\\_categories](#);
  - поле **caption** таблицы [proshop\\_articles\\_categories\\_texts](#) (если параметр **\$lang\_id** не равен 0).

## **Articles::CategoryGetAllFullest**

Назначение:

- Извлечение всех категорий статей из БД с их полными путями в дереве категории статей на указанном языке. Полученный список отсортирован по возрастанию путей.

Объявление:

- `function CategoryGetAllFullest($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь пути к извлекаемым категориям. Если данный параметр равен 0, пути не извлекаются.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_articles\\_categories](#);
  - **caption** - путь к категории на языке **\$lang\_id** (если он задан).

## **Articles::CategoryGetIdByName**

Назначение:

- Определение идентификатора категории статей по ее имени в URL (по ее служебному имени).

Объявление:

- `function CategoryGetIdByName($url_name)`

Параметры:

- **string \$url\_name**. Имя в URL категории (для modRewrite).

Возвращаемое значение:

- Тип **int**. Идентификатор категории статей из таблицы [proshop\\_articles\\_categories](#). Если категория не найдена, возвращается 0.

## **Articles::CategoryGetOne**

Назначение:

- Извлечение информации о категории (рубрике) статей из БД.

Объявление:

- `function CategoryGetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор категории статей из таблицы [proshop\\_articles\\_categories](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#), для которого необходимо извлечь название категории. Если данный параметр равен 0, название категории не извлекается.

Возвращаемое значение:

- Тип **array**. Массив, ключами которого являются:
  - все поля таблицы [proshop\\_articles\\_categories](#);
  - поле **caption** таблицы [proshop\\_articles\\_categories\\_texts](#).

## **Articles::CategoryGetPath**

Назначение:

- Извлечение пути к указанной категории статьи (путь категории – совокупность предшествующих ей родительских категорий, включая саму категорию).

Объявление:

- `function CategoryGetPath($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор категории из таблицы [proshop\\_articles\\_categories](#).
- **int \$lang\_id**. Идентификатор языка, для которого будут извлечены названия родительских категорий в пути. Если значение данного параметра равно 0, в возвращаемом массиве отсутствует ключ **caption**.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе - массив, каждый элемент которого является массивом со следующими ключами:
  - **id** – идентификатор категории;
  - **caption** – название категории (если параметр **\$lang\_id** не равен 0);
  - **url\_name** – служебное имя категории.

Пример:

- ```
$path = CategoryGetPath($cat_id);
foreach ($path as $item) {
    // $item['id'] – идентификатор одной из родительских категорий
    // $item['url_name'] – служебное имя одной из родительских категорий
}
```

## **Articles::CategoryIsNameExists**

Назначение:

- Определение, существует ли (занято ли) уже служебное имя категории статей (имя в URL для modRewrite) в БД.

Объявление:

- `function CategoryIsNameExists($url_name, $id = 0)`

Параметры:

- **string \$url\_name**. Имя, которое необходимо проверить.
- **int \$id**. Идентификатор категории (из таблицы [proshop\\_articles\\_categories](#)), которая не будет учтена при поиске (это нужно, когда модуль сохраняет то же самое служебное имя категории, добавленной ранее). Если равен 0, будут учитываться все категории.

Возвращаемое значение:

- Тип **bool**. `True`, если имя категории существует (уже занято). Иначе – `false`.

## **Articles::CategoryPathToCaptions**

Назначение:

- Конвертирование значения поля **path** таблицы [proshop\\_articles\\_categories](#) в путь к категории статей (поле **path** которой передано в качестве параметра) на заданном языке.

Объявление:

- `function CategoryPathToCaptions($path, $lang_id)`

Параметры:

- **string \$path**. Значение поля **path** таблицы [proshop\\_articles\\_categories](#) для какой-либо категории статей.
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#), на котором необходимо сформировать путь к категории.

Возвращаемое значение:

- Тип **string**. Путь к категории статей на заданном языке.

## **Articles::CategorySetCaption**

Назначение:

- Установка названия заданной категории статей на заданном языке.

Объявление:

- `function CategorySetCaption($cat_id, $caption, $lang_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории статей (из таблицы [proshop\\_articles\\_categories](#)), для которой необходимо установить название.
- **string \$caption**. Название категории статей. Макс. 254 символа.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого задается название **\$caption**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Articles::CategorySetPos**

Назначение:

- Установка позиции указанной категории статей по отношению к другим статьям того же уровня вложенности.

Объявление:

- `function CategorySetPos($id, $pos)`

Параметры:

- **int \$id**. Идентификатор категории статей из таблицы [proshop\\_articles\\_categories](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Articles::CategoryUpdate

Назначение:

- Обновление информации об указанной категории статей в БД.

Объявление:

- `function CategoryUpdate($id, $cat_id, $url_name)`

Параметры:

- **int \$id**. Идентификатор категории статей из таблицы [proshop\\_articles\\_categories](#).
- **int \$cat\_id**. Идентификатор родительской категории статей (из таблицы [proshop\\_articles\\_categories](#)) для данной категории **\$id**.
- **string \$url\_name**. Служебное имя категории статей (для modRewrite). Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Articles::Delete

Назначение:

- Удаление указанной страницы из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор удаляемой страницы.

Возвращаемое значение:

- Тип **bool**. Всегда true.

## Articles::GetAll

Назначение:

- Извлечение всех страниц из БД.

Объявление:

- `function GetAll($cat_id, $lang_id, $show_hidden, $only_ids = false, $complete_sql = "")`

Параметры:

- **int \$cat\_id**. Идентификатор категории (рубрики) статей (из таблицы [proshop\\_articles\\_categories](#)), для которой необходимо извлечь список статей. Если данный параметр равен 0, извлекаются статьи, не принадлежащие ни одной из категорий (рубрик).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого будут извлечены языкозависимые текстовые составляющие страниц (название, описание). Если значение данного параметра равно 0, текстовые составляющие не извлекаются. Значение данного параметра игнорируется, если **\$only\_ids** равен **true**.
- **bool \$show\_hidden**. Извлекать ли страницы, флаг **show\_in\_list** которых имеет значение **N**.
- **bool \$only\_ids**. Возвращать ли только идентификаторы страниц. Если данный параметр равен **true**, параметр **\$lang\_id** игнорируется.



- **string \$complete\_sql.** Часть SQL-запроса, присоединяемая в конец основного SQL-запроса, формируемого функцией (обычно нужна для механизма пейджинга – см. класс [Pager](#)).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого содержит следующие ключи:
  - все поля таблицы [proshop\\_articles](#) (или только поле **id**, если параметр **\$only\_ids** принимает значение **true**);
  - поля **caption** и **content** таблицы [proshop\\_articles\\_texts](#), если **\$lang\_id** не равен 0 и **\$only\_ids** равен **false**;

Пример:

- ```
$rows = Articles::GetAll($SHOP_LANGUAGE['id'], false);
foreach ($rows as $row) {
    // $row['id'] – идентификатор страницы
    // $row['caption'] – название страницы
    // ...
}
```

## **Articles::GetAllByProduct**

Назначение:

- Извлечение списка статей, связанных с заданным продуктом.

Объявление:

- `function GetAllByProduct($prod_id, $lang_id)`

Параметры:

- **int \$prod\_id.** Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия статей.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - **id** – идентификатор статьи из таблицы [proshop\\_articles](#);
  - **url\_name** – имя в URL (для modRewrite) статьи;
  - **caption** – название статьи на языке **\$lang\_id**.

## **Articles::GetOne**

Назначение:

- Извлечение указанной страницы из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id.** Идентификатор страницы, которую нужно извлечь.

- **int \$lang\_id.** Идентификатор языка, для которого нужно извлечь текстовые составляющие страницы (название и содержимое). Если данный параметр равен 0, текстовые составляющие не извлекаются.

Возвращаемое значение:

- Тип **array**. Массив, ключами которого являются:
  - все поля таблицы [proshop\\_articles](#);
  - поля **caption** и **content** таблицы [proshop\\_articles\\_texts](#);

Пример:

- ```
// ...
$row = Articles::GetOne($id, $SHOP_LANGUAGE['id']);
// $row['url_name'] – имя в URL (для modRewrite) страницы
// $row['caption'] – название страницы
// ...
```

### **Articles::GetOneByName**

Назначение:

- Извлечение страницы из БД по ее имени в URL (по ее служебному имени).

Объявление:

- `function GetOneByName($url_name, $lang_id)`

Параметры:

- **string \$url\_name.** Имя в URL страницы (для modRewrite).
- **int \$lang\_id.** Идентификатор языка, для которого необходимо извлечь языкозависимые текстовые составляющие страницы (название и содержимое).

Возвращаемое значение:

- Тип **array/bool**. False в случае ошибки. Иначе - массив, ключами которого являются:
  - все поля таблицы [proshop\\_articles](#);
  - поля **caption** и **content** таблицы [proshop\\_articles\\_texts](#);

Пример:

- ```
// ...
$row = Articles::GetOneByName('myarticle', $SHOP_LANGUAGE['id']);
// $row['id'] – идентификатор страницы
// $row['content'] – содержимое страницы
// ...
```

### **Articles::IsNameExists**

Назначение:

- Определение, существует ли в БД страница с указанным служебным именем (именем в URL – для modRewrite)

Объявление:

- `function IsNameExists($url_name, $id = 0)`

Параметры:

- **string \$url\_name**. Имя в URL страницы (для `modRewrite`).
- **int \$id**. Идентификатор страницы, которую не нужно учитывать при поиске в БД (это нужно, когда модуль сохраняет то же самое имя в URL страницы, добавленной ранее). Может быть 0 – в этом случае при поиске будут учитываться все страницы.

Возвращаемое значение:

- Тип **bool**. True/false в зависимости от успешности операции.

## **Articles::RecProdAdd**

Назначение:

- Привязка указанного продукта к указанной статье.

Объявление:

- `function RecProdAdd($article_id, $rprod_id)`

Параметры:

- **int \$article\_id**. Идентификатор статьи из таблицы [proshop\\_articles](#).
- **int \$rprod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Articles::RecProdDelete**

Назначение:

- Удаление указанной связи продукта и статьи из таблицы [proshop\\_articles\\_products](#).

Объявление:

- `function RecProdDelete($id)`

Параметры:

- **int \$id**. Идентификатор записи в таблице [proshop\\_articles\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Articles::RecProdGetAll**

Назначение:

- Извлечение всех записей из таблицы [proshop\\_articles\\_products](#) для указанной статьи в порядке, определенным полем **pos** этой таблицы.

Объявление:

- `function RecProdGetAll($article_id)`

Параметры:

- **int \$article\_id**. Идентификатор статьи из таблицы [proshop\\_articles](#).

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_articles\\_products](#).

## **Articles::RecProdSetPos**

Назначение:

- Установка позиции заданного рекомендуемого продукта к определенной статье по отношению к другим продуктам, рекомендуемым к этой же статье. Другими словами, установка значения поля **pos** таблицы [proshop\\_articles\\_products](#) для заданной записи этой таблицы.

Объявление:

- `function RecProdSetPos($rprod_id, $pos)`

Параметры:

- **int \$rprod\_id**. Идентификатор записи из таблицы [proshop\\_articles\\_products](#).
- **int \$pos**. Значение поля **pos**.

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

## **Articles::SetCaption**

Назначение:

- Сохранение названия страницы.

Объявление:

- `function SetCaption($article_id, $caption, $lang_id)`

Параметры:

- **int \$article\_id**. Идентификатор страницы.
- **string \$caption**. Название страницы на указанном языке. Макс. длина – 65533 символа.
- **int \$lang\_id**. Идентификатор языка, для которого сохраняется название.

Возвращаемое значение:

- Тип **bool**. `True/false` в зависимости от успешности операции.

## **Articles::SetPos**

Назначение:

- Сохранение позиции статьи относительно других статей в отображаемом списке статей на витрине.

Объявление:

- `function SetPos($id, $pos)`

Параметры:

- **int \$id**. Идентификатор статьи из таблицы [proshop\\_articles](#).
- **int \$pos**. Значение позиции статьи.

Возвращаемое значение:

- Тип **bool**. `True/false` в зависимости от успешности операции.

## **Articles::SetShowInList**

Назначение:

- Установка флага **show\_in\_list** (принадлежности к статьям) для указанной страницы.

Объявление:

- `function SetShowInList($id, $flag)`

Параметры:

- **int \$id**. Идентификатор страницы.
- **char \$flag**. Значение флага (**Y** – страница принадлежит статьям, **N** – не принадлежит).

Возвращаемое значение:

- Тип **bool**. True/false в зависимости от успешности операции.

## **Articles::Update**

Назначение:

- Обновление информации, связанной с указанной страницей, в БД.

Объявление:

- `function Update($id, $article_info, $lang_id)`

Параметры:

- **int \$id**. Идентификатор страницы.
- **array \$article\_info**. Массив информации о странице. Должен содержать следующие ключи:
  - **cat\_id** – идентификатор категории (рубрики) статей (из таблицы [proshop\\_articles\\_categories](#)), в которую необходимо поместить данную статью;
  - **url\_name** – имя в URL страницы (для modRewrite). Макс. длина – 254 символа;
  - **show\_in\_list** – флаг принадлежности к статьям (показывать ли страницу в списке статей на витрине) – принимает значение **Y** или **N**;
  - **caption** – название страницы для указанного языка. Макс. длина – 65533 символа;
  - **content** – содержимое страницы для указанного языка. Макс. длина – 65533 символа.
- **int \$lang\_id**. Идентификатор языка, для которого указываются текстовые составляющие страницы (название и содержимое).

Возвращаемое значение:

- Тип **bool**. True/false в зависимости от успешности операции.

## **Класс Banners**

Предназначен для управления баннерами. Используется баннерной системой.

### **Banners::Add**

Назначение:

- Добавление баннера.

Объявление:

- `function Add($info)`

Параметры:

- **array \$info**. Массив с информацией о баннере. Должен содержать следующие ключи:
  - **name** – имя баннера. Макс. 254 символа;
  - **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), для которого добавляется баннер;
  - **type** – тип добавляемого баннера (**html**, **flash** или **img**);
  - **file** – описание загруженного файла баннера (из **\$\_FILES**). См. пример;
  - **href** – ссылка баннера (страница, на которую он ссылается). Макс. 65533 символа;
  - **width** – ширина баннера в пикс.;
  - **height** – высота баннера в пикс.;
  - **place\_id** – идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#);
  - **begin\_timestamp** – метка времени Unix, с которой должны начаться показы баннера;
  - **end\_timestamp** – метка времени Unix, на которой должны прекратиться показы баннера.

Возвращаемое значение:

- Тип **int/bool**. `False` – в случае ошибки. Иначе - идентификатор созданного баннера (из таблицы [proshop\\_banners](#)).

Пример:

- ```
$info = array(
    'name' => $name,
    // ...
    'file' => $_FILES['file'],
    // ...
);
Banners::Add($info);
```

## **Banners::AddPlace**

Назначение:

- Добавление информации о новом баннероместе в БД.

Объявление:

- `function AddPlace($info)`

Параметры:

- **array \$info**. Массив с информацией о баннероместе. Должен содержать следующие ключи:
  - **name** – имя баннероместа. Макс. 254 символа;
  - **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), для которого добавляется баннероместо;

- **location** – расположение баннероместа (**L** – левая колонка, **R** – правая колонка, **CT** – над центром страницы, **CB** – под центром страницы);
- **pages** – массив адресов страниц магазина, на которых должно располагаться баннероместо;
- **weight** – вес блока баннероместа, если оно располагается в левой или правой колонке.
- **only\_homepage** – должно ли баннероместо отображаться только на главной странице витрины: **Y** – должно отображаться только на главной странице, **N** – может отображаться не только на главной странице.
- **not\_homepage** – должно ли баннероместо отображаться на всех страницах витрины, кроме главной: **Y** – не должно отображаться на главной странице, **N** – может отображаться на главной странице.

Возвращаемое значение:

- Тип **int/bool**. **False** – в случае ошибки. Иначе – идентификатор созданного баннероместа в таблице [proshop\\_banners\\_places](#).

### **Banners::AllPagesPlaces**

Назначение:

- Извлечение идентификаторов баннеромест, предназначенных для показа на всех страницах магазина.

Объявление:

- `function AllPagesPlaces($lang_id, $location = "", $id = 0, $homepage_matter = "")`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого произвести поиск баннеромест, показываемых на всех страницах магазина.
- **string \$location**. Фильтр по расположению баннероместа (**L** – левая колонка, **R** – правая колонка, **CT** – над центром страницы, **CB** – под центром страницы).
- **int \$id**. Идентификатор баннероместа (из таблицы [proshop\\_banners\\_places](#)), которое не нужно учитывать при поиске. Если данный параметр равен 0, учитываются все баннероместа.
- **string \$homepage\_matter**. Если данный параметр равен 'homepage', будут учтаны баннероместа только для главной страницы. Если параметр равен 'not\_homepage', будут учтаны только баннероместа не для главной страницы. Если равен пустой строке, будут учтаны все баннероместа.

Возвращаемое значение:

- Тип **array**. Массив идентификаторов баннеромест (из таблицы [proshop\\_banners\\_places](#)).

### **Banners::Click**

Назначение:

- Фиксирование факта нажатия на баннер в БД (в таблице [proshop\\_banners\\_clicks](#)).

Объявление:

- function Click(\$id, \$visitor\_id, \$page)

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).
- **int \$visitor\_id**. Идентификатор посетителя магазина (`$_SESSION['visitor_id']`).
- **string \$page**. Адрес страницы магазина, на которой было произведено нажатие на баннер. Макс. 2000 символов.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Banners::Delete**

Назначение:

- Удаление указанного баннера из БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Banners::DeletePlace**

Назначение:

- Удаление указанного баннероместа из БД.

Объявление:

- function DeletePlace(\$id)

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Banners::DelPlaceBanners**

Назначение:

- Операция, обратная привязке баннеров к баннероместу (все баннеры «отвязываются» от указанного баннероместа).

Объявление:

- function DelPlaceBanners(\$id)

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:



- Тип **bool**. True/false – в зависимости от успешности операции.

## **Banners::FindIntersepts**

Назначение:

- Поиск страниц магазина, на которых на одно место претендует больше одного баннероместа. Применяется при добавлении баннера, расположение которого - над центром или под центром, чтобы предотвратить размещение на одном и том же месте и на одной и той же странице более, чем одного баннероместа.

Объявление:

- `function FindIntersepts($pages, $location, $lang_id, $id = 0)`

Параметры:

- **array \$pages**. Массив адресов страниц, которые необходимо проверить.
- **string \$location**. Расположение баннероместа (**CT** – над центром, **CB** – под центром).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо произвести поиск пересекающихся страниц.
- **int \$id**. Идентификатор баннероместа (из таблицы [proshop\\_banners\\_places](#)), который не будет учитываться при поиске. Если значение данного параметра равно 0, будут учитываться все баннероместа.

Возвращаемое значение:

- Тип **array**. Массив пересекающихся страниц (страниц магазина, на которых на одно место претендует больше одного баннероместа для указанного языка и указанного расположения баннероместа).

## **Banners::GetAll**

Назначение:

- Извлечение из БД всех баннеров для указанного языка.

Объявление:

- `function GetAll($lang_id, $only_visible = false, $type = "", $place_id = 0)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)).
- **bool \$only\_visible**. Если значение данного параметра равно true, из базы извлекаются только видимые на данный момент баннеры (время показа которых не истекло). Иначе - извлекаются все баннеры не зависимо от текущего времени.
- **string \$type**. Фильтр по типу баннеров (**html**, **flash**, **img**). Если данный параметр пуст, извлекаются баннеры всех типов.
- **int \$place\_id**. Фильтр по идентификатору баннероместа, которому должен принадлежать баннер. Если данный параметр равен 0, извлекаются все баннеры вне зависимости от принадлежности тому или иному баннероместу.

Возвращаемое значение:

- Тип **array**. Массив, содержащий ключи, совпадающие с идентификаторами найденных баннеромест (из таблицы [proshop\\_banners\\_places](#)). При этом значениями данных ключей являются массивы, ключи которых совпадают с полями таблицы [proshop\\_banners](#). См. пример.

Пример:

- ```
$banners = Banners::GetAll($SHOP_LANGUAGE['id'], true);
foreach ($banners[$place_id] as $banner_id => $banner) {
    // $banner_id – идентификатор баннера, расположенного в
    // баннероместе с идентификатором $place_id
    // $banner['name'] – имя баннера
    // ...
}
```

## **Banners::GetAllPlaces**

Назначение:

- Извлечение баннеромест из таблицы [proshop\\_banners\\_places](#).

Объявление:

- `function GetAllPlaces($lang_id, $location = "")`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь баннероместа.
- **string \$location**. Фильтр по расположению баннеромест. Может принимать значения:
  - **CT** – возвращаются только баннероместа, расположенные над центральной частью;
  - **CB** – возвращаются только баннероместа, расположенные под центральной частью;
  - **L** – возвращаются только баннероместа, расположенные в левой колонке;
  - **R** – возвращаются только баннероместа, расположенные в правой колонке.

Возвращаемое значение:

- Тип **array**. Массив, содержащий следующие ключи:
  - **CT** – массив баннеромест, расположенных над центральной частью;
  - **CB** – массив баннеромест, расположенных под центральной частью;
  - **L** – массив баннеромест, расположенных в левой колонке;
  - **R** – массив баннеромест, расположенных в правой колонке.

Пример:

- ```
$places = Banners::GetAllPlaces($SHOP_LANGUAGE['id']);
foreach ($places['CT'] as $place_id => $place) {
    // $place['name'] – название баннероместа, расположенного над центральной частью
    // ...
}
```

## **Banners::GetClicksStat**

Назначение:

- Извлечение информации о нажатиях на указанный баннер из БД (журнал переходов).

Объявление:

- `function GetClicksStat($id, $complete_sql = "")`

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).
- **string \$complete\_sql**. Часть SQL-запроса, которую можно добавить в конец SQL-запроса, формируемого функцией. Используется пейджингом (см. класс [Pager](#)).

Возвращаемое значение:

- Тип **array**. Массив журнала переходов, содержащий следующие ключи:
  - все поля таблицы [proshop\\_banners\\_clicks](#);
  - **ip** – IP-адрес нажавшего на баннер пользователя;
  - **user\_id** – идентификатор пользователя из таблицы [proshop\\_users](#), или 0, если пользователь не зарегистрирован в магазине;
  - **uagent** – строка **User Agent** браузера пользователя;
  - **email** – E-Mail адрес пользователя из таблицы [proshop\\_users](#), или пустая строка, если пользователь не зарегистрирован в магазине.

## **Banners::GetFilename**

Назначение:

- Извлечение имени файла указанного баннера из таблицы [proshop\\_banners](#). Файлы баннеров располагаются в директории, путь к которой прописан в константе [BANNERS\\_PATH](#).

Объявление:

- `function GetFilename($id)`

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).

Возвращаемое значение:

- Тип **string/bool**. `False` – в случае ошибки. Иначе – значение поля **filename** таблицы [proshop\\_banners](#) для указанного баннера.

## **Banners::GetLocation**

Назначение:

- Определение места расположения указанного баннероместа.

Объявление:

- `function GetLocation($id)`

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – значение поля **location** таблицы [proshop\\_banners\\_places](#) для указанного баннероместа.

## **Banners::GetName**

Назначение:

- Извлечение названия указанного баннера из таблицы [proshop\\_banners](#).

Объявление:

- function GetName(\$id)

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – значение поля **name** таблицы [proshop\\_banners](#) для указанного баннера.

## **Banners::GetOne**

Назначение:

- Извлечение информации об указанном баннере из БД.

Объявление:

- function GetOne(\$id)

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив с ключами, совпадающими с полями таблицы [proshop\\_banners](#).

## **Banners::GetOnePlace**

Назначение:

- Извлечение информации об указанном баннероместе из БД.

Объявление:

- function GetOnePlace(\$id)

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля из таблицы [proshop\\_banners\\_places](#);
  - **pages** – массив страниц магазина, на которых располагается баннероместо.

## **Banners::GetPages**

Назначение:

- Извлечение списка адресов страниц (из таблицы [proshop\\_banners\\_pages](#)), на которых могут быть размещены баннеры.

Объявление:

- `function GetPages($is_homepage)`

Параметры:

- **bool \$is\_homepage**. Если значение данного параметра равно **true**, то предполагается, что функция вызвана с главной страницы витрины. В этом случае должны быть извлечены адреса страниц, которые могут быть отображены на главной странице витрины.

Возвращаемое значение:

- Тип **array**. Массив с информацией о страницах. См. пример.

Пример:

- ```
$pages = Banners::GetPages();
foreach ($pages as $addr => $page) {
    foreach ($page as $place_id) {
        // $place_id – идентификатор баннероместа, размещенного на странице с адресом $addr
    }
}
```

## **Banners::GetPlaceName**

Назначение:

- Извлечение названия указанного баннероместа из таблицы [proshop\\_banners\\_places](#).

Объявление:

- `function GetPlaceName($id)`

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:

- Тип **string/bool**. **False** – в случае ошибки. Иначе – значение поля **name** таблицы [proshop\\_banners\\_places](#) для указанного баннероместа.

## **Banners::GetStat**

Назначение:

- Извлечение кол-ва показов указанного баннера (уникальных и всего) и кол-ва нажатий на него.

Объявление:

- `function GetStat($id)`

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).

Возвращаемое значение:

- Тип **array**. Массив, содержащий следующие ключи:

- **unique** – кол-во уникальных показов баннера;
- **total** – общее кол-во показов баннера;
- **clicks** – кол-во переходов по баннеру (нажатий).

## **Banners::GetWeight**

Назначение:

- Извлечение веса бокового блока (из таблицы [proshop\\_cblocks](#)) указанного баннероместа. Применяется только для тех баннеромест, которые расположены в боковой колонке.

Объявление:

- function GetWeight(\$id)

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки (например, если указанное баннероместо не расположено в боковой колонке). Иначе – целочисленное значение веса блока (из таблицы [proshop\\_cblocks](#)).

## **Banners::SetPlace**

Назначение:

- Привязка указанного баннера к указанному баннероместу.

Объявление:

- function SetPlace(\$banner\_id, \$place\_id)

Параметры:

- **int \$banner\_id**. Идентификатор баннера из таблицы [proshop\\_banners](#).
- **int \$place\_id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Banners::Showed**

Назначение:

- Фиксирование факта просмотров указанных баннеров.

Объявление:

- function Showed(\$ids, \$visitor\_id)

Параметры:

- **array \$ids**. Идентификаторы баннеров из таблицы [proshop\\_banners](#).
- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#). При работе магазина данный идентификатор необходимо брать из [\\$\\_SESSION\['visitor\\_id'\]](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Banners::Update**

Назначение:

- Обновление информации об указанном баннере.

Объявление:

- `function Update($id, $info)`

Параметры:

- **int \$id**. Идентификатор баннера из таблицы [proshop\\_banners](#).
- **array \$info**. См. описание соответствующего параметра функции [Banners::Add](#) (за исключением ключей **lang\_id** и **type**).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Banners::UpdatePlace**

Назначение:

- Обновление информации об указанном баннероместе.

Объявление:

- `function UpdatePlace($id, $info)`

Параметры:

- **int \$id**. Идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#).
- **array \$info**. См. описание соответствующего параметра функции [Banners::AddPlace](#) (за исключением ключа **lang\_id**).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Bestsellers**

Предназначен для управления продуктами-бестселлерами (лидерами продаж). Используется модулем **bestsellers**. В число бестселлеров входят все продукты, проданные хотя бы один раз.

### **Bestsellers::GetAll**

Назначение:

- Извлечение бестселлеров из БД.

Объявление:

- `function GetAll($cat_id, $lang_id, $count)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, в которой производится поиск лидеров продаж (включая все ее подкатегории). Если равен 0, поиск производится во всех категориях.

- **int \$lang\_id.** Идентификатор языка, для которого будут извлечены текстовые составляющие продуктов (название, краткое описание и т.д.). Если равен 0, текстовые составляющие не извлекаются.
- **int \$count.** Максимальное кол-во извлекаемых лидеров продаж.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого содержит следующие ключи:
  - все поля таблицы [proshop\\_products](#);
  - поля **alt**, **caption** и **short\_descr** таблицы [proshop\\_products\\_texts](#).

Пример:

- ```
$rows = Bestsellers::GetAll(0, $SHOP_LANGUAGE['id'], 10);
foreach ($rows as $row) {
    // $row['id'] – идентификатор продукта
    // $row['sold_qty'] – кол-во продаж продукта
    // $row['caption'] – название продукта
    // ...
}
```

## **Bestsellers::GetCount**

Назначение:

- Определение общего кол-ва бестселлеров.

Объявление:

- `function GetCount($cat_id)`

Параметры:

- **int \$cat\_id.** Идентификатор категории, для которой производится определение кол-ва лидеров продаж (включая все ее подкатегории). Если равен 0, определение производится для всех категорий.

Возвращаемое значение:

- Тип **int**. Кол-во бестселлеров.

## **Bestsellers::GetIds**

Назначение:

- Извлечение идентификаторов продуктов-бестселлеров (лидеров продаж).

Объявление:

- `function GetIds($cat_id, $count)`

Параметры:

- **int \$cat\_id.** Идентификатор категории, продукты-бестселлеры которой (включая все ее подкатегории) будут найдены в результате работы функции. Если категория не задана (параметр равен 0), будут извлечены продукты-бестселлеры для всех категорий магазина.



- **int \$count**. Максимальное кол-во продуктов-бестселлеров, идентификаторы которых нужно извлечь.

Возвращаемое значение:

- Тип **array**. Массив идентификаторов продуктов (идентификаторы из таблицы [proshop\\_products](#)).

## **Bestsellers::SetSoldQty**

Назначение:

- Установка кол-ва продаж для указанного продукта.

Объявление:

- `function SetSoldQty($prod_id, $sold_qty)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта.
- **int \$sold\_qty**. Кол-во проданных экземпляров продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Bill**

Предназначен для генерирования счета на оплату заказа.

### **\$bill->Show**

Назначение:

- Генерирование и вывод счета в окно браузера.

Объявление:

- `public function Show($info)`

Параметры:

- **array \$info**. Массив с информацией, необходимой для генерирования счета. Должен содержать следующие ключи:
  - **date** – дата счета (в формате ГГГГ-ММ-ДД);
  - **num** – номер счета (в любом формате);
  - **no\_vat** – флаг того, что НДС не предусмотрен (принимает значения true/false);
  - **recipient** – массив с информацией об организации, выставившей счет (грузоотправителе). Может содержать следующие ключи:
    - **director** – ФИО директора (полностью, либо в формате Фамилия И.О.);
    - **accountant** - ФИО главного бухгалтера (полностью, либо в формате Фамилия И.О.);
    - **company** – название организации;
    - **zipcode** – почтовый индекс юридического адреса организации;
    - **city** – город юридического адреса организации;

- **address** – улица, номер дома, номер квартиры/офиса организации;
- **inn** – ИНН организации;
- **kpp** – КПП организации;
- **bank\_name** – наименование банка организации;
- **bank\_bik** – БИК банка организации;
- **bank\_rs** – р/с организации;
- **bank\_ks** – к/с организации;
- **customer** – массив с информацией об организации, которой выставляется счет (грузополучателе). Может содержать следующие ключи:
  - **company** – название организации;
  - **inn** – ИНН организации;
  - **kpp** – КПП организации;
- **products** – массив с информацией о заказанных продуктах/услугах. Каждый элемент этого массива может содержать следующие ключи:
  - **caption** – наименование продукта/услуги;
  - **qty** – заказанное кол-во;
  - **price** – стоимость одного экземпляра продукта/услуги;
  - **total** – сумма;
  - **tax\_percent** – величина (ставка) НДС в процентах. Данный параметр имеет значение только при `$info['no_vat']` равным **false**;
  - **tax\_value** – сумма НДС (в рублях или валюте). Данный параметр имеет значение только при `$info['no_vat']` равным **false**;
  - **sum** – итоговая сумма с НДС. Данный параметр имеет значение только при `$info['no_vat']` равным **false**;
- **total** – сумма счета без учета НДС;
- **tax** – величина НДС счета (в рублях или валюте);
- **discount** – величина скидки в процентах;
- **shipping** – стоимость доставки (в рублях или валюте);
- **sum** – итоговая сумма счета с учетом НДС;
- **curr\_code** – международное трехбуквенное обозначение валюты, в которой выставляется счет (RUB или RUR - рубль). Регистр имеет значение. Если счет выставляется не в рублях, то сумма прописью не отображается в счете.

Возвращаемое значение:

- Функция ничего не возвращает.

## Класс BillOfParcels

Предназначен для генерирования товарной накладной в формате PDF. Применяется преимущественно для русских интернет-магазинов.

## **\$bop->Generate**

Назначение:

- Генерирование накладной.

Объявление:

- `public function Generate($info, $name = "")`

Параметры:

- **array \$info**. Массив с информацией, необходимой для генерирования накладной. Должен содержать следующие ключи:
  - **num** – номер накладной;
  - **date** – дата накладной;
  - **supplier** – реквизиты поставщика;
  - **customer** – реквизиты покупателя;
  - **no\_vat** – учитываются ли налоги (**true** – налоги не учитываются, **false** - учитываются);
  - **products** – массив с информацией о заказанных продуктах. Должен содержать следующие ключи:
    - **caption** – наименование продукта;
    - **qty** – количество заказанных экземпляров продукта;
    - **price** – цена одного экземпляра продукта в рублях;
    - **total** – сумма в рублях для данного продукта;
    - **tax\_percent** – процентная ставка НДС для данного продукта (если **\$info['no\_vat']** имеет значение **true**, данный параметр игнорируется);
    - **tax\_value** – сумма НДС для данного продукта (если **\$info['no\_vat']** имеет значение **true**, данный параметр игнорируется);
    - **sum** – сумма в рублях с учетом НДС для данного продукта (если **\$info['no\_vat']** имеет значение **true**, данный параметр игнорируется);
  - **total** – итоговая сумма в рублях без учета НДС;
  - **tax** – сумма НДС в рублях (если **\$info['no\_vat']** имеет значение **true**, данный параметр игнорируется);
  - **sum** – итоговая сумма в рублях с учетом НДС;
  - **curr\_code** – международный трехбуквенный код валюты, в которой представлены значения.
- **string \$name**. Имя генерируемого PDF-файла. Если значение данного параметра пустое, функция не выбрасывает файл в браузер, а возвращает его содержимое.

Возвращаемое значение:

- Тип **string**. Если параметр **\$name** пустой, возвращается содержимое сгенерированного PDF-файла. Иначе – ничего не возвращается (файл выкидывается в браузер).

## Класс Cache

Предназначен для реализации и использования механизма кэширования данных. Если константа [MEMCACHE\\_ALLOWED](#) имеет значение **true**, то в качестве хранилища кэшированных данных используется Memcache. Иначе – используются база данных и сессии. Кэширование отключается путем изменения значения константы [CACHE\\_ENABLED](#) на **false**.

### **\$CACHE->Clean**

Назначение:

- Очистка кэша и откомпилированных Smarty-шаблонов дизайна, перекомпиляция шаблонов, принудительное обновление кэшированной информации.

Объявление:

- `function Clean($smarty_recompile = false)`

Параметры:

- **bool \$smarty\_recompile**. Флаг, определяющий, перекомпилировать ли шаблоны дизайна Smarty и освежить ли информацию в кэше.

Возвращаемое значение:

- Тип **bool**. True/false в зависимости от успешности операции.

### **\$CACHE->Get**

Назначение:

- Извлечение кэшированной информации из кэша.

Объявление:

- `function Get($key, $id = '0')`

Параметры:

- **string \$key**. Уникальный ключ кэшированной информации.
- **string \$id**. Дополнение к основному ключу \$key – у основного ключа может быть несколько подключей. Например – кэширование текстовой информации о лидерах продаж. Для примера, ключом может быть **c\_best sellers**, а дополнением к нему – идентификатор языка. Как правило, для каждой отдельной сущности ключ кэширования должен быть уникален по своему названию, а также иметь префикс **c\_** или **a\_**, обозначающий зону, в которой используется (витрина или админка).

Возвращаемое значение:

- Тип **любой**. Возвращаемое значение может быть любого типа – в зависимости от того, что было записано к кэш.

### **\$CACHE->IsActual**

Назначение:

- Определение, актуальна ли кэшированная информация, принадлежащая указанному ключу и подключу. Применяется перед тем, как вызвать `$CACHE->Get` или `$CACHE->Set`.

Определение:

- `function IsActual($key, $sid = '0', $ttl = -1)`

Параметры:

- **string \$key**. Уникальный ключ кэшированной информации.
- **string \$sid**. См. описание [\\$CACHE->Get](#).
- **int \$ttl**. Количество секунд, в течение которого кэшированная информация считается актуальной. Если значение данного параметра равно -1 (по умолчанию), актуальность информации определяется значением времени, сохраненным в базе данных, для данного ключа. В большинстве случаев параметр `ttl` не нужен.

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от актуальности информации.

Пример:

- ```
if ($CACHE->IsActual($key, $sid)) {  
    list($bestsellers, $prods_ids) = $CACHE->Get($key, $sid);  
} else {  
    // ...  
    $CACHE->Set($key, array($bestsellers, $prods_ids), $sid);  
}
```

## **`$CACHE->Set`**

Назначение:

- Сохранение информации в кэше.

Объявление:

- `function Set($key, $data, $sid = '0')`

Параметры:

- **string \$key**. Уникальный ключ кэшированной информации.
- **mixed \$data**. Сохраняемая информация (любой тип).
- **string \$sid**. См. описание [\\$CACHE->Get](#).

Возвращаемое значение:

- Функция ничего не возвращает.

Пример:

- См. пример к [\\$CACHE->IsActual](#).

## **`$CACHE->UpdateTimestamp`**

Назначение:

- Обновление меток времени в БД для указанных ключей с целью сделать некоторую информацию в кэше неактуальной.

Объявление:

- `function UpdateTimestamp($keys)`

Параметры:

- **string/array \$keys**. Может быть как строкой, так и массивом строк. Данный параметр содержит ключ (или массив ключей), метку времени которого требуется обновить в базе данных.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс Cart

Предназначен для обеспечения работы корзины на витрине. Используется модулем **cart**. Корзина представляет собой область памяти в сессии, в которой хранится информация о продуктах, заказываемых покупателем.

### **Cart::AddProduct**

Назначение:

- Добавление продукта в корзину покупателя.

Объявление:

- `function AddProduct($prod, $qty, $cat_name, $cat_id, $options, $pending, $pos)`

Параметры:

- **array \$prod**. Массив информации о добавляемом продукте, ключи которого совпадают с полями из таблицы [proshop\\_products](#).
- **int \$qty**. Кол-во экземпляров добавляемого продукта.
- **string \$cat\_name**. Название категории, из которой продукт добавляется в корзину, либо название категории самого продукта.
- **int \$cat\_id**. Идентификатор категории, из которой продукт добавляется в корзину, либо идентификатор категории самого продукта.
- **array \$options**. Массив идентификаторов вариантов опций продукта. Варианты опций продукта перечислены в таблице [proshop\\_products\\_options\\_vars](#). Опции продукта перечислены в таблице [proshop\\_products\\_options](#). Если продукт не имеет опций, данный массив должен быть пустым.
- **bool \$pending**. Если значение данного параметра равно **true**, то считается, что добавляемый продукт на момент добавления его в корзину отсутствовал на складе. Если значение равно **false**, предполагается, что продукт присутствовал на складе.
- **int \$pos**. Позиция добавляемого товара в корзине. Если равна 0, товар добавляется в конец списка продуктов корзины. Требуется для правильной сортировки товаров корзины при их обновлении/удалении.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Cart::CalcDiscount**

Назначение:

- Подсчет абсолютного и процентного значения скидки на корзину товаров покупателя для указанной суммы. Учитываются накопительные скидки для зарегистрированного покупателя.

Объявление:

- `function CalcDiscount($sum)`

Параметры:

- **float \$sum**. Значение суммы (в базовой валюте магазина), для которой необходимо определить величину скидки.

Возвращаемое значение:

- Тип **array**. Массив, первый элемент которого является абсолютным значением скидки в базовой валюте магазина, а второй элемент – значением скидки в процентах.

## **Cart::CalcToShow**

Назначение:

- Пересчет показателей (кол-во продуктов, сумма) в корзине. Данная функция также подготавливает SMARTY-переменную **cart\_qty** (склоненное существительное «товар») для ее отображения на витрине.

Объявление:

- `function CalcToShow($shop_currency, $forced = false)`

Параметры:

- **array \$shop\_currency**. Массив информации о текущей выбранной валюте магазина. Должен совпадать с глобальной переменной [\\$SHOP\\_CURRENCY](#).
- **bool \$forced**. Если данный параметр имеет значение `false`, функция пересчитает показатели только в том случае, если содержимое корзины поменялось. Если значение параметра равно `true`, пересчет будет произведен в любом случае.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::Clear**

Назначение:

- Очистка корзины покупателя.

Объявление:

- `function Clear()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::ClearRTShippingDates**

Назначение:

- Очистка кэшированных дат доставки, рассчитанных в реальном времени (например, сервисом FedEx), в корзине. Применяется, когда содержимое корзины изменилось и нужно заново определить диапазон дат доставки.

Объявление:

- `function ClearRTShippingDates()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::ClearRTShippingPrices**

Назначение:

- Очистка кэшированных стоимостей доставки, рассчитанных в реальном времени (например, сервисом FedEx), в корзине. Применяется, когда содержимое корзины изменилось и нужно заново определить стоимости доставок.

Объявление:

- `function ClearRTShippingPrices()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::DelAllProducts**

Назначение:

- Удаление всех продуктов из корзины.

Объявление:

- `function DelAllProducts($inc_qty = true)`

Параметры:

- **bool \$inc\_qty**. Если данный параметр равен true, при удалении продуктов из корзины производится возврат удаляемых товаров на склад. Если равен false – возврат не производится.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::DelProduct**

Назначение:

- Удаление продукта из корзины.

Объявление:



- `function DelProduct($cart_id, $inc_qty = true)`

Параметры:

- **int \$cart\_id**. Идентификатор элемента в корзине, который требуется удалить.
- **bool \$inc\_qty**. Если данный параметр равен `true`, при удалении продукта из корзины производится возврат удаляемого товара на склад. Если равен `false` – возврат не производится.

Возвращаемое значение:

- Тип **bool**. Всегда `true`.

## **Cart::GetAllowEgoodsShipping**

Назначение:

- Извлечение значения флага, который показывает, нужна ли покупателю физическая доставка электронных продуктов (например, наземным способом), имеющихся в корзине.

Объявление:

- `function GetAllowEgoodsShipping()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **char**. **Y**, если покупателю нужна физическая доставка электронных продуктов. Любое другое значение – если физическая доставка не нужна.

## **Cart::GetCurrentCarts**

Назначение:

- Извлечение списка заказанных товаров, находящихся в данный момент в корзинах покупателей.

Объявление:

- `function GetCurrentCarts()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - **prod\_id** – см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **variant\_id** - см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **options** - см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **qty** - см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **datetime** - см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **visitor\_id** - см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **session\_id** - см. описание одноименного поля в таблице [proshop\\_carts](#);
  - **user\_first\_name** – имя покупателя, если он зарегистрирован в магазине;

- **user\_last\_name** - фамилия покупателя, если он зарегистрирован в магазине;
- **user\_email** – e-mail покупателя, если он зарегистрирован в магазине;
- **user\_id** – идентификатор покупателя из таблицы [proshop\\_users](#), если он зарегистрирован в магазине.

## **Cart::GetCurrentCartsCount**

Назначение:

- Определение кол-ва открытых корзин покупателей в данный момент времени.

Объявление:

- `function GetCurrentCartsCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во открытых корзин.

## **Cart::GetOrderID**

Назначение:

- Определение идентификатора заказа для корзины.

Объявление:

- `function GetOrderID()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Номер заказа (хранится в таблице [proshop\\_orders](#)), или 0, если заказ еще не создан.

## **Cart::GetPaymentID**

Назначение:

- Определение идентификатора платежной системы (способа платежа), используемой покупателем для оплаты заказа.

Объявление:

- `function GetPaymentID()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Идентификатор платежной системы (хранятся в таблице [proshop\\_payments](#)), или 0, если способ платежа еще не выбран покупателем.

## **Cart::GetProdsCount**

Назначение:

- Определение кол-ва позиций, находящихся в данный момент в корзине покупателя.

Объявление:

- `function GetProdsCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во позиций.

## **Cart::GetProduct**

Назначение:

- Извлечение информации о продукте, находящимся в корзине.

Объявление:

- `function GetProduct($cart_id)`

Параметры:

- **int \$cart\_id**. Идентификатор элемента в корзине.

Возвращаемое значение:

- Тип **array**. Массив, содержащий следующие ключи:
  - **data** – массив с информацией о продукте, ключи которого совпадают с полями таблицы [proshop\\_products](#);
  - **qty** – кол-во экземпляров продукта, расположенных в корзине;
  - **cat\_name** – название категории, соответствующей продукту;
  - **cat\_id** – идентификатор категории, соответствующей продукту;
  - **options** – см. описание параметров [Cart::AddProduct](#);
  - **pending** – см. описание параметров [Cart::AddProduct](#);
  - **pin\_code** - см. описание поля **egood\_pin\_code** таблицы [proshop\\_carts](#);
  - **pos** – см. описание параметров [Cart::AddProduct](#).

## **Cart::GetProds**

Назначение:

- Извлечение информации о продуктах, находящихся в корзине.

Объявление:

- `function GetProds()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, состоящий из элементов, ключом каждого из которых является идентификатор элемента, а значением – массив, который может быть получен с помощью [Cart::GetProduct](#).

## **Cart::GetProdPrice**

Назначение:

- Определение цены продукта, находящегося в корзине, с учетом его особенностей в виде опций (см. параметр \$options в [Cart::AddProduct](#)) и их модификаторов цен.

Объявление:

- `function GetProdPrice($cart_id)`

Параметры:

- **int \$cart\_id**. Идентификатор элемента корзины.

Возвращаемое значение:

- Тип **float**. Цена продукта в базовой валюте магазина.

## **Cart::GetQty**

Назначение:

- Определение общего кол-ва экземпляров продуктов, находящихся в корзине.

Объявление:

- `function GetQty($prod_id = 0, $options = "")`

Параметры:

- **int \$prod\_id**. Идентификатор продукта, для которого необходимо определить кол-во экземпляров, имеющихся в корзине. Если равен 0, учитываются все продукты в корзине.
- **array \$options**. Массив идентификаторов вариантов опций продукта, который формирует комбинацию опций продукта, для которой необходимо определить кол-во экземпляров в корзине. Варианты опций продукта хранятся в таблице [proshop\\_products\\_options\\_vars](#). Опции продукта хранятся в таблице [proshop\\_products\\_options](#). Если продукт не имеет опций, данный массив должен быть пустым.

Возвращаемое значение:

- Тип **int**. Кол-во экземпляров продуктов в корзине по указанным в параметрах критериям.

## **Cart::GetRTShippingDates**

Назначение:

- Считывание сохраненных ранее дат доставок (работающих в режиме реального времени, например, FedEx) из кэша корзины для указанной службы доставки.

Объявление:

- `function GetRTShippingDates($name)`

Параметры:

- **string \$name**. Имя службы доставки.

Возвращаемое значение:

- Тип **string**. Диапазон предполагаемых дат доставки заказа в текстовом виде, или пустая строка, если даты определить не удалось.

## **Cart::GetRTShippingPrice**

Назначение:

- Считывание сохраненной ранее стоимости доставки (работающей в режиме реального времени, например, FedEx) из кэша корзины для указанной службы доставки.

Объявление:

- `function GetRTShippingPrice($name)`

Параметры:

- **string \$name**. Имя службы доставки.

Возвращаемое значение:

- Тип **float**. Значение стоимости доставки в базовой валюте магазина.

## **Cart::GetShipping**

Назначение:

- Извлечение информации о службе доставки, указанной покупателем в процессе оформления заказа.

Объявление:

- `function GetShipping()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, состоящий из следующих ключей:
  - **id** - идентификатор службы доставки в одном из следующих вариантов:
    - **r\_[id]**, где id – идентификатор службы доставки реального времени (хранятся в таблице [proshop\\_rtshippings](#));
    - **c\_[id]**, где id – идентификатор службы доставки выборочного вида (такой, как Почта России). Такие идентификаторы хранятся в таблице [proshop\\_cshippings](#);
    - **s\_[id]**, где id – идентификатор статичного способа доставки, созданного администратором через админку. Такие идентификаторы хранятся в таблице [proshop\\_shippings](#).
  - **caption** – название способа доставки на текущем языке;
  - **cost** – стоимость доставки данным способом в базовой валюте.

## **Cart::GetSum**

Назначение:

- Определение общей суммы всех товаров в корзине без учета стоимости доставки.

Объявление:

- `function GetSum()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **float**. Сумма корзины в базовой валюте.

## **Cart::GetWeight**

Назначение:

- Извлечение общего веса всех товаров (определенного и установленного с помощью метода [Cart::SetWeight](#)) из кэша корзины.

Объявление:

- `function GetWeight()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **float**. Общий вес всех товаров корзины в килограммах.

## **Cart::IsOnlyEgoods**

Назначение:

- Определение, содержит ли корзина только цифровые товары в данный момент.

Объявление:

- `function IsOnlyEgoods()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. Функция возвращает true, если в корзине содержатся только цифровые товары.

## **Cart::ReturnExpired**

Назначение:

- Возврат товара из просроченной корзины на склад.

Объявление:

- `function ReturnExpired($hours, $prod_id = 0)`

Параметры:

- **int \$hours**. Кол-во часов, по истечении которых корзина считается просроченной.
- **int \$prod\_id**. Идентификатор продукта (из таблицы [proshop\\_products](#)), возврат которого необходимо осуществить, если он содержится в корзине. Если данный параметр равен 0, просроченность корзины проверяется для любого товара.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SaveComment**

Назначение:

- Сохранение комментария покупателя к заказу в корзине.

Объявление:

- `function SaveComment($comment)`

Параметры:

- **string \$comment**. Текстовый комментарий покупателя.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SetAllowEgoodsShipping**

Назначение:

- Установка значения флага, который показывает, нужна ли покупателю физическая доставка электронных продуктов (например, наземным способом), имеющихся в корзине.

Объявление:

- `function SetAllowEgoodsShipping($allow)`

Параметры:

- **char \$allow**. Значение флага - **Y**, если покупателю нужна физическая доставка электронных продуктов. Любое другое значение – если физическая доставка не нужна.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SetOrderID**

Назначение:

- Сохранение идентификатора заказа в корзине.

Объявление:

- `function SetOrderID($order_id)`

Параметры:

- **int \$order\_id**. Номер заказа (хранится в таблице [proshop\\_orders](#)).

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SetPaymentID**

Назначение:

- Установка идентификатора платежной системы (способа платежа), используемой покупателем для оплаты заказа.

Объявление:

- `function SetPaymentID($payment_id)`

Параметры:

- **int \$payment\_id.** Идентификатор платежной системы (хранится в таблице [proshop\\_payments](#)).

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SetRTShippingDates**

Назначение:

- Сохранение диапазона дат доставок (работающих в режиме реального времени, например, FedEx) в кэш корзины для указанной службы доставки.

Объявление:

- `function SetRTShippingDates($name, $dates)`

Параметры:

- **string \$name.** Имя службы доставки.
- **string \$dates.** Диапазон предполагаемых дат доставки заказа в текстовом виде.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SetRTShippingPrice**

Назначение:

- Сохранение стоимости доставки (работающей в режиме реального времени, например, FedEx) в кэш корзины для указанной службы доставки.

Объявление:

- `function SetRTShippingPrice($name, $price)`

Параметры:

- **string \$name.** Имя службы доставки.
- **float \$price.** Стоимость доставки в базовой валюте.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::SetShipping**

Назначение:

- Установка информации о службе доставки, указанной покупателем в процессе оформления заказа.

Объявление:

- `function SetShipping($shipping)`

Параметры:

- **array \$shipping.** См. возвращаемое значение метода [Cart::GetShipping](#).

Возвращаемое значение:

- Функция ничего не возвращает.



## **Cart::SetWeight**

Назначение:

- Определение общего веса всех товаров в корзине и его сохранение в кэше корзины. Сохраненное значение может быть извлечено с помощью метода [Cart::GetWeight](#).

Объявление:

- `function SetWeight()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Cart::UpdateProduct**

Назначение:

- Обновление информации о продукте в корзине покупателя.

Объявление:

- `function UpdateProduct($cart_id, $qty, $options)`

Параметры:

- **int \$cart\_id**. Идентификатор элемента корзины.
- **int \$qty**. Кол-во экземпляров продукта.
- **array \$options**. Массив идентификаторов вариантов опций продукта. Варианты опций продукта перечислены в таблице [proshop\\_products\\_options\\_vars](#). Опции продукта перечислены в таблице [proshop\\_products\\_options](#). Если продукт не имеет опций, данный массив должен быть пустым.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс CartesianProduct**

Предназначен для декартового произведения множеств (получения всех возможных комбинаций значений, указанных в заданных массивах).

### **CartesianProduct::Get**

Назначение:

- Декартово произведение указанных множеств и получение результата в виде массива всех возможных комбинаций, полученных в результате произведения.

Объявление:

- `public static function Get($data)`

Параметры:

- **array \$data**. Массив множеств, которые нужно перемножить. Каждое множество также представляет собой массив значений. Таким образом, данный параметр является массивом массивов (множеств).

Возвращаемое значение:

- Тип **array**. Массив всех полученных комбинаций значений (результат декартового произведения).

## Класс **Categories**

Предназначен для управления категориями магазина.

### **Categories::Add**

Назначение:

- Добавление информации о категории в БД.

Объявление:

- `function Add($info)`

Параметры:

- **array \$info**. Массив информации о категории. Содержит следующие ключи:
  - **cat** – идентификатор родительской категории. Если категория является корневой, значение должно быть равно 0;
  - **eng\_name** – служебное имя категории (оно же – имя в URL для modRewrite);
  - **lang\_data** – массив с языкозависимыми текстами (название и описание категории и т.д.). См. пример.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданной категории.

Пример:

- ```
// ...
foreach ($LANGUAGES as $lang) {
    $lang_data[] = Array(
        'lang_id' => $lang['id'],
        'caption' => $caption,
        'descr' => $descr,
        'alt' => $alt
    );
}

$info = array(
    'cat' => $cat,
    'eng_name' => $eng_name,
    'lang_data' => $lang_data
```

```
);  
  
$new_cat_id = Categories::Add($info);  
// ...
```

## **Categories::ContainsProduct**

Назначение:

- Определение, принадлежит ли указанный продукт указанной категории.

Объявление:

- function ContainsProduct(\$cat\_id, \$prod\_id)

Параметры:

- **int \$cat\_id**. Идентификатор категории.
- **int \$prod\_id**. Идентификатор продукта.

Возвращаемое значение:

- Тип **bool**. True, если продукт принадлежит категории. Иначе – false.

## **Categories::Delete**

Назначение:

- Удаление категории и всех ее подкатегорий.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор категории.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## **Categories::DeleteImage**

Назначение:

- Удаление изображения указанной категории.

Объявление:

- function DeleteImage(\$id)

Параметры:

- **int \$id**. Идентификатор категории.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## **Categories::GetAll**

Назначение:

- Извлечение подкатегорий указанной категории из БД.

Объявление:

- `function GetAll($cat, $lang_id, $only_visible = false)`

Параметры:

- **int \$cat**. Категория, подкатегории которой требуется извлечь. Если данный параметр равен 0, извлекаются все корневые категории.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие информации о категории (название, описание и т.д.). Если данный параметр равен 0, текстовые составляющие не извлекаются.
- **bool \$only\_visible**. Извлекать ли только видимые (активные) категории (**true**), или все подряд (**false**).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом со следующими ключами:
  - все поля таблицы [proshop\\_categories](#);
  - текстовые поля таблицы [proshop\\_categories texts](#);
  - **subcats\_num** – кол-во подкатегорий категории.

Пример:

- ```
$rows = Categories::GetAll(0, $SHOP_LANGUAGE['id'], true);
foreach ($rows as $row) {
    // $row['id'] – идентификатор категории
    // $row['img'] – имя файла изображения категории
    // $row['caption'] – название категории
    // ...
}
```

## **Categories::GetAllFloat**

Назначение:

- Извлечение всех категорий магазина и их названий в неиерархическом виде, т.е. в виде простого списка.

Объявление:

- `function GetAllFloat($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь названия категорий.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом со следующими ключами:
  - все поля таблицы [proshop\\_categories](#);
  - **caption** – название категории;
  - **has\_subcats** – true, если категория имеет подкатегории. Иначе – false.

Пример:

- ```
$rows = Categories::GetAllFloat($SHOP_LANGUAGE['id']);
```

```
foreach ($rows as $row) {  
    // $row['id'] – идентификатор категории  
    // $row['img'] – имя файла изображения категории  
    // $row['caption'] – название категории  
    // ...  
}
```

## **Categories::GetDiscountByCat**

Назначение:

- Определение идентификатора скидки по умолчанию для указанной категории.

Объявление:

- `function GetDiscountByCat($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории.

Возвращаемое значение:

- Тип **int**. Идентификатор скидки по умолчанию (скидки хранятся в таблице [proshop\\_discounts](#)).

## **Categories::GetIdByName**

Назначение:

- Определение идентификатора категории по ее уникальному служебному имени (имени в URL для modRewrite).

Объявление:

- `function GetIdByName($eng_name)`

Параметры:

- **string \$eng\_name**. Служебное имя категории (имя в URL для modRewrite) из таблицы [proshop\\_categories](#). Данный параметр может также быть идентификатором категории.

Возвращаемое значение:

- Тип **int**. Идентификатор категории из таблицы [proshop\\_categories](#) или 0, если категория не найдена.

## **Categories::GetNameById**

Назначение:

- Определение служебного имени категории (имени в URL для modRewrite) по ее идентификатору. Если не удастся определить имя категории, возвращается ее идентификатор.

Объявление:

- `function GetNameById($id)`

Параметры:

- **int \$id**. Идентификатор категории.

Возвращаемое значение:

- Тип **string**. Имя в URL (для modRewrite) категории.

## **Categories::GetOne**

Назначение:

- Извлечение указанной категории из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор категории.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие категории (название, описание и т.д.). Если данный параметр равен 0, текстовые составляющие не извлекаются.

Возвращаемое значение:

- Тип **array**. Массив, ключами которого являются:
  - все поля таблицы [proshop\\_categories](#);
  - текстовые поля таблицы [proshop\\_categories\\_texts](#).

Пример:

- ```
// ...
$row = Categories::GetOne($cat_id, $SHOP_LANGUAGE['id']);
// $row['id'] – идентификатор категории
// $row['img'] – имя файла изображения категории
// $row['caption'] – название категории
// ...
```

## **Categories::GetPath**

Назначение:

- Извлечение пути к указанной категории (путь категории – совокупность предшествующих ей родительских категорий, включая саму категорию).

Объявление:

- `function GetPath($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор категории.
- **int \$lang\_id**. Идентификатор языка, для которого будут извлечены названия родительских категорий в пути. Если значение данного параметра равно 0, в возвращаемом массиве вместо ключа **caption** будет возвращен ключ **eng\_name** (служебное имя категории).

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе - массив, каждый элемент которого является массивом со следующими ключами:
  - **id** – идентификатор категории;

- **caption** – название категории. Если название категории пустое, возвращается ее служебное имя (имя в URL для modRewrite);
- **eng\_name** – служебное имя категории. Данный ключ присутствует в массиве только в том случае, если значение параметра **\$lang\_id** было равно 0.

Пример:

- ```
$path = Categories::GetPath($cat_id);  
foreach ($path as $item) {  
    // $item['id'] – идентификатор одной из родительских категорий  
    // $item['eng_name'] – служебное имя одной из родительских категорий  
}
```

## **Categories::GetSubcatsCount**

Назначение:

- Подсчет кол-ва дочерних категорий указанной категории.

Объявление:

- `function GetSubcatsCount($id)`

Параметры:

- **int \$id**. Идентификатор категории, кол-во дочерних категорий которой необходимо подсчитать.

Возвращаемое значение:

- Тип **int**. Кол-во дочерних категорий.

## **Categories::GetSubIds**

Назначение:

- Извлечение массива идентификаторов всех подкатегорий указанной категории, исключая идентификатор самой категории.

Объявление:

- `function GetSubIds($id)`

Параметры:

- **int \$id**. Идентификатор категории, для которой требуется извлечь идентификаторы ее подкатегорий (всех вложенных в нее категорий).

Возвращаемое значение:

- Тип **array**. Массив идентификаторов подкатегорий.

## **Categories::GetTaxByCat**

Назначение:

- Определение идентификатора налога по умолчанию для указанной категории.

Объявление:

- `function GetTaxByCat($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории.

Возвращаемое значение:

- Тип **int**. Идентификатор налога по умолчанию (налоги хранятся в таблице [proshop\\_taxes](#)).

## **Categories::GetTotalCount**

Назначение:

- Определение общего кол-ва всех категорий магазина.

Объявление:

- `function GetTotalCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во категорий в базе магазина.

## **Categories::IdToCaption**

Назначение:

- Определение названия указанной категории для указанного языка.

Объявление:

- `function IdToCaption($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор категории, название которой необходимо определить.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо определить название.

Возвращаемое значение:

- Тип **string**. Название категории на указанном языке. Если название пусто, будет возвращено служебное имя категории (имя в URL для modRewrite).

## **Categories::IdToCaptions**

Назначение:

- Определение пути к указанной категории товаров на указанном языке (извлечение значения поля **path** таблицы [proshop\\_categories\\_paths\\_texts](#) для указанной категории на указанном языке).

Объявление:

- `function IdToCaptions($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор категории из таблицы [proshop\\_categories](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string**. Путь к указанной категории на указанном языке.



## Categories::IsCaptionExists

Назначение:

- Определение, существует ли (занято ли) уже название категории в БД.

Объявление:

- `function IsCaptionExists($cat, $caption, $lang_id, $id = 0)`

Параметры:

- **int \$cat**. Идентификатор категории, в пределах которой будет осуществлен поиск дочерней категории с заданным названием. Данный параметр игнорируется, если значение параметра **\$id** больше нуля. Если значение параметра **\$cat** равно нулю, поиск будет осуществлен среди корневых категорий.
- **string \$caption**. Название, которое необходимо проверить.
- **int \$lang\_id**. Идентификатор языка, для которого проверяется название.
- **int \$id**. Идентификатор категории, которая не будет учтена при поиске (это нужно, когда модуль сохраняет то же самое название категории, добавленной ранее). Если равен 0, будут учитываться все категории.

Возвращаемое значение:

- Тип **bool**. True, если название категории существует (уже занято). Иначе – false.

## Categories::IsExists

Назначение:

- Определение, существует ли категория с заданным идентификатором в таблице [proshop\\_categories](#).

Объявление:

- `function IsExists($id)`

Параметры:

- **int \$id**. Идентификатор категории.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## Categories::IsNameExists

Назначение:

- Определение, существует ли (занято ли) уже служебное имя категории (имя в URL для `modRewrite`) в БД.

Объявление:

- `function IsNameExists($name, $id = 0)`

Параметры:

- **string \$name**. Имя, которое необходимо проверить.

- **int \$id**. Идентификатор категории, которая не будет учтена при поиске (это нужно, когда модуль сохраняет то же самое служебное имя категории, добавленной ранее). Если равен 0, будут учитываться все категории.

Возвращаемое значение:

- Тип **bool**. True, если имя категории существует (уже занято). Иначе – false.

## **Categories::IsVisible**

Назначение:

- Определение, видна ли категория покупателю.

Объявление:

- `function IsVisible($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, видимость которой необходимо определить.

Возвращаемое значение:

- Тип **bool**. True, если категория (и все ее родительские категории) видна покупателю. Иначе – false.

## **Categories::ModPrices**

Назначение:

- Модификация цен всех товаров указанной категории (и всех ее подкатегорий) на заданный процент (как в большую, так и в меньшую сторону). Модифицирует поля **price** и **prev\_price** таблицы [proshop\\_products](#).

Объявление:

- `function ModPrices($cat_id, $price_mod)`

Параметры:

- **int \$cat\_id**. Идентификатор категории (из таблицы [proshop\\_categories](#)), цены товаров которой нужно модифицировать.
- **float \$price\_mod**. Значение процента, на который необходимо модифицировать цены товаров. Может быть как положительным, так и отрицательным.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Categories::PathToCaptions**

Назначение:

- Конвертирование пути к категории (поле **path** таблицы [proshop\\_categories](#)) в путь с названиями родительских категорий на заданном языке.

Объявление:

- `function PathToCaptions($path, $lang_id)`

Параметры:

- **string \$path**. Путь к категории, который необходимо преобразовать.
- **int \$lang\_id**. Язык, для которого необходимо получить названия категорий в пути.

Возвращаемое значение:

- Тип **string**. Путь к категории, например: **Книги/Художественная литература**.

## **Categories::RecalcProductsCount**

Назначение:

- Пересчет показателей кол-ва продуктов во всех товарных категориях магазина.

Объявление:

- `function RecalcProductsCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/False – в зависимости от успешности операции.

## **Categories::SetDiscount**

Назначение:

- Установка идентификатора скидки по умолчанию для указанной категории (и для всех ее подкатегорий).

Объявление:

- `function SetDiscount($discount_id, $cat_id)`

Параметры:

- **int \$discount\_id**. Идентификатор скидки (хранится в таблице [proshop\\_discounts](#)).
- **int \$cat\_id**. Идентификатор категории.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив идентификаторов подкатегорий указанной категории, включая саму категорию.

## **Categories::SetImage**

Назначение:

- Установка имени файла изображения указанной категории.

Объявление:

- `function SetImage($id, $img)`

Параметры:

- **int \$id**. Идентификатор категории из таблицы [proshop\\_categories](#).
- **string \$img**. Имя файла изображения категории.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Categories::SetPos**

Назначение:

- Установка позиции (поле **pos** таблицы [proshop\\_categories](#)) указанной категории относительно других категорий того же уровня. Применяется при сортировке категорий.

Объявление:

- `function SetPos($cat_id, $pos)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, позицию которой необходимо установить.
- **int \$pos**. Положительное целое значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Categories::SetProductsCount**

Назначение:

- Определение общего кол-ва продуктов и кол-ва видимых покупателю продуктов для заданной категории и запись соответствующих значений в поля **prods\_num**, **visible\_prods\_num**, **this\_prods\_num**, **this\_visible\_prods\_num** таблицы [proshop\\_categories](#).

Объявление:

- `function SetProductsCount($cat_id, $recursive = true)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, для которой необходимо подсчитать и обновить значения.
- **bool \$recursive**. Параметр для внутреннего использования в самой функции. При применении (вызове) программистом должен быть всегда **true**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Categories::SetTax**

Назначение:

- Установка идентификатора налога по умолчанию для указанной категории (и для всех ее подкатегорий).

Объявление:

- `function SetTax($tax_id, $cat_id)`

Параметры:

- **int \$tax\_id**. Идентификатор налога (хранится в таблице [proshop\\_taxes](#)).
- **int \$cat\_id**. Идентификатор категории.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив идентификаторов подкатегорий указанной категории, включая саму категорию.

## **Categories::setVisible**

Назначение:

- Установка флага видимости категории покупателю.

Объявление:

- `function setVisible($id, $mode)`

Параметры:

- **int \$id**. Идентификатор категории.
- **char \$flag**. Флаг видимости категории. Если равен **Y**, категория будет видима. Если **N** - не видима.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## **Categories::Update**

Назначение:

- Обновление информации об указанной категории в БД.

Объявление:

- `function Update($id, $info, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор категории, информацию о которой необходимо обновить.
- **array \$info**. Массив информации, в который входят следующие ключи:
  - **cat** – идентификатор родительской категории;
  - **eng\_name** – служебное имя категории (имя в URL для modRewrite);
  - **caption** – название категории на указанном языке (если **\$lang\_id** задан);
  - **descr** – описание категории (может содержать HTML) на указанном языке (если **\$lang\_id** задан);
  - **alt** – альтернативный текст изображения категории (на указанном языке), которое устанавливается с помощью [Categories::UploadImage](#) или [Categories::SetImage](#) (если **\$lang\_id** задан).
- **int \$lang\_id**. Идентификатор языка, для которого устанавливаются текстовые составляющие описания категории из массива **\$info**. Если данный параметр не задан, текстовые составляющие не устанавливаются.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Categories::UploadImage**

Назначение:

- Загрузка изображения категории в магазин.

Объявление:

- `function UploadImage($cat_id, $img, $width)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, для которой требуется загрузить изображение.
- **file \$img**. Описание загружаемого файла изображения (из переменной `$_FILES`).
- **int \$width**. Ширина, которую примет изображение (с сохранением пропорций) после загрузки в магазин.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

Пример:

- `UploadImage($cat_id, $_FILES['new_img'], $CONFIG['settings']['category_img_width']);`

## Класс CBlocks

Предназначен для управления боковыми блоками витрины.

### **CBlocks::GetAll**

Назначение:

- Извлечение боковых блоков с информацией о них из БД. Список блоков отсортирован по их весу (см. поле **weight** таблицы [proshop\\_cblocks](#)).

Объявление:

- `function GetAll($ctHEME, $column = "", $module = "", $load_banners = true)`

Параметры:

- **string \$ctHEME**. Название темы оформления, для которой необходимо извлечь набор блоков.
- **char \$column**. Колонка, для которой загрузить список блоков (**L** - левая, **C** – центральная, **R** - правая). Данное значение может быть пустым (что означает загрузить список всех блоков, независимо от колонки).
- **string \$module**. Название модуля, который вызывает данную функцию. Данное значение может быть пустым (не учитывать модуль). Данный параметр нужен, чтобы правильно определить список блоков при загрузке того или иного модуля.
- **bool \$load\_banners**. Учитывать ли баннерные блоки (принадлежащие модулю **banners**).

Возвращаемое значение:

- Тип **array**. Массив, содержащий ключи, соответствующие именам блоков (полю **name** таблицы [proshop\\_cblocks](#)). Каждый элемент массива содержит информацию из таблицы [proshop\\_cblocks](#). См. пример.

Пример:

- ```
$lblocks = CBlocks::GetAll('L', $_REQUEST['module'], false);
foreach ($lblocks as $name => $lblock) {
    // $name – имя блока
    // $lblock['label'] – метка блока
```

```
// ...  
}
```

## **СBlocks::GetColumn**

Назначение:

- Определение колонки, в которой расположен указанный блок.

Объявление:

- `function GetColumn($id)`

Параметры:

- **int \$id**. Идентификатор блока из таблицы [proshop\\_cblocks](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **СBlocks::IsColumnFixed**

Назначение:

- Определение, можно ли изменять расположение указанного блока (его колонку).

Объявление:

- `function IsColumnFixed($id)`

Параметры:

- **int \$id**. Идентификатор блока из таблицы [proshop\\_cblocks](#).

Возвращаемое значение:

- Тип **bool**. True, если изменять расположение нельзя. Иначе – можно.

## **СBlocks::SetColumn**

Назначение:

- Установка колонки блока (в какой колонке блок будет отображаться).

Объявление:

- `function SetColumn($id, $column)`

Параметры:

- **int \$id**. Идентификатор блока.
- **char \$column**. Колонка, в которой будет отображаться блок (**L** – левая, **C** – центральная, **R** – правая).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **СBlocks::SetInHomepage**

Назначение:

- Установка флага отображения блока, означающего, что блок должен отображаться на главной странице магазина.

Объявление:

- `function SetInHomepage($id, $show)`

Параметры:

- **int \$id**. Идентификатор блока.
- **bool \$show**. Отображать блок на главной странице или нет.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **CBlocks::SetNotInHomepage**

Назначение:

- Установка флага отображения блока, означающего, что блок должен отображаться на страницах, отличных от главной страницы магазина.

Объявление:

- `function SetNotInHomepage($id, $show)`

Параметры:

- **int \$id**. Идентификатор блока.
- **bool \$show**. Отображать блок на неглавных страницах магазина или нет.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **CBlocks::SetWeight**

Назначение:

- Установка веса блока.

Объявление:

- `function SetWeight($id, $weight)`

Параметры:

- **int \$id**. Идентификатор блока.
- **int \$weight**. Значение веса.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Config**

Предназначен для управления настройками магазина, для загрузки настроек и списка модулей магазина.

### **Config::GetAllGroups**

Назначение:



- Извлечение из БД групп настроек. Каждая настройка принадлежит конкретной группе (например, группа «Продукты»).

Объявление:

- `function GetAllGroups()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_settings\\_groups](#). Если группа настроек имеет подгруппы, то к ключам массива добавляется еще один ключ **sections**, значение которого является массивом, каждый элемент которого содержит описание группы в виде массива с ключами, совпадающими с полями таблицы [proshop\\_settings\\_groups](#). См. пример.

Пример:

- ```
$groups = Config::GetAllGroups();
foreach ($groups as $group) {
    // $group['caption'] – название группы
    if (is_array($group['sections'])) {
        foreach ($group['sections'] as $section) {
            // $section['caption'] – название подгруппы
            // ...
        }
    }
}
```

## **Config::GetAllSettings**

Назначение:

- Извлечение настроек указанной группы.

Объявление:

- `function GetAllSettings($group_id)`

Параметры:

- **int \$group\_id**. Идентификатор группы из таблицы [proshop\\_settings\\_groups](#), для которой требуется получить массив настроек.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_settings](#), содержащим также ключ **type** (значение поля **name** из таблицы [proshop\\_settings\\_types](#)). См. пример.

Пример:

- `// ...`

```
$settings = Config::GetAllSettings($group_id);
foreach ($settings as $val) {
    // $val['name'] – имя настройки
    // $val['value'] – значение настройки
    // ...
    // $val['type'] – тип настройки (из таблицы proshop\_settings\_types)
}
// ...
```

## **Config::GetFullSettings**

Назначение:

- Извлечение всех настроек, содержащихся в таблице [proshop\\_settings](#).

Объявление:

- function GetFullSettings()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_settings](#).

## **Config::GetSettingByName**

Назначение:

- Извлечение настройки из БД по ее имени.

Объявление:

- function GetSettingByName(\$name)

Параметры:

- **string \$name**. Имя настройки (совпадающее со значением поля name в таблице [proshop\\_settings](#)).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив с ключами, совпадающими с полями таблицы [proshop\\_settings](#).

## **Config::GetSettingType**

Назначение:

- Извлечение информации о типе настройки из таблицы [proshop\\_settings\\_types](#).

Объявление:

- function GetSettingType(\$id)

Параметры:

- **int \$id**. Идентификатор типа настройки.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив с ключами, совпадающими с полями таблицы [proshop\\_settings\\_types](#).

## **Config::Init**

Назначение:

- Загрузка и кэширование настроек, опций и списка модулей во внутреннюю переменную **\$config** класса **Config**. Настройки загружаются из таблицы [proshop\\_settings](#) (методом [Config::LoadSettings](#)), опции – из таблицы [proshop\\_options](#) (методом [Config::LoadOptions](#)), модули – из таблицы [proshop\\_modules](#) (методом [Config::LoadModules](#)).

Объявление:

- function Init()

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **Config::LoadModules**

Назначение:

- Загрузка списка модулей во внутреннюю переменную **\$config** класса **Config**. Используется методом [Config::Init](#).

Объявление:

- function LoadModules()

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **Config::LoadOptions**

Назначение:

- Загрузка опций во внутреннюю переменную **\$config** класса **Config**. Используется методом [Config::Init](#).

Объявление:

- function LoadOptions()

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **Config::LoadSettings**

Назначение:

- Загрузка настроек во внутреннюю переменную **\$config** класса **Config**. Используется методом [Config::Init](#).

Объявление:

- `function LoadSettings()`

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **Config::UpdateSettings**

Назначение:

- Обновление настроек в базе данных и сброс их кэша.

Объявление:

- `function UpdateSettings($new_settings)`

Параметры:

- **array \$new\_settings**. Массив настроек, каждый ключ которого соответствует имени настройки, а каждое соответствующее значение – значению настройки (см. описание таблицы [proshop\\_settings](#)).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Controller**

Предназначен для подключения PHP-файлов модулей при загрузке ядра CMS.

### **Controller::Controller**

Назначение:

- Конструктор класса.

Объявление:

- `function Controller($mods_dir)`

Параметры:

- **string \$mods\_dir**. Путь к директории модулей (имеются в виду директории **admin/modules** или **modules**).

### **Controller::Run**

Назначение:

- Подключение и запуск текущего модуля, загрузка всех автозапускаемых модулей, проверка прав администратора (в админке).

Объявление:

- `function Run()`

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **Controller::SetDefModule**

Назначение:

- Установка имени модуля по умолчанию, который загружается в тех случаях, когда GET-переменная **module** не задана.

Объявление:

- `function SetDefModule($mod_name)`

Параметры:

- **string \$mod\_name**. Имя модуля.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Controller::SetModules**

Назначение:

- Сохранение списка включенных модулей во внутреннем массиве **\$mods** класса **Controller**.

Объявление:

- `function SetModules($mods)`

Параметры:

- **array \$mods**. Массив модулей, который берется из **\$CONFIG['modules']**. Из этого массива функция выделяет только те модули, параметр **avail** которых равен **Y** (т.е. только включенные администратором модули), подготавливая таким образом внутренний массив **\$mods** класса для его использования в методе [Controller::Run](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс Countries**

Предназначен для управления странами и областями, использующимися в магазине.

## **Countries::Add**

Назначение:

- Добавление страны в БД.

Объявление:

- `function Add($info)`

Параметры:

- **array \$info**. Массив с информацией о добавляемой стране. Должен содержать следующие ключи:
  - **name** – название страны. Макс. 254 символа;
  - **code** – двухбуквенный код страны. Макс. 2 символа. Буквы должны быть английскими заглавными;
  - **code\_edost** – код страны для сервиса расчета стоимости доставки eDost;
  - **pos** – позиция относительно других стран в списке (для сортировки);
  - **avail** – выводить ли страну в списке (**Y** – выводить, **N** – отключить ее вывод).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции. Операция может быть неуспешна и в том случае, если страна с заданным именем уже существует в базе.

## **Countries::AddState**

Назначение:

- Добавление области (региона страны) в БД.

Объявление:

- `function AddState($country_id, $name, $code, $code_edost, $pos)`

Параметры:

- **int \$country\_id**. Идентификатор страны, с которой необходимо связать добавляемую область.
- **string \$name**. Название области. Макс. 254 символа.
- **string \$code**. Код области (например, номер региона для России, или двухбуквенный код штата для США или Канады). Макс. 254 символа.
- **string \$code\_edost**. Код области для сервиса расчета стоимости доставки eDost.
- **int \$pos**. Позиция области относительно других областей этой же страны в списке (для сортировки).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Countries::CodeToCountry**

Назначение:

- Определение идентификатора страны по ее двухбуквенному коду.

Объявление:

- `function CodeToCountry($code)`

Параметры:

- **string \$code**. Двубуквенный код страны.

Возвращаемое значение:

- Тип **int|bool**. False – в случае ошибки. Иначе - идентификатор страны (хранится в таблице [proshop\\_countries](#)).

## **Countries::CountryToCode**

Назначение:

- Определение двубуквенного кода страны по ее идентификатору.

Объявление:

- `function CountryToCode($id)`

Параметры:

- **int \$id**. Идентификатор страны (хранится в таблице [proshop\\_countries](#)).

Возвращаемое значение:

- Тип **string|bool**. False – в случае ошибки. Иначе - двубуквенное обозначение страны (хранится в таблице [proshop\\_countries](#)).

## **Countries::Delete**

Назначение:

- Удаление указанной страны из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор удаляемой страны (хранится в таблице [proshop\\_countries](#)).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Countries::DeleteState**

Назначение:

- Удаление указанной области (региона) страны.

Объявление:

- `function DeleteState($id)`

Параметры:

- **int \$id**. Идентификатор области (хранится в таблице `proshop_states`).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Countries::GetAll**

Назначение:

- Извлечение всех стран из БД.

Объявление:

- `function GetAll($sql = ")`

Параметры:

- **string \$sql**. Часть SQL-запроса, добавляемая в конец основного SELECT-запроса (`SELECT * FROM '$TABLES['countries']`). Для сортировки или какого-нибудь условия. Если значение данного параметра не задано, к SQL-запросу добавляется оператор `ORDER BY` с сортировкой по возрастанию позиции и названия страны.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_countries](#).

## **Countries::GetAllCities**

Назначение:

- Извлечение списка городов указанной страны из таблицы [proshop\\_cities](#).

Объявление:

- `function GetAllCities($country_id)`

Параметры:

- **int \$country\_id**. Идентификатор страны (из таблицы [proshop\\_countries](#)), список городов которой необходимо извлечь.

Возвращаемое значение:

- Тип **array**. Массив названий городов указанной страны.

## **Countries::GetAllStates**

Назначение:

- Извлечение всех областей (регионов) указанной страны из БД.

Объявление:

- `function GetAllStates($country_id, $sql = ")`

Параметры:

- **int \$country\_id**. Идентификатор страны.
- **string \$sql**. Часть SQL-запроса, добавляемая в конец основного SELECT-запроса (`SELECT * FROM '$TABLES['states']`). Для сортировки или какого-нибудь условия. Если значение данного параметра не задано, к SQL-запросу добавляется оператор `ORDER BY` с сортировкой по возрастанию позиции и названия области.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе - массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_states](#).



## **Countries::GetAllVisible**

Назначение:

- Извлечение всех стран из БД, которые видны покупателю (значение поля **avail** которых в таблице [proshop\\_countries](#) равно Y).

Объявление:

- `function GetAllVisible()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_countries](#). Массив отсортирован по возрастанию позиции и названия страны.

## **Countries::GetCity**

Назначение:

- Извлечение информации об указанном городе указанной страны (по имени города) из таблицы [proshop\\_cities](#).

Объявление:

- `function GetCity($city, $country_id)`

Параметры:

- **string \$city**. Название города.
- **int \$country\_id**. Идентификатор страны (из таблицы [proshop\\_countries](#)), которой этот город принадлежит.

Возвращаемое значение:

- Тип **array**. Массив с информацией о городе, ключи которого совпадают с полями таблицы [proshop\\_cities](#).

## **Countries::GetName**

Назначение:

- Определение названия страны по ее идентификатору.

Объявление:

- `function GetName($id)`

Параметры:

- **int \$id**. Идентификатор страны (хранится в таблице [proshop\\_countries](#)).

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – название страны.

## **Countries::GetOne**

Назначение:

- Извлечение информации об указанной стране из БД.

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор страны (хранится в таблице [proshop\\_countries](#)).

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_countries](#).

## **Countries::GetOneState**

Назначение:

- Извлечение информации об указанной области (региона страны) из БД.

Объявление:

- `function GetOneState($id)`

Параметры:

- **int \$id**. Идентификатор области (хранится в таблице [proshop\\_states](#)).

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_states](#).

## **Countries::GetStateName**

Назначение:

- Определение названия области (региона) страны по идентификатору этой области.

Объявление:

- `function GetStateName($id)`

Параметры:

- **int \$id**. Идентификатор области (хранится в таблице [proshop\\_states](#)).

Возвращаемое значение:

- Тип **string/bool**. `False` – в случае ошибки. Иначе – название области.

## **Countries::IsCodeExists**

Назначение:

- Определение, занят ли уже заданный двубуквенный код страны.

Объявление:

- `function IsCodeExists($code, $id = 0)`

Параметры:

- **string \$code**. Двубуквенный код страны (хранится в таблице [proshop\\_countries](#)).

- **int \$id**. Идентификатор страны, которую не нужно учитывать при поиске заданного кода **\$code** в таблице [proshop\\_countries](#). Если значение данного параметра равно 0, учитываются все страны.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от результата.

## **Countries::IsNameExists**

Назначение:

- Определение, занято ли уже заданное название страны.

Объявление:

- `function IsNameExists($name, $id = 0)`

Параметры:

- **string \$name**. Название страны (хранится в таблице [proshop\\_countries](#)).
- **int \$id**. Идентификатор страны, которую не нужно учитывать при поиске заданного названия **\$name** в таблице [proshop\\_countries](#). Если значение данного параметра равно 0, учитываются все страны.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от результата.

## **Countries::IsStateCodeExists**

Назначение:

- Определение, занят ли уже заданный код области (региона страны).

Объявление:

- `function IsStateCodeExists($country_id, $code, $state_id = 0)`

Параметры:

- **int \$country\_id**. Идентификатор страны, в пределах которой осуществляется проверка на существование кода области.
- **string \$code**. Код области (хранится в таблице [proshop\\_states](#)).
- **int \$state\_id**. Идентификатор области, которую не нужно учитывать при поиске заданного кода **\$code** в таблице [proshop\\_states](#). Если значение данного параметра равно 0, учитываются все области заданной страны.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от результата.

## **Countries::IsStateNameExists**

Назначение:

- Определение, занято ли уже название области (региона страны).

Объявление:

- `function IsStateNameExists($country_id, $name, $state_id = 0)`

Параметры:

- **int \$country\_id**. Идентификатор страны, в пределах которой осуществляется проверка на существование названия области.
- **string \$name**. Название области (хранится в таблице [proshop\\_states](#)).
- **int \$state\_id**. Идентификатор области, которую не нужно учитывать при поиске заданного названия **\$name** в таблице [proshop\\_states](#). Если значение данного параметра равно 0, учитываются все области заданной страны.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от результата.

## **Countries::Update**

Назначение:

- Обновление информации о заданной стране в БД.

Объявление:

- function Update(\$id, \$info)

Параметры:

- **int \$id**. Идентификатор страны, информацию о которой требуется обновить в таблице [proshop\\_countries](#).
- **array \$info**. Массив с информацией, содержащий следующие ключи:
  - **name** – название страны;
  - **code** – двухбуквенный код страны;
  - **code\_edost** – код страны для сервиса расчета стоимости доставки eDost;
  - **pos** – позиция страны относительно других стран в таблице [proshop\\_countries](#) (для сортировки);
  - **avail** – выводить ли страну в списке (**Y** – выводить, **N** – отключить ее вывод).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Countries::UpdateState**

Назначение:

- Обновление информации об области (региона страны) в БД.

Объявление:

- function UpdateState(\$id, \$name, \$code, \$code\_edost, \$pos)

Параметры:

- **int \$id**. Идентификатор области, информацию о которой необходимо обновить в таблице [proshop\\_states](#).
- **string \$name**. Название области. Макс. 254 символа.
- **string \$code**. Код области (например, номер региона для России, или двухбуквенный код штата для США или Канады). Макс. 254 символа.

- **string \$code\_edost.** Код области для сервиса расчета стоимости доставки eDost.
- **int \$pos.** Позиция области относительно других областей этой же страны в списке (для сортировки).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс CThemes

Предназначен для извлечения из базы списков названий тем оформлений витрины и их цветовых схем.

### CThemes::GetAll

Назначение:

- Извлечение списка названий тем оформления и названий их цветовых схем в виде массива.

Объявление:

- function GetAll()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Ассоциативный массив тем оформления и их цветовых схем.

Пример:

- ```
$themes = CThemes::GetAll();
foreach ($themes as $theme_name => $colors) {
    // $theme_name – название темы оформления
    // $colors – массив названий цветовых схем данной темы оформления
}
```

### CThemes::GetColors

Назначение:

- Извлечение списка цветовых схем для указанной темы оформления.

Объявление:

- function GetColors(\$theme)

Параметры:

- **string \$theme.** Название темы оформления.

Возвращаемое значение:

- Тип **array**. Массив названий цветовых схем.

## **CThemes::GetFirstColor**

Назначение:

- Извлечение первого попавшегося названия цветовой схемы для указанной темы оформления.

Объявление:

- `function GetFirstColor($theme)`

Параметры:

- **string \$theme**. Название темы оформления.

Возвращаемое значение:

- Тип **string**. Название цветовой схемы.

## **CThemes::GetThemeCaption**

Назначение:

- Определение отображаемого названия темы оформления витрины по ее служебному названию.

Объявление:

- `function GetThemeCaption($theme)`

Параметры:

- **string \$theme**. Служебное название темы оформления.

Возвращаемое значение:

- Тип **string**. Отображаемое название указанной темы оформления.

## **CThemes::GetThemes**

Назначение:

- Извлечение списка названий тем оформления витрины (из таблицы [proshop\\_cthemes](#)).

Объявление:

- `function GetThemes()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив названий тем оформления, каждый элемент которого состоит из массива с ключами:
  - **theme** – служебное название темы оформления;
  - **caption** – отображаемое название темы оформления.

## **Класс Currencies**

Предназначен для управления валютами магазина.

## **Currencies::Add**

Назначение:

- Добавление валюты в базу.

Объявление:

- `function Add($info)`

Параметры:

- **array \$info**. Массив информации о валюте. Должен содержать следующие ключи:
  - **name** – трехбуквенное международное обозначение валюты (заглавными английскими буквами);
  - **short\_name** – короткая приставка к валюте (например, \$, €, р.);
  - **code** – целочисленный код валюты по ISO 4217;
  - **course** – курс валюты по отношению к базовой валюте магазина (к валюте, флаг **is\_base** которой равен **Y**);
  - **decimals** – количество знаков после запятой при отображении цен в данной валюте (может быть 0 или 2);
  - **dec\_point** – знак десятичной точки (например, запятая или точка) при отображении цен в данной валюте;
  - **thousands\_sep** – разделитель тысячных (например, пробел или просто пустая строка) при отображении цен в данной валюте;
  - **left\_side** – если значение данного ключа равно **Y**, короткая приставка (**short\_name**) к валюте отображается слева от значения цены, иначе – справа.

Возвращаемое значение:

- Тип **int/bool**. `False` – в случае ошибки. Иначе – идентификатор добавленной валюты.

## **Currencies::Delete**

Назначение:

- Удаление указанной валюты из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор удаляемой валюты.

Возвращаемое значение:

- Тип **bool**. Всегда `true`.

## **Currencies::GetAll**

Назначение:

- Извлечение всех (или только указанных) валют из БД.

Объявление:

- `function GetAll($only_this_ids = array())`

Параметры:

- **array \$only\_this\_ids**. Массив идентификаторов валют, которые необходимо извлечь. Если данный массив пуст, извлекаются все валюты из таблицы [proshop\\_currencies](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_currencies](#).

## **Currencies::GetBase**

Назначение:

- Извлечение информации о базовой валюте магазина. Базовая валюта – валюта из таблицы [proshop\\_currencies](#), значение поля **is\_base** которой равно **Y**.

Объявление:

- `function GetBase()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив с ключами, совпадающими с полями таблицы [proshop\\_currencies](#).

## **Currencies::GetBaseID**

Назначение:

- Извлечение идентификатора базовой валюты магазина. Базовая валюта – валюта из таблицы [proshop\\_currencies](#), значение поля **is\_base** которой равно **Y**.

Объявление:

- `function GetBaseID()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. **False** – в случае ошибки. Иначе – идентификатор базовой валюты (хранится в таблице [proshop\\_currencies](#)).

## **Currencies::GetOne**

Назначение:

- Извлечение информации об указанной валюте магазина (из таблицы [proshop\\_currencies](#)).

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор валюты.

Возвращаемое значение:



- Тип **array/bool**. False – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_currencies](#).

## **Currencies::GetOneByName**

Назначение:

- Извлечение информации об указанной валюте магазина (из таблицы [proshop\\_currencies](#)) по ее международному трехбуквенному обозначению.

Объявление:

- `function GetOneByName($name)`

Параметры:

- **string \$name**. Международное трехбуквенное обозначение валюты.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_currencies](#).

## **Currencies::SetBaseID**

Назначение:

- Смена базовой валюты магазина на указанную и переконвертирование всех цен в базе данных по новому курсу.

Объявление:

- `function SetBaseID($id)`

Параметры:

- **int \$id**. Идентификатор валюты, которую необходимо установить в качестве базовой.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Currencies::SetCourse**

Назначение:

- Установка курса указанной валюты по отношению к базовой валюте.

Объявление:

- `function SetCourse($id, $course)`

Параметры:

- **int \$id**. Идентификатор валюты, курс которой необходимо сменить.
- **float \$course**. Новый курс валюты по отношению к базовой (может содержать до 7-ми знаков после запятой).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Currencies::Update

Назначение:

- Обновление информации об указанной валюте в БД.

Объявление:

- function Update(\$id, \$info)

Параметры:

- **int \$id**. Идентификатор обновляемой валюты (из таблицы [proshop\\_currencies](#)).
- **array \$info**. См. описание соответствующего параметра для [Currencies::Add](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс DB

Предназначен для доступа к базе данных MySQL.

### \$DB->Close

Назначение:

- Закрытие соединения с БД.

Объявление:

- function Close()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

### \$DB->DB

Назначение:

- Конструктор класса.

Объявление:

- function DB(\$only\_connect = false, \$dbhost = "", \$dbuser = "", \$dbpass = "", \$dbname = "")

Параметры:

- **bool \$only\_connect**. Если значение данного параметра – true, производится только соединение с сервером баз данных без выбора (подключения) базы данных и без установки кодировки по умолчанию UTF-8.
- **string \$dbhost**. Имя хоста сервера БД. Если данный параметр пуст (и значения параметров \$dbuser, \$dbpass и \$dbname также пусты), используется значение константы [DB\\_HOST](#).
- **string \$dbuser**. Имя пользователя для соединения с сервером БД. Если данный параметр пуст (и значения параметров \$dbhost, \$dbpass и \$dbname также пусты), используется значение константы [DB\\_USER](#).

- **string \$dbpass.** Пароль пользователя для соединения с сервером БД. Если данный параметр пуст (и значения параметров \$dbhost, \$dbuser и \$dbname также пусты), используется значение константы [DB\\_PASS](#).
- **string \$dbname.** Имя БД. Если данный параметр пуст (и значения параметров \$dbhost, \$dbuser и \$dbpass также пусты), используется значение константы [DB\\_NAME](#).

## **\$DB->Delete**

Назначение:

- Удаление записей, соответствующих заданным условиям, из указанной таблицы.

Объявление:

- `function Delete($table, $conditions)`

Параметры:

- **string \$table.** Имя таблицы БД.
- **array \$conditions.** Названия и значения полей таблицы, при совпадении которых записи будут удалены. См. пример.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

Пример:

- ```
if ($DB->Delete($TABLES['articles'], array('id' => $article_id))) {  
    // Статья с идентификатором $article_id удалена  
} else {  
    // Произошла ошибка удаления статьи  
}
```

## **\$DB->GetAssocs**

Назначение:

- Выполнение запроса SELECT с последующим извлечением результатов выборки в виде массива.

Объявление:

- `function GetAssocs($sql)`

Параметры:

- **string \$sql.** Текст SELECT-запроса.

Возвращаемое значение:

- Тип **array**. Массив, каждым элементом которого является массив с ключами, совпадающими с полями, перечисленными в SELECT-запросе. См. пример.

Пример:

- ```
$sql = 'SELECT * FROM '.$TABLES['articles'];  
$rows = $DB->GetAssocs($sql);  
foreach ($rows as $row) {
```

```
// $row['id'] – идентификатор записи
// ...
}
```

## **\$DB->GetError**

Назначение:

- Получение текстового описания ошибки, возникшей после выполнения очередного SQL-запроса.

Объявление:

- `function GetError()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. HTML-строка с описанием возникшей ошибки.

## **\$DB->GetLastId**

Назначение:

- Определение идентификатора последней вставленной с помощью INSERT-запроса записи.

Объявление:

- `function GetLastId()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Идентификатор последней вставленной записи.

## **\$DB->GetRow**

Назначение:

- Выполнение запроса SELECT с последующим извлечением одного-единственного результата выборки в виде массива.

Объявление:

- `function GetRow($sql, $type = 'assoc')`

Параметры:

- **string \$sql**. Текст SELECT-запроса.
- **string \$type**. В каком виде извлекать данные (**array** – эквивалент функции **mysql\_fetch\_array**, **row** – эквивалент функции **mysql\_fetch\_row**, **assoc** – эквивалент функции **mysql\_fetch\_assoc**).

Возвращаемое значение:

- Тип **array**. Массив, содержимое которого зависит от параметра \$type.

Пример:

- `$row = $DB->GetRow('SELECT * FROM '.$TABLES['articles'].' WHERE id = '.$id);`  
`// $row['url_name'] - ...`

## **`$DB->GetRows`**

Назначение:

- Выполнение запроса `SELECT` с последующим извлечением результатов выборки в виде массива.

Объявление:

- `function GetRows($sql)`

Параметры:

- **string \$sql**. Текст `SELECT`-запроса.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого являются номерами извлекаемых полей (см. описание функции `mysql_fetch_row`), затрагиваемых `SELECT`-запросом.

## **`$DB->GetRowsNum`**

Назначение:

- Определение кол-ва записей, получаемых с помощью заданного `SELECT`-запроса.

Объявление:

- `function GetRowsNum($sql)`

Параметры:

- **string \$sql**. Текст `SELECT`-запроса.

Возвращаемое значение:

- Тип **int**. См. описание возвращаемого значения функции `mysql_num_rows`.

## **`$DB->GetServerVersion`**

Назначение:

- Определение версии используемого MySQL-сервера.

Объявление:

- `function GetServerVersion()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string/bool**. `False` – в случае ошибки (например, соединение с сервером не установлено). Иначе – строка с номером версии сервера, к которому заранее было установлено подключение.

## **\$DB->Insert**

Назначение:

- Формирование и выполнение INSERT-запроса.

Объявление:

- function Insert(\$table, \$fields)

Параметры:

- **string \$table**. Название таблицы, в которую необходимо вставить данные.
- **array \$fields**. Массив, ключами которого являются названия полей, а соответствующими значениями – значения этих полей, которые нужно установить.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор вставленной в указанную таблицу записи.

## **\$DB->IsConnected**

Назначение:

- Определение, установлено ли подключение к серверу БД у данного экземпляра класса **DB**.

Объявление:

- function IsConnected()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **\$DB->IsError**

Назначение:

- Определение, произошла ли ошибка при выполнении последнего SQL-запроса.

Объявление:

- function IsError()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **\$DB->Query**

Назначение:

- Выполнение SQL-запроса (аналог функции **mysql\_query**).

Объявление:

- function Query(\$sql)

Параметры:

- **string \$sql**. Текст SQL-запроса.

Возвращаемое значение:

- Тип **resource**. См. возвращаемое значение функции **mysql\_query**.

## **\$DB->Replace**

Назначение:

- Формирование и выполнение запроса REPLACE.

Объявление:

- `Replace($table, $conditions, $replacements)`

Параметры:

- **string \$table**. Имя таблицы, над которой следует выполнить REPLACE-запрос.
- **array \$conditions**. Массив имен (имя поля – в качестве ключа массива) и значений полей, которые формируют условия выполнения REPLACE-запроса. См. пример.
- **array \$replacements**. Массив имен полей (имя поля – в качестве ключа массива) и новых значений этих полей. См. пример.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

Пример:

- `$DB->Replace($TABLES['articles'], array('id' => $article_id), array('id' => $article_id, 'url_name' => $url_name));`

## **\$DB->Select**

Назначение:

- Формирование и выполнение SELECT-запроса.

Объявление:

- `function Select($table, $conditions = Array(), $one_row = false, $fields = '*')`

Параметры:

- **string \$table**. Имя таблицы, в которой необходимо сделать выборку.
- **array \$conditions**. Массив условий, которые формируют часть WHERE запроса. Данный массив состоит из ключей, соответствующих полям массива, и их значений. Если данный массив пуст, часть WHERE в запросе будет отсутствовать.
- **bool \$one\_row**. Если значение данного параметра равно **true**, будет сформирован запрос, который извлечет из всего набора только первую запись (использование LIMIT 1).
- **string/array \$fields**. Массив полей таблицы, значения которых необходимо извлечь, либо строка с перечислением необходимых полей через запятую.

Возвращаемое значение:

- Тип **array**. Если параметр **\$one\_row** имеет значение **true**, возвращаемым значением является массив, ключи которого совпадают с извлекаемыми полями таблицы. Если параметр **\$one\_row** имеет значение **false**, возвращаемым значением является массив, каждым из

элементов которого является массив с ключами, совпадающими с извлекаемыми полями таблицы.

## **DB::TestConnect**

Назначение:

- Тестирование соединения с БД.

Объявление:

- `function TestConnect($dbname, $dbhost, $dbuser, $dbpwd)`

Параметры:

- **string \$dbname**. Имя базы данных.
- **string \$dbhost**. Адрес сервера баз данных.
- **string \$dbuser**. Имя пользователя.
- **string \$dbpwd**. Пароль пользователя.

Возвращаемое значение:

- Тип **bool**. True, если удалось подключиться к БД. Иначе – false.

## **\$DB->Update**

Назначение:

- Формирование и выполнение UPDATE-запроса.

Объявление:

- `function Update($table, $conditions, $updates)`

Параметры:

- **string \$table**. Имя таблицы, для которой необходимо выполнить UPDATE-запрос.
- **array \$conditions**. Массив условий (в части WHERE SQL-запроса), который состоит из ключей (соответствующих полям таблицы) и их значений. См. пример.
- **array \$updates**. Массив обновляемых полей таблицы, который состоит из ключей (соответствующих полям таблицы, значения которых нужно установить) и их значений. См. пример.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

Пример:

- `$DB->Update($TABLES['articles'], array('id' => $article_id), array('url_name' => $url_name));`

## **Класс DBBackup**

Предназначен для управления резервными копиями магазина. Применяется модулем **db\_backup**.

### **DBBackup::AppendDump**

Назначение:



- Запись (добавление) текстовых данных (SQL-запросов создания таблиц и формирования их записей) в формируемый SQL-файл.

Объявление:

- `function AppendDump($file, $dump)`

Параметры:

- **string \$file**. Путь к файлу, в который необходимо дописать текстовую информацию.
- **string \$dump**. Текстовая информация, которую необходимо дописать в указанный файл.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **DBBackup::GetDump**

Назначение:

- Формирование SQL-запросов для создания указанной таблицы и ее записей.

Объявление:

- `function GetDump($table)`

Параметры:

- **string \$table**. Название таблицы, дампа которой необходимо получить.

Возвращаемое значение:

- Тип **string/bool**. Дамп таблицы в виде текста или false – в случае ошибки.

## **DBBackup::GetDumpsList**

Назначение:

- Извлечение списка ZIP-архивов резервных копий, созданных ранее.

Объявление:

- `function GetDumpsList($root)`

Параметры:

- **string \$root**. Путь к директории, в которой хранятся файлы резервных копий. Обычно в качестве значения данного параметра используется константа [BACKUP\\_PATH](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом со следующими ключами:
  - **filename** – имя файла резервной копии;
  - **filesize** – размер файла (с приставкой bytes, Kb или Mb – в зависимости от размера файла).

## **DBBackup::GetUploadedFile**

Назначение:

- Определяет временный путь к загруженному пользователем файлу резервной копии (для ее восстановления).

Объявление:

- function GetUploadedFile(\$sql\_file)

Параметры:

- **file \$sql\_file**. Описание загружаемого методом POST файла из **\$\_FILES**.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – путь к временному файлу резервной копии (содержится во временной директории, путь к которой хранится в константе [TMP\\_PATH](#)).

## Класс DBClear

Предназначен для очистки БД (удаления всей информации из некоторых таблиц).

Используется модулем **db\_clear**.

### **DBClear::Access**

Назначение:

- Очистка таблиц, относящихся к черному списку IP-адресов ([proshop\\_access\\_ip](#), [proshop\\_access\\_ip\\_texts](#)).

Объявление:

- function Access()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

### **DBClear::Articles**

Назначение:

- Очистка таблиц, относящихся к HTML-страницам и статьям ([proshop\\_articles](#), [proshop\\_articles\\_texts](#)).

Объявление:

- function Articles()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

### **DBClear::Categories**

Назначение:

- Очистка таблиц, относящихся к категориям и продуктам.

Объявление:

- function Categories()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBClear::Discounts**

Назначение:

- Очистка таблиц, связанных со скидками.

Объявление:

- function Discounts()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBClear::LoginHistory**

Назначение:

- Очистка журнала заходов администраторов (таблицы [proshop\\_login\\_history](#)).

Объявление:

- function LoginHistory()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBClear::Manufacturers**

Назначение:

- Очистка таблиц, связанных с производителями ([proshop\\_manufacturers](#), [proshop\\_manufacturers\\_texts](#)).

Объявление:

- function Manufacturers()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBClear::Messages**

Назначение:

- Очистка таблиц, связанных с хранением личных сообщений и тем (переписка администрации и покупателей) – [proshop\\_msgs](#), [proshop\\_msgs\\_topics](#).

Объявление:

- function Messages()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBCclear::MetaTags**

Назначение:

- Очистка таблиц, связанных с хранением мета-тэгов ([proshop\\_meta\\_tags](#), [proshop\\_meta\\_texts](#)).

Объявление:

- function MetaTags()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBCclear::News**

Назначение:

- Очистка таблиц, связанных с новостями (proshop\_news, proshop\_news\_texts).

Объявление:

- function News()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBCclear::NewsSubscribers**

Назначение:

- Удаление из базы всех подписчиков на новости (хранятся в таблице [proshop\\_news\\_subscribers](#)).

Объявление:

- function NewsSubscribers()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DVClear::Orders**

Назначение:

- Удаление заказов из базы.

Объявление:

- function Orders()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DVClear::Products**

Назначение:

- Удаление информации о продуктах из БД.

Объявление:

- function Products()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DVClear::Stats**

Назначение:

- Очистка статистики просмотра категорий и поиска продуктов.

Объявление:

- function Stats()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DVClear::Taxes**

Назначение:

- Очистка таблиц, связанных с налогами ([proshop\\_taxes](#), [proshop\\_taxes\\_texts](#)).

Объявление:

- function Taxes()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **DBClear::Votes**

Назначение:

- Удаление всех голосований (очистка таблиц [proshop\\_votes](#), [proshop\\_votes\\_texts](#), ...).

Объявление:

- function Votes()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс Discounts**

Предназначен для управления скидками. Используется модулем **discounts**.

### **Discounts::AddRoot**

Назначение:

- Добавление базовой информации о скидке в БД (в таблицу [proshop\\_discounts](#)).

Объявление:

- function AddRoot(\$type, \$selem\_type, \$value, \$active)

Параметры:

- **string \$type**. Тип скидки – абсолютная (**abs**) или в процентах (**per**).
- **int \$selem\_type**. Идентификатор категории скидки (из таблицы [proshop\\_discounts\\_types](#)) – глобальная, для продуктов или для пользователей.
- **string \$value**. Значение скидки в базовой валюте магазина или в процентах (в зависимости от значения параметра **\$type**).
- **char \$active**. Если значение данного параметра равно **Y**, скидка активна. Иначе – скидка временно не действует.

Возвращаемое значение:

- Тип **int/bool**. **False** – в случае ошибки. Иначе – идентификатор созданной скидки.

### **Discounts::AddText**

Назначение:

- Добавление в базу названия указанной скидки для указанного языка.

Объявление:

- function AddText(\$id, \$lang\_id, \$caption)

Параметры:

- **int \$id**. Идентификатор скидки, созданной с помощью [Discounts::AddRoot](#).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо добавить название скидки.
- **string \$caption**. Название скидки. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Discounts::CartAdd**

Назначение:

- Добавление условия скидки (разовой или накопительной) на корзину товаров покупателя в таблицу [proshop\\_discounts\\_cart](#).

Объявление:

- function CartAdd(\$low\_band, \$discount, \$per, \$accumulative)

Параметры:

- **float \$low\_band**. Нижняя граница диапазона суммы заказа в базовой валюте, начиная с которой будет действовать скидка.
- **float \$discount**. Абсолютное значение скидки в базовой валюте.
- **float \$per**. Относительное значение скидки – в процентах от суммы заказа.
- **bool \$accumulative**. Если значение данного параметра равно **true**, скидка является накопительной. Иначе – скидка разовая.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Discounts::CartDelete**

Назначение:

- Удаление указанного условия скидки (разовой или накопительной) на корзину товаров покупателя из таблицы [proshop\\_discounts\\_cart](#).

Объявление:

- function CartDelete(\$id)

Параметры:

- **int \$id**. Идентификатор условия скидки из таблицы [proshop\\_discounts\\_cart](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Discounts::CartGetAll**

Назначение:

- Извлечение всех условий скидок (накопительных или разовых) на корзину товаров покупателя, заданных в таблице [proshop\\_discounts\\_cart](#), в порядке, определенном полем **low\_band** (по возрастанию значения данного поля).

Объявление:

- function CartGetAll(\$accumulative)

Параметры:

- **bool \$accumulative**. Если значение данного параметра равно **true**, функция вернет список условий накопительных скидок. Иначе – список условий разовых скидок.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_discounts\\_cart](#).

## **Discounts::CartGetAmount**

Назначение:

- Определение скидки (разовой или накопительной) на корзину товаров покупателя для указанной суммы.

Объявление:

- function CartGetAmount(\$sum, \$accumulative)

Параметры:

- **float \$sum**. Значение сумм в базовой валюте (сумма заказа, если **\$accumulative** равно **false**, или сумма всех завершённых заказов, если **\$accumulative** равно **true**), для которой нужно определить размер скидки (в базовой валюте).
- **bool \$accumulative**. Если данный параметр равен **true**, то считается, что **\$sum** – сумма всех завершённых заказов покупателя. Иначе – считается, что **\$sum** – сумма одного заказа.

Возвращаемое значение:

- Тип **float**. Значение скидки в базовой валюте.

## **Discounts::CartGetCount**

Назначение:

- Определение кол-ва условий скидок (накопительных или разовых) на корзину товаров покупателя, заданных в таблице [proshop\\_discounts\\_cart](#).

Объявление:

- function CartGetCount(\$accumulative)

Параметры:

- **bool \$accumulative**. Если значение данного параметра равно **true**, функция вернет кол-во условий накопительных скидок. Иначе – кол-во условий разовых скидок.

Возвращаемое значение:

- Тип **int**. Кол-во условий скидок.

## **Discounts::Delete**

Назначение:

- Удаление указанной скидки из БД.

Объявление:



- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор удаляемой скидки (из таблицы [proshop\\_discounts](#)).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Discounts::ETypeToId**

Назначение:

- Определение идентификатора категории скидки по названию этой категории.

Объявление:

- function ETypeToId(\$elem\_type)

Параметры:

- **string \$elem\_type**. Служебное название категории скидки (из таблицы [proshop\\_discounts\\_types](#)).

Возвращаемое значение:

- Тип **int**. Идентификатор категории скидки.

## **Discounts::GetActiveConf**

Назначение:

- Определение способа учета пересекающихся скидок – суммировать все скидки или выбрать наибольшую. Способ учета задается через админку и хранится в таблице [proshop\\_discounts\\_conf](#).

Объявление:

- function GetActiveConf()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – способ учета пересекающихся скидок: **sum\_of\_all** или **max\_of\_all**.

## **Discounts::GetActiveCount**

Назначение:

- Определение количества активных скидок (не отключенных администратором), хранящихся в таблице [proshop\\_discounts](#) и имеющих значение поля **active**, равное Y.

Объявление:

- function GetActiveCount()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во активных скидок.

## **Discounts::GetAll**

Назначение:

- Извлечение всех скидок заданной категории (из таблицы [proshop\\_discounts](#)).

Объявление:

- `function GetAll($type_id, $lang_id)`

Параметры:

- **int \$type\_id**. Идентификатор категории скидки. Категории скидок хранятся в таблице [proshop\\_discounts\\_types](#).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь названия скидок.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, каждый элемент которого также является массивом, содержащим следующие ключи:
  - **id** – идентификатор скидки;
  - **caption** – название скидки на указанном языке;
  - **type** – тип скидки (абсолютная – в базовой валюте магазина (**abs**), или процентная - в процентах (**per**));
  - **elem\_type** – категория скидки (из таблицы [proshop\\_discounts\\_types](#));
  - **value** – значение скидки;
  - **active** – активна скидка (**Y**), или нет (**N**).

## **Discounts::GetConfList**

Назначение:

- Извлечение списка способов учета пересекающихся скидок (из таблицы [proshop\\_discounts\\_conf](#)).

Объявление:

- `function GetConfList()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, каждым элементом которого является массив с ключом **name** (из таблицы [proshop\\_discounts\\_conf](#)) – именем способа учета пересекающихся скидок.

## **Discounts::GetGlobals**

Назначение:

- Извлечение активных глобальных скидок из таблицы [proshop\\_discounts](#).

Объявление:

- function GetGlobals()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_discounts](#). В данном массиве содержатся только скидки, категория которых – **глобальные**, и которые активны (поле **active** таблицы [proshop\\_discounts](#) имеет значение Y).

## Discounts::GetOne

Назначение:

- Извлечение информации об указанной скидке из таблицы [proshop\\_discounts](#).

Объявление:

- function GetOne(\$id)

Параметры:

- **int \$id**. Идентификатор скидки из таблицы [proshop\\_discounts](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив с ключами, совпадающими с полями таблицы [proshop\\_discounts](#).

## Discounts::GetTypes

Назначение:

- Извлечение категорий скидок из таблицы [proshop\\_discounts\\_types](#).

Объявление:

- function GetTypes()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими полями таблицы [proshop\\_discounts\\_types](#).

## Discounts::IsTextExists

Назначение:

- Определение, занято ли уже заданное название скидки для указанного языка.

Объявление:

- function IsTextExists(\$caption, \$lang\_id, \$type\_id, \$id = 0)

Параметры:

- **string \$caption**. Название скидки на заданном языке.
- **int \$lang\_id**. Идентификатор языка.

- **int \$type\_id**. Идентификатор категории скидки (из таблицы [proshop\\_discounts\\_types](#)) – в пределах этой категории будет осуществляться поиск названия **\$caption**.
- **int \$id**. Идентификатор скидки, которую не следует учитывать при поиске названия. Если значение данного параметра равно 0, учитываются все скидки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Discounts::SetActiveConf**

Назначение:

- Установка активного способа учета пересекающихся скидок (из таблицы [proshop\\_discounts\\_conf](#)).

Объявление:

- function SetActiveConf(\$id)

Параметры:

- **int \$id**. Идентификатор способа учета пересекающихся скидок из таблицы [proshop\\_discounts\\_conf](#), который необходимо сделать активным по умолчанию.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Discounts::Update**

Назначение:

- Обновление информации о скидке в БД.

Объявление:

- function Update(\$lang\_id, \$info)

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо обновить название скидки из параметра **\$info**.
- **array \$info**. Информация о скидке. Ключи данного массива должны быть следующими:
  - **id** – идентификатор скидки, которую необходимо обновить;
  - **type** – тип скидки: абсолютная (**abs**) или процентная (**per**);
  - **value** – значение скидки;
  - **caption** – название скидки на заданном языке **\$lang\_id**;
  - **active** – активна ли скидка (**Y** – активна, **N** – временно выключена).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс eDost**

Предназначен для взаимодействия с сервисом онлайн-расчета стоимости доставки eDost (<http://edost.ru>).

## **\$edost->Calc**

Назначение:

- Взаимодействие с сервером eDost и определение стоимости доставки товара по его весу, габаритам и сумме страховки в заданный город, страну или область. Функция возвращает ассоциативный массив с информацией о возможных способах доставки, их стоимости и срока доставки.

Объявление:

- `public function Calc($edost_id, $edost_pwd, $to_city, $weight, $strah, $lenght, $width, $height)`

Параметры:

- **int \$edost\_id**. Идентификатор магазина.
- **string \$edost\_pwd**. Пароль для доступа к серверу расчетов.
- **string \$to\_city**. Название города, страны или области в кодировке Windows-1251 или код соответствующего географического объекта в базе данных eDost (коды можно найти на сайте <http://edost.ru>).
- **float \$weight**. Вес посылки в кг.
- **float \$strah**. Сумма страховки в рублях. Может быть равна 0.
- **float \$lenght**. Длина коробки в единицах измерения, указанных в настройках магазина на сайте edost.ru. Задавать необязательно. Служит для более точных расчетов стоимости доставки.
- **float \$width**. Ширина коробки в единицах измерения, указанных в настройках магазина на сайте edost.ru. Задавать необязательно. Служит для более точных расчетов стоимости доставки.
- **float \$height**. Высота коробки в единицах измерения, указанных в настройках магазина на сайте edost.ru. Задавать необязательно. Служит для более точных расчетов стоимости доставки.

Возвращаемое значение:

- Тип **array**. Формат возвращаемого массива:
  - result['stat'] - код результата запроса
  - result['qty\_companu'] - количество тарифов
  - Записи по каждому тарифу (N):
    - result['id'.N] - код тарифа
    - result['price'.N] - цена доставки в рублях
    - result['day'.N] - количество дней доставки
    - result['starh'.N] - код страхования (1 - рассчитано со страховкой, 0 - без страховки)
    - result['company'.N] - название компании
    - result['name'.N] - название тарифа

## **\$edost->SetSiteUTF**

Назначение:

- Установка флага кодировки UTF-8. Если данная функция будет вызвана, функция [Calc](#) вернет информацию в кодировке UTF-8.

Объявление:

- `public function SetSiteUTF()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$edost->SetSiteWin**

Назначение:

- Установка флага кодировки Windows-1251. Если данная функция будет вызвана, функция [Calc](#) вернет информацию в кодировке Windows-1251.

Объявление:

- `public function SetSiteWin()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс EMSPost**

Предназначен для работы с сервисом онлайн-расчета стоимости доставки методом EMS.

### **EMSPost::Calculate**

Назначение:

- Подсчет стоимости и сроков доставки методом EMS по указанным параметрам: место отправления, место назначения, вес посылки. Данная функция работает с удаленным сервисом (при проблемах с соединением делается три попытки подключения к сервису).

Объявление:

- `function Calculate($from, $to, $weight)`

Параметры:

- **string \$from**. Служебное имя места отправления. Определяется в специальном формате (например, для москвы этот параметр должен быть равен `city--moskva`), который можно посмотреть, выполнив функцию [EMSPost::GetLocations](#) (возвращает перечень городов и их служебных названий).
- **string \$to**. Служебное имя места назначения.
- **float \$weight**. Вес посылки в килограммах.

Возвращаемое значение:

- Тип **array**. Массив, первый элемент которого определяет стоимость доставки (если не удалось определить, возвращается -1). Вторым элементом является ассоциативный массив, содержащий ключи **delivery\_time1** и **delivery\_time2**, значения которых определяют соответственно минимальный и максимальный срок доставки в днях.

## **EMSPost::GetLocations**

Назначение:

- Определяет и возвращает перечень поддерживаемых сервисом EMS географических объектов (городов России). Данная функция работает с удаленным сервисом (при проблемах с соединением делается три попытки подключения к сервису).

Объявление:

- `function GetLocations()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/string**. В случае ошибки возвращается пустая строка. Иначе – массив, соответствующий JSON-выводу запроса на сервер EMS. Более подробно – см. вывод этой функции с помощью `print_r`.

## **EMSPost::GetMaxWeight**

Назначение:

- Определяет максимально допустимый вес посылки в килограммах. Данная функция работает с удаленным сервисом (при проблемах с соединением делается три попытки подключения к сервису).

Объявление:

- `function GetMaxWeight()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **float**. Максимально допустимый вес посылки в килограммах. В случае ошибки возвращается 0.

## **EMSPost::GetOption**

Назначение:

- Извлечение значения указанной опции сервиса доставки EMS из таблицы [proshop\\_emspost\\_options](#).

Объявление:

- `function GetOption($name)`

Параметры:

- **string \$name**. Название опции.

Возвращаемое значение:

- Тип **string/bool**. В случае ошибки возвращается **false**. Иначе – значение заданной опции.

## **EMSPost::SetOption**

Назначение:

- Установка значения указанной опции сервиса доставки EMS в таблице [proshop\\_emspost\\_options](#).

Объявление:

- `function SetOption($name, $value)`

Параметры:

- **string \$name**. Название опции.
- **string \$value**. Значение опции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс ErrorLogger**

Предназначен для хранения сообщений об ошибках, возникающих в работе CMS.

### **ErrorLogger::ArrayToXML**

Назначение:

- Преобразование массива с информацией об ошибке в XML-представление.

Объявление:

- `function ArrayToXML($in_errors)`

Параметры:

- **array \$in\_errors**. Массив ошибок, каждый элемент которого является массивом со следующими ключами:
  - **add\_date** – дата возникновения ошибки;
  - **errno** – код ошибки;
  - **errmsg** – текст описания ошибки;
  - **file** – путь к файлу, в котором произошла ошибка;
  - **line** – номер строки в файле, на которой произошла ошибка;
  - **uri** – адрес страницы, на которой произошла ошибка.

Возвращаемое значение:

- Тип **string**. XML-представление массива ошибок.

### **ErrorLogger::Clear**

Назначение:



- Очистка лога ошибок (удаление файла лога, путь к которому прописан в константе [ERRORS\\_LOG\\_FILE](#), и очистка таблицы [proshop\\_errors](#)).

Объявление:

- `function Clear()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ErrorLogger::DeleteOldNErrorsFromDB**

Назначение:

- Удаление заданного количества самых первых ошибок из БД.

Объявление:

- `function DeleteOldNErrorsFromDB($count)`

Параметры:

- **int \$count**. Количество первых ошибок в БД, которые необходимо удалить.

Возвращаемое значение:

- Функция ничего не возвращает.

## **ErrorLogger::DeleteOldNErrorsFromFile**

Назначение:

- Удаление заданного количества самых первых ошибок из файла лога, путь к которому хранится в константе [ERRORS\\_LOG\\_FILE](#).

Объявление:

- `function DeleteOldNErrorsFromFile($in_count)`

Параметры:

- **int \$count**. Количество первых ошибок в файле лога, которые необходимо удалить.

Возвращаемое значение:

- Функция ничего не возвращает.

## **ErrorLogger::GetAllXML**

Назначение:

- Извлечение всех ошибок из лога в виде XML.

Объявление:

- `function GetAllXML($storage = 'db')`

Параметры:

- **string \$storage**. Название хранилища, из которого требуется извлечь ошибки. Если значение данного параметра равно **db**, лог будет извлечен из БД. Иначе – из файла, путь к которому прописан в константе [ERRORS\\_LOG\\_FILE](#).

Возвращаемое значение:

- Тип **string**. Лог ошибок в виде XML.

## **ErrorLogger::GetAllXMLFromDB**

Назначение:

- Извлечение лога ошибок из БД в виде XML.

Объявление:

- function GetAllXMLFromDB()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Лог ошибок в виде XML.

## **ErrorLogger::GetAllXMLFromFile**

Назначение:

- Извлечение лога ошибок из файла лога (путь к файлу лога хранится в константе [ERRORS\\_LOG\\_FILE](#)) в виде XML.

Объявление:

- function GetAllXMLFromFile()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Лог ошибок в виде XML.

## **ErrorLogger::GetDBLogLength**

Назначение:

- Определение кол-ва записей об ошибках в БД.

Объявление:

- function GetDBLogLength()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во записей в логе ошибок (в таблице [proshop\\_errors](#)).

## **ErrorLogger::GetFileLogLength**

Назначение:

- Определение кол-ва записей об ошибках в файле лога, путь к которому хранится в константе [ERRORS\\_LOG\\_FILE](#).

Объявление:

- function GetFileLogLength()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во записей в лог-файле ошибок.

## **ErrorLogger::LogError**

Назначение:

- Фиксирование ошибки в логе. Если значение константы [ENABLE\\_ERROR\\_LOG\\_ROTATION](#) равно true, из лога будут удалены старые записи – останутся только последние [ERRORS\\_LOG\\_LIMIT](#) записей.

Объявление:

- function LogError(\$data, \$log\_type = 'db')

Параметры:

- **array \$data**. Массив с информацией об ошибке. Должен содержать следующие ключи:
  - **errno** – код ошибки;
  - **errmsg** – текст описания ошибки;
  - **filename** – путь к файлу, в котором произошла ошибка;
  - **linenum** – номер строки в файле, на которой произошла ошибка;
  - **uri** – адрес страницы, на которой произошла ошибка.
- **string \$log\_type**. Название хранилища, в которое будет записана ошибка. Если значение данного параметра равно **db**, лог будет записан в БД. Иначе (если значение равно **file**) – в файл, путь к которому прописан в константе [ERRORS\\_LOG\\_FILE](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## **ErrorLogger::SaveErrorToDB**

Назначение:

- Фиксирование ошибки в таблице [proshop\\_errors](#). Если значение константы [ENABLE\\_ERROR\\_LOG\\_ROTATION](#) равно true, из лога будут удалены старые записи – останутся только последние [ERRORS\\_LOG\\_LIMIT](#) записей.

Объявление:

- function SaveErrorToDB(\$data)

Параметры:

- **array \$data**. Массив с информацией об ошибке. Должен содержать следующие ключи:
  - **errno** – код ошибки;
  - **errmsg** – текст описания ошибки;
  - **filename** – путь к файлу, в котором произошла ошибка;
  - **linenum** – номер строки в файле, на которой произошла ошибка;

- **uri** – адрес страницы, на которой произошла ошибка.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ErrorLogger::SaveErrorToFile**

Назначение:

- Фиксирование ошибки в лог-файле, путь к которому прописан в константе [ERRORS\\_LOG\\_FILE](#). Если значение константы [ENABLE\\_ERROR\\_LOG\\_ROTATION](#) равно true, из лога будут удалены старые записи – останутся только последние [ERRORS\\_LOG\\_LIMIT](#) записей.

Объявление:

- function SaveErrorToFile(\$data)

Параметры:

- **array \$data**. Массив с информацией об ошибке. Должен содержать следующие ключи:
  - **errno** – код ошибки;
  - **errmsg** – текст описания ошибки;
  - **filename** – путь к файлу, в котором произошла ошибка;
  - **linenum** – номер строки в файле, на которой произошла ошибка;
  - **uri** – адрес страницы, на которой произошла ошибка.

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Класс Excel**

Предназначен для работы с CSV-файлом, содержащим список продуктов (или любых других элементов) с информацией о них. Используется модулями **import** и **export**.

### **\$csv->AddHeader**

Назначение:

- Добавление информации о заголовке CSV-файла для последующего формирования первой строки CSV-файла со столбцами.

Объявление:

- function AddHeader(\$header)

Параметры:

- **array \$header**. Массив с информацией о столбцах CSV-файла. См. модуль экспорта для примера.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$csv->AddItem**

Назначение:

- Добавление информации об элементе в массив для подготовки ее вывода в CSV-файл.

Объявление:

- `function AddItem($info)`

Параметры:

- **array \$info**. См. описание параметра **\$header** функции [\\$csv->AddHeader](#). Ключи массива **\$info** должны совпадать с ключами массива **\$header**. Отличие параметра **\$info** от **\$header** в том, что вместо названий полей должны содержаться соответствующие значения этих полей (например, вместо **[PRODUCT\_ID]** – целое число – идентификатор продукта).

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$csv->Escape**

Назначение:

- Подготовка текстовой строки для ее записи в CSV-файл (преобразование кавычек, символов перевода строки и т.п.).

Объявление:

- `function Escape($str)`

Параметры:

- **string \$str**. Строка, которую необходимо обработать перед записью в CSV-файл (перед помещением ее значения в соответствующий ключ массива **\$info** и вызовом [\\$csv->AddItem](#)).

Возвращаемое значение:

- Тип **string**. Фильтрованная строка.

## **\$csv->GetCSV**

Назначение:

- Генерирование CSV-текста на основе информации, полученной путем вызовов [\\$csv->AddHeader](#) и [\\$csv->AddItem](#), для его записи в CSV-файл.

Объявление:

- `function GetCSV($sep, $no_header = false)`

Параметры:

- **char \$sep**. Символ разделителя ячеек (обычно точка с запятой).
- **bool \$no\_header**. Если значение данного параметра равно true, заголовок не включается в CSV-текст.

Возвращаемое значение:

- Тип **string**. CSV-текст, готовый для записи в CSV-файл.

## **`$csv->GetCSVFileName`**

Назначение:

- Генерирование имени CSV-файла на основе текущего времени и заданной кодировки файла.

Объявление:

- `function GetCSVFileName($type, $encoding)`

Параметры:

- **string \$type**. Тип файла – может быть любой строкой, которая войдет в название файла.
- **string \$encoding**. Название кодировки, в которой будет сохранен CSV-файл, имя которого требуется сгенерировать.

Возвращаемое значение:

- Тип **string**. Сгенерированное имя CSV-файла.

## **`$csv->LoadHeader`**

Назначение:

- Загрузка массива полей из CSV-файла (первая строчка CSV-файла в виде массива).

Объявление:

- `function LoadHeader($file, $sep = ';', $delim = "")`

Параметры:

- **string \$file**. Имя CSV-файла (или путь к нему), который необходимо прочитать.
- **char \$sep**. Символ-разделитель ячеек файла.
- **char \$delim**. Символ-ограничитель текстовых строк.

Возвращаемое значение:

- Тип **array**. Массив из названий полей CSV-файла, разделенных указанным разделителем. См. возвращаемое значение функции [\\$csv->Load](#) для справки о возможных полях.

## **`$csv->Load`**

Назначение:

- Извлечение информации об элементах из CSV-файла – конвертирование CSV-файла в массив элементов.

Объявление:

- `function Load($file, $encoding, $sep = ';', $delim = "")`

Параметры:

- **string \$file**. Имя CSV-файла (или путь к нему), который необходимо прочитать.
- **string \$encoding**. Название кодировки файла.
- **char \$sep**. Символ-разделитель ячеек файла.
- **char \$delim**. Символ-ограничитель текстовых строк.

Возвращаемое значение:

- Тип **array**. Массив элементов. См. модуль импорта для примера.

## **\$csv->Save**

Назначение:

- Сохранение текста, сформированного с помощью [\\$csv->GetCSV](#), в CSV-файл.

Объявление:

- `function Save($csv, $file, $append = false)`

Параметры:

- **string \$csv**. Текст, сформированный ранее с помощью [\\$csv->GetCSV](#).
- **string \$file**. Имя CSV-файла (или путь к нему), в который необходимо сохранить текст.
- **bool \$append**. Если данный параметр равен true, уже существующий файл не будет удален, а текст \$csv будет дописан в конец этого файла.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **\$csv->UnEscape**

Назначение:

- Обрезание символов пробела в начале и в конце строки – после чтения строки из CSV-файла.

Объявление:

- `function UnEscape($str)`

Параметры:

- **string \$str**. Строка, которую необходимо обработать.

Возвращаемое значение:

- Тип **string**. Обработанная строка.

## **Класс FastMenu**

Предназначен для управления быстрым меню на витрине. Используется модулем **fast\_menu**.

### **FastMenu::AddRoot**

Назначение:

- Добавление базовой информации о ссылке в быстром меню (в таблицу [proshop\\_fastmenu](#)).

Объявление:

- `function AddRoot($is_visible)`

Параметры:

- **bool \$is\_visible**. Флаг видимости добавляемой ссылки на витрине.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор добавленной ссылки.

## **FastMenu::AddTexts**

Назначение:

- Добавление текстовой информации (зависимой от языка) о ссылке (адрес и название ссылки).

Объявление:

- `function AddTexts($menu_id, $lang_id, $title, $url)`

Параметры:

- **int \$menu\_id**. Идентификатор ссылки, для которой необходимо добавить текстовую информацию.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо добавить адрес ссылки и ее название.
- **string \$title**. Название ссылки на указанном языке. Макс. 254 символа.
- **string \$url**. Адрес ссылки на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **FastMenu::Delete**

Назначение:

- Удаление указанной ссылки из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор ссылки из таблицы [proshop\\_fastmenu](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **FastMenu::GetAll**

Назначение:

- Извлечение всех ссылок быстрого меню из БД.

Объявление:

- `function GetAll($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь адрес и название каждой из ссылок.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом со следующими ключами:
  - **id** – идентификатор ссылки (из таблицы [proshop\\_fastmenu](#));
  - **pos** – позиция ссылки относительно других ссылок быстрого меню;



- **title** – название ссылки на указанном языке;
- **url** – адрес ссылки на указанном языке;
- **is\_visible** – видима ли ссылка на витрине (**Y** – видима, **N** - невидима).

## **FastMenu::GetAllVisible**

Назначение:

- Извлечение всех ссылок быстрого меню с флагом видимости **is\_visible** (поле таблицы [proshop\\_fastmenu](#)), равным **Y**, из БД.

Объявление:

- function GetAllVisible(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь адрес и название каждой из ссылок.

Возвращаемое значение:

- Тип **array/bool**. См. возвращаемое значение [FastMenu::GetAll](#).

## **FastMenu::SetPos**

Назначение:

- Установка позиции указанной ссылки относительно других ссылок быстрого меню.

Объявление:

- function SetPos(\$item\_id, \$pos)

Параметры:

- **int \$item\_id**. Идентификатор ссылки из таблицы [proshop\\_fastmenu](#).
- **int \$pos**. Позиция ссылки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **FastMenu::Update**

Назначение:

- Обновление информации об указанной ссылке быстрого меню в БД.

Объявление:

- function Update(\$id, \$lang\_id, \$info)

Параметры:

- **int \$id**. Идентификатор ссылки из таблицы [proshop\\_fastmenu](#).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо установить адрес и название ссылки.
- **array \$info**. Массив с информацией о ссылке, ключи которого должны быть следующими:
  - **title** – название ссылки на указанном языке;
  - **url** – адрес ссылки на указанном языке;

- **is\_visible** – видима ли ссылка на витрине (**Y** – видима, **N** - невидима).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс **FeaturedProducts**

Предназначен для управления специальными предложениями – товарами, которые отображаются на главной странице. Используется модулем **featured\_products**.

### **FeaturedProducts::Delete**

Назначение:

- Удаление специального предложения из таблицы [proshop\\_products\\_featured](#).

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор специального предложения из таблицы [proshop\\_products\\_featured](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

### **FeaturedProducts::GetAll**

Назначение:

- Извлечение списка специальных предложений из таблицы [proshop\\_products\\_featured](#).

Объявление:

- function GetAll()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом, содержащим ключи, совпадающие с полями таблицы [proshop\\_products\\_featured](#).

### **FeaturedProducts::GetIds**

Назначение:

- Извлечение списка идентификаторов продуктов, являющихся спецпредложениями, из таблицы [proshop\\_products\\_featured](#).

Объявление:

- function GetIds()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив идентификаторов продуктов-спецпредложений.

## **FeaturedProducts :: IsFeatured**

Назначение:

- Определение, является ли указанный продукт специальным предложением.

Объявление:

- `function IsFeatured($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **FeaturedProducts :: Remove**

Назначение:

- Удаление продукта из списка специальных предложений (из таблицы [proshop\\_products\\_featured](#)).

Объявление:

- `function Remove($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#), который необходимо удалить из списка спецпредложений.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **FeaturedProducts :: Set**

Назначение:

- Добавление указанного продукта в список спецпредложений (в таблицу [proshop\\_products\\_featured](#)).

Объявление:

- `function Set($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **FeaturedProducts :: SetPos**

Назначение:

- Установка позиции спецпредложения относительно других спецпредложений в таблице [proshop\\_products\\_featured](#).

Объявление:

- `function SetPos($fprod_id, $pos)`

Параметры:

- **int \$fprod\_id**. Идентификатор спецпредложения в таблице [proshop\\_products\\_featured](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс FedEx

Предназначен для взаимодействия с онлайн-сервисом по расчету стоимости доставки компанией FedEx. Более подробную информацию о форматах передачи данных на сервис FedEx смотрите на сайте [fedex.com](#).

### **\$fedex->getCurrency**

Назначение:

- Определение валюты, в которой задана стоимость доставки указанным сервисом. Стоимость доставки можно определить с помощью метода [\\$fedex->getPrice](#).

Объявление:

- `function getCurrency($service)`

Параметры:

- **string \$service**. Название указанного сервиса из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **string**. Международное трехбуквенное обозначение валюты.

### **\$fedex->getDates**

Назначение:

- Определение предполагаемой даты доставки заказа для указанного сервиса.

Объявление:

- `function getDates($service)`

Параметры:

- **string \$service**. Название указанного сервиса из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **string**. Дата предполагаемой доставки заказа.

### **\$fedex->getPrice**

Назначение:

- Извлечение стоимости доставки заказа указанным сервисом в валюте, которая определена с помощью метода [\\$fedex->getCurrency](#).

Объявление:

- `function getPrice($service)`

Параметры:

- **string \$service**. Название указанного сервиса из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **float**. Стоимость доставки заказа.

## **\$fedex->sendRequest**

Назначение:

- Отправка XML-запроса на сервер FedEx, сформированного вызовом других методов класса, и запись XML-ответа во внутреннюю переменную **reqResult**.

Объявление:

- `function sendRequest()`

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$fedex->setAccountNumber**

Назначение:

- Установка номера аккаунта пользователя на сайте FedEx, который используется при расчете стоимости доставки.

Объявление:

- `function setAccountNumber($accountNumber)`

Параметры:

- **string \$accountNumber**. Номер аккаунта.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$fedex->setCarrierCode**

Назначение:

- Установка компании, ответственной за транспортировку заказа (FedEx Express или FedEx Ground). Данный класс поддерживает работу только с FedEx Express.

Объявление:

- `setCarrierCode($carrierCode)`

Параметры:

- **string \$carrierCode**. Код компании. В данном классе может принимать только одно значение – **FDXE**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setDestCountryCode`**

Назначение:

- Установка страны назначения, куда следует доставить заказ.

Объявление:

- `function setDestCountryCode($code)`

Параметры:

- **string \$code**. Международное двухбуквенное обозначение страны назначения.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setDestPostalCode`**

Назначение:

- Установка почтового индекса места назначения, куда следует доставить заказ.

Объявление:

- `function setDestPostalCode($code)`

Параметры:

- **string \$code**. Строка индекса места назначения.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setDestStateOrProvinceCode`**

Назначение:

- Установка кода области/штата/провинции места назначения, куда следует доставить заказ.

Объявление:

- `function setDestStateOrProvinceCode($code)`

Параметры:

- **string \$code**. Код области/штата/провинции места назначения.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setDropoffType`**

Назначение:

- Установка метода доставки заказа до FedEx, чтобы отправить его с помощью данной службы.

Объявление:

- `function setDropoffType($dropoffType)`

Параметры:

- **string \$dropoffType**. Тип доставки. Для справки о возможных значениях см. документацию на сайте [fedex.com](http://fedex.com) (элемент XML-запроса под названием **DropoffType**).

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setHeight`**

Назначение:

- Установка высоты коробки в дюймах или в сантиметрах (в зависимости от установленного значения методом [\\$fedex->setUnits](#)), если тип упаковки установлен в **YOURPACKAGING** (с помощью метода [\\$fedex->setPackaging](#)).

Объявление:

- `function setHeight($height)`

Параметры:

- **int \$height**. Целочисленное значение высоты коробки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setLength`**

Назначение:

- Установка длины коробки в дюймах или в сантиметрах (в зависимости от установленного значения методом [\\$fedex->setUnits](#)), если тип упаковки установлен в **YOURPACKAGING** (с помощью метода [\\$fedex->setPackaging](#)).

Объявление:

- `function setLength($length)`

Параметры:

- **int \$length**. Целочисленное значение длины коробки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setMeterNumber`**

Назначение:

- Установка опции «MeterNumber» - уникальный номер оборудования, с которого будет отправлен запрос на сервер FedEx. Данный номер присваивается вместе с номером аккаунта при регистрации на FedEx.

Объявление:

- `function setMeterNumber($meterNumber)`

Параметры:

- **string \$meterNumber**. Значение номера оборудования.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setOriginCountryCode`**

Назначение:

- Установка страны места отправления заказа.

Объявление:

- `function setOriginCountryCode($code)`

Параметры:

- **string \$code**. Международный двубуквенный код страны.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setOriginPostalCode`**

Назначение:

- Установка почтового индекса места отправления заказа.

Объявление:

- `function setOriginPostalCode($code)`

Параметры:

- **string \$code**. Значение почтового индекса.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setOriginStateOrProvinceCode`**

Назначение:

- Установка кода штата/области/провинции места отправления заказа.

Объявление:

- `function setOriginStateOrProvinceCode($code)`

Параметры:

- **string \$code**. Код штата/области/провинции.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setPackaging`**

Назначение:

- Установка типа отправляемой упаковки.

Объявление:

- `function setPackaging($packaging)`

Параметры:

- **string \$packaging**. Тип упаковки. Для справки о возможных значениях см. документацию на сайте [fedex.com](http://fedex.com) (элемент XML-запроса под названием **Packaging**).

Возвращаемое значение:



- Функция ничего не возвращает.

## **`$fedex->setServer`**

Назначение:

- Установка адреса сервера FedEx (полный путь к скрипту), на который необходимо отправить запрос.

Объявление:

- `function setServer($server)`

Параметры:

- **string \$server**. Полный URL-адрес скрипта FedEx, которому необходимо отправить запрос.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setUnits`**

Назначение:

- Установка единиц измерения размеров коробки, которые задаются с помощью [\\$fedex->setWidth](#), [\\$fedex->setHeight](#), [\\$fedex->setLength](#).

Объявление:

- `function setUnits($units)`

Параметры:

- **string \$units**. Код единицы измерения: **IN** – дюймы, **CM** – сантиметры.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setWeight`**

Назначение:

- Установка веса заказа, который необходимо переслать с помощью FedEx. Вес задается в единицах, заданных с помощью [\\$fedex->setWeightUnits](#).

Объявление:

- `function setWeight($weight)`

Параметры:

- **float \$weight**. Значение веса.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$fedex->setWeightUnits`**

Назначение:

- Установка единицы веса заказа, значение которого устанавливается с помощью [\\$fedex->setWeight](#).

Объявление:

- function setWeightUnits(\$units)

Параметры:

- **string \$units**. Код единицы веса: **KG** – килограммы, **LB** – фунты.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$fedex->setWidth**

Назначение:

- Установка ширины коробки в дюймах или в сантиметрах (в зависимости от установленного значения методом [\\$fedex->setUnits](#)), если тип упаковки установлен в **YOURPACKAGING** (с помощью метода [\\$fedex->setPackaging](#)).

Объявление:

- function setWidth(\$width)

Параметры:

- **int \$width**. Целочисленное значение ширины коробки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс GBaseProds**

Предназначен для экспорта продуктов магазина в Google Base. Используется модулем экспорта.

### **\$gbase->AddProduct**

Назначение:

- Добавление информации о продукте во внутренний массив класса **GbaseProds** для его последующего экспорта в XML-файл.

Объявление:

- function AddProduct(\$info)

Параметры:

- **array \$info**. Массив с информацией о продукте. Должен содержать следующие ключи:
  - **id** – идентификатор продукта;
  - **title** – название продукта;
  - **brand** – название производителя;
  - **descr** – описание продукта;
  - **image** – URL изображения продукта;
  - **link** – ссылка на страницу с описанием продукта;
  - **price** – цена продукта в долларах;
  - **qty** – доступное кол-во продукта на складе;

- **category** – название категории продукта.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$gbase->Escape**

Назначение:

- Подготовка строки перед ее записью в тэг XML-файла.

Объявление:

- `function Escape($str, $len = 0)`

Параметры:

- **string \$str**. Строка, которую необходимо отфильтровать.
- **int \$len**. Максимальная длина строки **\$str**. Если длина строки **\$str** превышает данное значение, она будет обрезана, а в ее конец будет добавлено многоточие.

Возвращаемое значение:

- Тип **string**. Обработанная строка.

## **\$gbase->GetGBaseFileName**

Назначение:

- Возвращает имя XML-файла, который будет создан в результате экспорта продуктов.

Объявление:

- `function GetGBaseFileName()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Имя XML-файла (с расширением xml).

## **\$gbase->GetXML**

Назначение:

- Генерирование XML-контента для его сохранения в XML-файл.

Объявление:

- `function GetXML()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. XML-контент, готовый для записи в файл.

## **\$gbase->Save**

Назначение:

- Сохранение XML-содержимого, подготовленного с помощью [\\$gbase->GetXML](#), в XML-файл.

Объявление:

- `function Save($xml, $file)`

Параметры:

- `string $xml`. XML-контент, который требуется сохранить в файл.
- `string $file`. Путь к файлу. Если данный файл существует, он будет перезаписан.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **`$gbase->SetCurrency`**

Назначение:

- Установка трехбуквенного кода валюты, в которой будут представлены цены товаров в XML-файле (обычно - USD).

Объявление:

- `function SetCurrency($currency)`

Параметры:

- **string \$currency**. Трехбуквенный код валюты.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$gbase->SetShopDescr`**

Назначение:

- Установка описания магазина для его записи в XML-файл.

Объявление:

- `function SetShopDescr($shopDescr)`

Параметры:

- **string \$shopDescr**. Описание магазина.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$gbase->SetShopName`**

Назначение:

- Установка названия магазина для его записи в XML-файл.

Объявление:

- `function SetShopName($shopName)`

Параметры:

- **string \$shopName**. Название магазина.

Возвращаемое значение:

- Функция ничего не возвращает.

## `$gbase->SetShopUrl`

Назначение:

- Установка URL-адреса магазина для его записи в XML-файл.

Объявление:

- `function SetShopUrl($shopUrl)`

Параметры:

- **string \$shopUrl**. URL-адрес магазина.

Возвращаемое значение:

- Функция ничего не возвращает.

## Класс `HomePage`

Предназначен для управления HTML-текстом главной страницы магазина.

### `HomePage::GetContent`

Назначение:

- Извлечение контента главной страницы магазина из БД (таблица `proshop_homepage_texts`).

Объявление:

- `function GetContent($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текст главной страницы.

Возвращаемое значение:

- Тип **string**. HTML-текст главной страницы магазина.

### `HomePage::UpdateContent`

Назначение:

- Сохранение HTML-текста главной страницы магазина для указанного языка в таблице [proshop\\_homepage\\_texts](#).

Объявление:

- `function UpdateContent($lang_id, $content)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо установить контент главной страницы.
- **string \$content**. HTML-текст главной страницы на указанном языке.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс InterShipper

Предназначен для взаимодействия с онлайн-сервисом по расчету стоимости доставки компанией InterShipper. Более подробную информацию о форматах передачи данных на сервис InterShipper смотрите на сайте [intershipper.com](http://intershipper.com).

### `$is->getCurrency`

Назначение:

- Определение валюты, в которой задана стоимость доставки указанным сервисом. Стоимость доставки можно определить с помощью метода [\\$is->getPrice](#).

Объявление:

- `function getCurrency($service)`

Параметры:

- **string \$service**. Название указанного сервиса из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **string**. Международное трехбуквенное обозначение валюты.

### `$is->getDates`

Назначение:

- Определение предполагаемой даты доставки заказа для указанного сервиса (или диапазона предполагаемых дат).

Объявление:

- `function getDates($service)`

Параметры:

- **string \$service**. Название указанного сервиса из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **string**. Дата (или диапазон дат) предполагаемой доставки заказа.

### `$is->getPrice`

Назначение:

- Извлечение стоимости доставки заказа указанным сервисом в валюте, которая определена с помощью метода [\\$is->getCurrency](#).

Объявление:

- `function getPrice($service)`

Параметры:

- **string \$service**. Название указанного сервиса из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **float**. Стоимость доставки заказа или -1, если стоимость определить не удалось.

## **`$is->sendRequest`**

Назначение:

- Отправка XML-запроса на сервер InterShipper, сформированного вызовом других методов класса, и запись XML-ответа во внутреннюю переменную **reqResult**.

Объявление:

- `function sendRequest()`

Параметры:

- Функция не принимает параметров

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setARBAccountNumber`**

Назначение:

- Установка номера аккаунта в службе AirBorne.

Объявление:

- `function setARBAccountNumber($ARB_Account)`

Параметры:

- **string \$ARB\_Account**. Номер аккаунта в службе AirBorne.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setARBInvoiced`**

Назначение:

- Установка значения параметра **CarrierInvoiced** для аккаунта в службе AirBorne.

Объявление:

- `function setARBInvoiced($ARB_Invoiced)`

Параметры:

- **bool \$ARB\_Invoiced**. Значение параметра **CarrierInvoiced**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setCANAccountNumber`**

Назначение:

- Установка номера аккаунта в службе Canada P.S.

Объявление:

- `function setCANAccountNumber($CAN_Account)`

Параметры:

- **string \$CAN\_Account**. Номер аккаунта в службе Canada P.S.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setCANInvoiced`**

Назначение:

- Установка значения параметра **CarrierInvoiced** для аккаунта в службе Canada P.S.

Объявление:

- `function setCANInvoiced($CAN_Invoiced)`

Параметры:

- **bool \$CAN\_Invoiced**. Значение параметра **CarrierInvoiced**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setDeliveryType`**

Назначение:

- Установка типа доставки – коммерческая или обычная.

Объявление:

- `function setDeliveryType($DeliveryType)`

Параметры:

- **string \$DeliveryType**. Данный параметр принимает одно из следующих значений: **COM** – доставка коммерческая, **RES** – доставка обычная.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setDestCountry`**

Назначение:

- Установка страны адреса назначения доставки заказа.

Объявление:

- `function setDestCountry($DestCountry)`

Параметры:

- **string \$DestCountry**. Двубуквенное обозначение страны, в которую необходимо доставить заказ.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setDestPostal`**

Назначение:

- Установка индекса адреса назначения доставки заказа.

Объявление:

- `function setDestPostal($DestPostal)`



Параметры:

- **string \$DestPostal.** Индекс адреса назначения.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$is->setDHLAccountNumber**

Назначение:

- Установка номера аккаунта в службе DHL.

Объявление:

- function setDHLAccountNumber(\$DHL\_Account)

Параметры:

- **string \$DHL\_Account.** Номер аккаунта в службе DHL.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$is->setDHLInvoiced**

Назначение:

- Установка значения параметра **CarrierInvoiced** для аккаунта в службе DHL.

Объявление:

- function setDHLInvoiced(\$DHL\_Invoiced)

Параметры:

- **bool \$DHL\_Invoiced.** Значение параметра **CarrierInvoiced**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$is->setDimHeight**

Назначение:

- Установка высоты коробки в сантиметрах или дюймах (в зависимости от установленного значения с помощью [\\$is->setDimUnit](#)).

Объявление:

- function setDimHeight(\$Height)

Параметры:

- **float \$Height.** Значение высоты коробки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$is->setDimLength**

Назначение:

- Установка длины коробки в сантиметрах или дюймах (в зависимости от установленного значения с помощью [\\$is->setDimUnit](#)).

Объявление:

- `function setDimLength($Length)`

Параметры:

- **float \$Length**. Значение длины коробки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setDimUnit`**

Назначение:

- Установка единицы измерения длины/высоты/ширины коробки – в дюймах или в сантиметрах.

Объявление:

- `function setDimUnit($Unit)`

Параметры:

- **string \$Unit**. Данный параметр может принимать одно из следующих значений: **CM** - сантиметры, **IN** – дюймы.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setDimWidth`**

Назначение:

- Установка ширины коробки в сантиметрах или дюймах (в зависимости от установленного значения с помощью [\\$is->setDimUnit](#)).

Объявление:

- `function setDimWidth($Width)`

Параметры:

- **float \$Width**. Значение ширины коробки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setFDXAccountNumber`**

Назначение:

- Установка номера аккаунта в службе FedEx.

Объявление:

- `function setFDXAccountNumber($FDX_Account)`

Параметры:

- **string \$FDX\_Account**. Номер аккаунта в службе FedEx.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setFDXInvoiced`**

Назначение:

- Установка значения параметра **CarrierInvoiced** для аккаунта в службе FedEx.

Объявление:

- `function setFDXInvoiced($FDX_Invoiced)`

Параметры:

- **bool \$FDX\_Invoiced**. Значение параметра **CarrierInvoiced**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setOptions`**

Назначение:

- Установка массива опций доставки. Для справки о возможных значениях опций (OptionCode1, OptionCode2, ...) см. документацию на сайте [intershipper.com](http://intershipper.com).

Объявление:

- `function setOptions($Options)`

Параметры:

- **array \$Options**. Массив опций доставки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setOrigCountry`**

Назначение:

- Установка страны адреса назначения, куда необходимо доставить заказ.

Объявление:

- `function setOrigCountry($OrigCountry)`

Параметры:

- **string \$OrigCountry**. Международное двубуквенное обозначение страны.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setOrigPostal`**

Назначение:

- Установка индекса адреса назначения, куда необходимо доставить заказ.

Объявление:

- `function setOrigPostal($OrigPostal)`

Параметры:

- **string \$OrigPostal**. Значение индекса.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setPackaging`**

Назначение:

- Установка типа упаковки заказа. Для справки о возможных значениях (параметр **Packaging1**) см. документацию на сайте [intershipper.com](http://intershipper.com).

Объявление:

- `function setPackaging($Packaging)`

Параметры:

- **string \$Packaging**. Тип упаковки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setPassword`**

Назначение:

- Установка пароля для доступа к системе **InterShipper**.

Объявление:

- `function setPassword($Password)`

Параметры:

- **string \$Password**. Строка пароля.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setServer`**

Назначение:

- Установка URL-адреса скрипта системы InterShipper, на который необходимо послать запрос.

Объявление:

- `function setServer($Server)`

Параметры:

- **string \$Server**. URL-адрес скрипта.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setShipMethod`**

Назначение:

- Установка кода метода доставки заказа. Для справки о возможных значениях (параметр **ShipMethod**) см. документацию на сайте [intershipper.com](http://intershipper.com).

Объявление:

- `function setShipMethod($ShipMethod)`

Параметры:

- **string \$ShipMethod**. Код метода доставки.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setUPSAccountNumber`**

Назначение:

- Установка номера аккаунта в службе UPS.

Объявление:

- `function setUPSAccountNumber($UPS_Account)`

Параметры:

- **string \$UPS\_Account**. Номер аккаунта в службе UPS.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setUPSInvoiced`**

Назначение:

- Установка значения параметра **CarrierInvoiced** для аккаунта в службе UPS.

Объявление:

- `function setUPSInvoiced($UPS_Invoiced)`

Параметры:

- **bool \$UPS\_Invoiced**. Значение параметра **CarrierInvoiced**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setUsername`**

Назначение:

- Установка имени пользователя для доступа к системе **InterShipper**.

Объявление:

- `function setUsername($Username)`

Параметры:

- **string \$Username**. Имя пользователя.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setUSPAccountNumber`**

Назначение:

- Установка номера аккаунта в службе U.S.P.S.

Объявление:

- `function setUSPAccountNumber($USP_Account)`

Параметры:

- **string \$USP\_Account.** Номер аккаунта в службе USP.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setUSPInvoiced`**

Назначение:

- Установка значения параметра **CarrierInvoiced** для аккаунта в службе U.S.P.S.

Объявление:

- `function setUSPInvoiced($USP_Invoiced)`

Параметры:

- **bool \$USP\_Invoiced.** Значение параметра **CarrierInvoiced**.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$is->setWeight`**

Назначение:

- Установка веса заказа в килограммах.

Объявление:

- `function setWeight($Weight)`

Параметры:

- **float \$Weight.** Значение веса.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс Labels**

Предназначен для управления разноязычными текстовыми метками магазина.

### **Labels::Add**

Назначение:

- Добавление информации о метке в БД.

Объявление:

- `function Add($label_info)`

Параметры:

- **array \$label\_info**. Массив с информацией о метке. Должен содержать следующие ключи:
  - **eng\_name** – служебное имя метки;
  - **zone** – зона, в которую добавляется метка (**A** - админка или **C** - витрина);
  - идентификатор языка с информацией о тексте метки на указанном языке. См. пример.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

Пример:

- ```
$info = array(
    'eng_name' => 'lbl_test_label',
    'zone' => 'C',
    1 => array('caption' => 'Текст метки для языка с идентификатором 1'),
    2 => array('caption' => 'Текст метки для языка с идентификатором 2'),
    // ...
);
Labels::Add($info);
```

## **Labels::Delete**

Назначение:

- Удаление указанной метки из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор метки из таблицы [proshop\\_labels](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Labels::GetAdminCount**

Назначение:

- Определение кол-ва меток в зоне A (в админке).

Объявление:

- `function GetAdminCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Количество меток.

## Labels::GetAll

Назначение:

- Извлечение всех меток из БД.

Объявление:

- `function GetAll($zone = "", $lang_id = 0)`

Параметры:

- **char \$zone**. Зона, метки которой необходимо извлечь (**A** – для админки, **C** – для витрины). Если значение данного параметра пустое, возвращаются метки обеих зон.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь тексты меток. Если данный параметр равен 0, извлекаются только поля таблицы [proshop\\_labels](#). Иначе – поля **eng\_name** и **zone** таблицы [proshop\\_labels](#) и поле **caption** таблицы [proshop\\_labels\\_texts](#).

Возвращаемое значение:

- Тип **array**. Если значение параметра **\$lang\_id** равно нулю, то возвращается массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_labels](#). Иначе – возвращается массив, каждый элемент которого является массивом с ключами:
  - **eng\_name** – служебное имя метки;
  - **zone** – зона метки;
  - **caption** – текст метки на указанном языке.

Пример:

- ```
$labels = Labels::GetAll('C', $SHOP_LANGUAGE['id']);
foreach ($labels as $eng_name => $label) {
    // $eng_name – служебное имя метки
    // $label['caption'] – текст метки
    // ...
}
```

## Labels::GetAllEmpty

Назначение:

- Извлечение всех меток из БД, тексты которых на указанном языке не заданы.

Объявление:

- `function GetAllEmpty($lang_id, $cur_page = "", $page_size = "")`

Параметры:

- **int \$lang\_id**. Идентификатор языка.
- **int \$cur\_page**. Номер страницы списка найденных меток (для пейджинга, класса [Pager](#)).
- **int \$page\_size**. Кол-во меток на одной странице найденных меток (для пейджинга, класса [Pager](#)).

Возвращаемое значение:



- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_labels](#).

Пример:

- `$empty_labels = Labels::GetAllEmpty($SHOP_LANGUAGE['id'], $pager->getCurrentPage(), $pager->getPageSize());`

## **Labels::GetCaption**

Назначение:

- Определение текста указанной метки в указанной зоне и на указанном языке.

Объявление:

- `function GetCaption($name, $zone, $lang_id)`

Параметры:

- **string \$name**. Служебное имя метки.
- **char \$zone**. Зона (**A** – админка, **C** - витрина), для которой необходимо определить текст указанной метки **\$name**.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо определить текст для указанной метки **\$name**.

Возвращаемое значение:

- Тип **string/bool**. **False** – в случае ошибки (например, текст метки не задан для указанной зоны). Иначе – текст метки для указанной зоны и указанного языка.

## **Labels::GetCustomerCount**

Назначение:

- Определение кол-ва меток в зоне C (на витрине).

Объявление:

- `function GetCustomerCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Количество меток.

## **Labels::GetOne**

Назначение:

- Извлечение информации об указанной метке из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор метки из таблицы [proshop\\_labels](#).

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь название метки из таблицы [proshop\\_labels\\_texts](#). Если данный параметр равен 0, название метки не извлекается.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив со следующими ключами:
  - все поля таблицы [proshop\\_labels](#);
  - поле **caption** таблицы [proshop\\_labels\\_texts](#) (если параметр **\$lang\_id** не равен нулю).

## **Labels::GetTotalCount**

Назначение:

- Определение общего кол-ва меток в БД (в таблице [proshop\\_labels](#)).

Объявление:

- `function GetTotalCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во меток.

## **Labels::IsNameExists**

Назначение:

- Определение, не занято ли служебное имя метки для указанной зоны.

Объявление:

- `function IsNameExists($name, $zone, $id = 0)`

Параметры:

- **string \$name**. Имя метки.
- **char \$zone**. Зона, существование имени метки в которой необходимо проверить (**A** – в админке, **C** – на витрине). Если данный параметр пуст, существование имени проверяется в любой зоне.
- **int \$id**. Идентификатор метки, которую необходимо проигнорировать при поиске указанного имени. Если данный параметр равен 0, учитываются все метки.

Возвращаемое значение:

- Тип **int**. Идентификатор метки из таблицы [proshop\\_labels](#) или 0, если имя не занято.

## **Labels::Search**

Назначение:

- Поиск метки по тексту, который может в ней содержаться, либо по ее служебному имени, в указанной зоне на указанном языке.

Объявление:

- `function Search($find_text, $zone, $lang_id, $pager_sql = "")`

Параметры:

- **string \$find\_text.** Текст (либо его часть), который необходимо найти, либо служебное имя метки.
- **char \$zone.** Зона поиска (**A** – админка, **C** - витрина). Если данный параметр пуст, метка ищется в обеих зонах.
- **int \$lang\_id.** Идентификатор языка, для которого необходимо произвести поиск.
- **string \$pager\_sql.** Часть SQL-запроса, которую можно добавить к основному SQL-запросу. Используется пейджингом (класс [Pager](#)).

Возвращаемое значение:

- Тип **array.** Массив, каждый элемент которого является массивом, описывающим найденную метку, и содержащим следующие ключи:
  - все поля таблицы [proshop\\_labels](#);
  - поле **caption** таблицы [proshop\\_labels\\_texts](#).

## **Labels::SetCaption**

Назначение:

- Установка текста указанной метки на указанном языке.

Объявление:

- function SetCaption(\$id, \$caption, \$lang\_id)

Параметры:

- **int \$id.** Идентификатор метки из таблицы [proshop\\_labels](#).
- **string \$caption.** Текст метки на языке **\$lang\_id**. Может содержать HTML-код.
- **int \$lang\_id.** Идентификатор языка, для которого необходимо установить текст метки.

Возвращаемое значение:

- Тип **bool.** True/false – в зависимости от успешности операции.

## **Labels::Update**

Назначение:

- Обновление информации об указанной метке.

Объявление:

- function Update(\$id, \$label\_info)

Параметры:

- **int \$id.** Идентификатор метки из таблицы [proshop\\_labels](#).
- **array \$label\_info.** См. описание параметра функции [Labels::Add](#).

Возвращаемое значение:

- Тип **bool.** True/false – в зависимости от успешности операции.

## **Класс Languages**

Предназначен для управления языками магазина.

## Languages : : Add

Назначение:

- Добавление языка в БД (в таблицу [proshop\\_languages](#)).

Объявление:

- `function Add($title, $short, $flag, $curr_id)`

Параметры:

- **string \$title**. Название языка. Макс. 254 символа.
- **string \$short**. Международное двубуквенное обозначение языка (английские буквы нижнего регистра).
- **string \$flag**. Имя файла изображения флага языка, который располагается в директории, определенной в константе [FLAGS\\_PICS](#). Макс. 20 символов.
- **int \$curr\_id**. Идентификатор валюты из таблицы [proshop\\_currencies](#), которая будет автоматически выбрана на витрине при выборе данного языка.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор добавленного языка.

## Languages : : BindCurrencies

Назначение:

- Привязка идентификаторов валют к заданному языку. При выборе указанного языка на витрине можно будет выбирать только те валюты, которые указаны в параметре к данной функции.

Объявление:

- `function BindCurrencies($lang_id, $currencies)`

Параметры:

- **int \$lang\_id**. Идентификатор языка.
- **array \$currencies**. Массив идентификаторов валют (идентификаторы хранятся в таблице [proshop\\_currencies](#)), которые необходимо привязать к заданному языку.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Languages : : CopyTexts

Назначение:

- Копирование всех языкозависимых текстов в БД с одного языка на другой. При копировании затрагиваются таблицы, оканчивающиеся на **\_texts** и имеющие поле **lang\_id**.

Объявление:

- `function CopyTexts($from_id, $to_id)`

Параметры:

- **int \$from\_id**. Идентификатор языка, тексты которого необходимо скопировать на указанный язык с идентификатором **\$to\_id**.

- **int \$to\_id.** Идентификатор языка, для которого необходимо записать тексты, скопированные с языка **\$from\_id**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Languages::Delete**

Назначение:

- Удаление указанного языка и всех языкозависимых записей, ассоциированных с ним, из БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), который необходимо удалить.

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Languages::Exists**

Назначение:

- Определение, существует ли указанный язык в БД.

Объявление:

- function Exists(\$id)

Параметры:

- **int \$id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), который необходимо проверить на существование.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Languages::GetAll**

Назначение:

- Извлечение всех языков из таблицы [proshop\\_languages](#).

Объявление:

- function GetAll()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_languages](#).

## **Languages::GetCurrencies**

Назначение:

- Извлечение списка идентификаторов валют, привязанных к заданному языку.

Объявление:

- `function GetCurrencies($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является идентификатором валюты (валюты хранятся в таблице [proshop\\_currencies](#)).

## **Languages::GetDefault**

Назначение:

- Извлечение информации о языке магазина, выбранном администратором по умолчанию.

Объявление:

- `function GetDefault()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_languages](#), - информация о языке по умолчанию (язык, значение поля **is\_default** которого установлено в **Y**).

## **Languages::GetDefaultID**

Назначение:

- Определение идентификатора языка, выбранного администратором по умолчанию.

Объявление:

- `function GetDefaultID()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Идентификатор языка по умолчанию (язык, значение поля **is\_default** которого установлено в **Y**) из таблицы [proshop\\_languages](#).

## **Languages::GetIDByShort**

Назначение:

- Определение идентификатора языка по его международному двубуквенному обозначению.

Объявление:

- `function GetIDByShort($short)`

Параметры:

- **string \$short**. Двубуквенное обозначение языка (например, для русского - ru).

Возвращаемое значение:

- Тип **int**. Идентификатор языка из таблицы [proshop\\_languages](#).

## **Languages::GetIDByTitle**

Назначение:

- Определение идентификатора языка по его названию.

Объявление:

- `function GetIDByTitle($title)`

Параметры:

- **string \$title**. Название языка.

Возвращаемое значение:

- Тип **int**. Идентификатор языка из таблицы [proshop\\_languages](#).

## **Languages::GetOne**

Назначение:

- Извлечение информации об указанном языке.

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_languages](#).

## **Languages::GetShortByID**

Назначение:

- Определение международного двубуквенного обозначения языка по его идентификатору из таблицы [proshop\\_languages](#).

Объявление:

- `function GetShortByID($id)`

Параметры:

- **int \$id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string**. Международное двубуквенное обозначение языка (две английских буквы нижнего регистра). Например, для русского – **ru**.

## **Languages::GetTitleByID**

Назначение:

- Определение названия языка по его идентификатору из таблицы [proshop\\_languages](#).

Объявление:

- `function GetTitleByID($id)`

Параметры:

- **int \$id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string**. Название указанного языка.

## **Languages::IsCurrencyAllowed**

Назначение:

- Определение, разрешено ли отображение указанной валюты на витрине совместно с указанным языком.

Объявление:

- `function IsCurrencyAllowed($lang_id, $curr_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **int \$curr\_id**. Идентификатор валюты из таблицы [proshop\\_currencies](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Languages::SetDefaultID**

Назначение:

- Установка указанного языка в качестве языка по умолчанию.

Объявление:

- `function SetDefaultID($id)`

Параметры:

- **int \$id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), который необходимо сделать языком по умолчанию.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Languages::Update**

Назначение:

- Обновление информации об указанном языке в таблице [proshop\\_languages](#).

Объявление:

- `function Update($id, $lang_info)`

Параметры:

- **int \$id**. Идентификатор языка, информацию о котором необходимо обновить.
- **array \$lang\_info**. Массив с информацией, который должен содержать следующие ключи:
  - **title** – название языка. Макс. 254 символа;
  - **short** – двухбуквенное международное обозначение языка (две английские буквы нижнего регистра);



- **flag** - имя файла изображения флага языка, который располагается в директории, определенной в константе [FLAGS\\_PICS](#). Макс. 20 символов.
- **curr\_id** – идентификатор валюты из таблицы [proshop\\_currencies](#), которая будет автоматически выбрана на витрине при выборе данного языка.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс Links

Предназначен для формирования ссылок на страницы различных модулей на витрине.

### **\$LINKS->Account**

Назначение:

- Формирование ссылки на указанную страницу модуля витрины **account**.

Объявление:

- `function Account($act = "")`

Параметры:

- **string \$act**. Действие модуля, адрес страницы которого необходимо сгенерировать. Если значение данного параметра пустое, генерируется адрес главной страницы модуля.

Возвращаемое значение:

- Тип **string**. Сформированный адрес страницы.

### **\$LINKS->AccountActivate**

Назначение:

- Формирование ссылки активации пользовательского аккаунта на витрине после его регистрации.

Объявление:

- `function AccountActivate($key)`

Параметры:

- **string \$key**. Ключ активации – значение поля **activation\_key** таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **string**. Ссылка активации аккаунта.

### **\$LINKS->AccountPwdRecover**

Назначение:

- Формирование ссылки на секретную страницу установки нового пароля пользовательского аккаунта на витрине.

Объявление:

- function AccountPwdRecover(\$key)

Параметры:

- **string \$key**. Ключ восстановления – значение поля **pwd\_recover\_key** таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **string**. Ссылка на страницу установки нового пароля пользователя.

## **\$LINKS->Article**

Назначение:

- Формирование ссылки на статью.

Объявление:

- function Article(\$art\_info)

Параметры:

- **array \$art\_info**. Массив с информацией о HTML-странице (из таблицы [proshop\\_articles](#)).

Возвращаемое значение:

- Тип **string**. Ссылка на страницу.

## **\$LINKS->Articles**

Назначение:

- Формирование ссылки на страницу со списком статей (на страницу рубрики).

Объявление:

- function Articles(\$cat\_info = null, \$page = 1)

Параметры:

- **array \$cat\_info**. Массив с информацией о категории (рубрике) статей (из таблицы [proshop\\_articles\\_categories](#)). Если данный параметр равен **null**, возвращается ссылка на корневую страницу категорий.
- **int \$page**. Номер страницы при постраничной навигации по статьям категории (рубрики).

Возвращаемое значение:

- Тип **string**. Ссылка на страницу статей.

## **\$LINKS->Bestsellers**

Назначение:

- Формирование ссылки на страницу списка лидеров продаж (бестселлеров).

Объявление:

- function Bestsellers()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы бестселлеров.

## **\$LINKS->Cart**

Назначение:

- Формирование ссылки на страницу корзины (процесса оформления заказа).

Объявление:

- `function Cart($act = "")`

Параметры:

- **string \$act**. Действие модуля корзины, для которого необходимо сформировать адрес страницы.

Возвращаемое значение:

- Тип **string**. Адрес страницы корзины.

## **\$LINKS->CompareProds**

Назначение:

- Формирование адреса страницы сравнения продуктов пользователем.

Объявление:

- `function CompareProds($act = "")`

Параметры:

- **string \$act**. Название действия, которое нужно произвести в модуле сравнения товаров на витрине.

Возвращаемое значение:

- Тип **string**. URL-адрес страницы сравнения продуктов (или URL-адрес выполнения какого-либо действия в модуле сравнения продуктов, если задан параметр **\$act**).

## **\$LINKS->Contacts**

Назначение:

- Формирование ссылки на страницу контактов.

Объявление:

- `function Contacts()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы контактов.

## **\$LINKS->GetEgoodFile**

Назначение:

- Формирование ссылки на скачивание электронного продукта.

Объявление:

- `function GetEgoodFile($file_id, $lang_short, $key)`

Параметры:

- **int \$file\_id.** Идентификатор файла продукта, хранящийся в таблице [proshop\\_products\\_files](#).
- **string \$lang\_short.** Международное двухбуквенное обозначение языка, на котором откроется страница скачивания в случае возникновения ошибки (например, файл не найден).
- **string \$key.** Ключ для скачивания электронного продукта, который хранится в таблице [proshop\\_products\\_files\\_keys](#).

Возвращаемое значение:

- Тип **string**. Ссылка на скачивание электронного продукта.

## **\$LINKS->GetEgoodFree**

Назначение:

- Формирование ссылки на страницу скачивания бесплатного электронного продукта.

Объявление:

- function GetEgoodFree(\$cat\_id, \$prod\_id)

Параметры:

- **int \$cat\_id.** Идентификатор категории продукта из таблицы [proshop\\_categories](#).
- **int \$prod\_id.** Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **string**. Адрес страницы скачивания бесплатного электронного продукта.

## **\$LINKS->Home**

Назначение:

- Формирование ссылки на главную страницу витрины магазина.

Объявление:

- function Home()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес главной страницы витрины магазина.

## **\$LINKS->Manufacturer**

Назначение:

- Формирование ссылки на страницу описания производителя.

Объявление:

- function Manufacturer(\$eng\_name, \$page = 1)

Параметры:

- **string \$eng\_name.** Служебное имя (имя в URL для modRewrite) производителя из таблицы [proshop\\_manufacturers](#).
- **int \$page.** Номер страницы при постраничной навигации по продуктам производителя.

Возвращаемое значение:

- Тип **string**. Адрес страницы описания производителя.

## **\$LINKS->Manufacturers**

Назначение:

- Формирование ссылки на страницу списка производителей.

Объявление:

- `function Manufacturers()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы списка производителей.

## **\$LINKS->Newest**

Назначение:

- Формирование адреса страницы со списком новинок магазина.

Объявление:

- `function Newest()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы со списком новинок.

## **\$LINKS->News**

Назначение:

- Формирование ссылки на страницу со списком новостей или на страницу указанной новости.

Объявление:

- `function News($fact_id = 0)`

Параметры:

- **int \$fact\_id**. Идентификатор новости, адрес страницы которой требуется сформировать. Если значение данного параметра нулевое, формируется адрес страницы со списком новостей.

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## **\$LINKS->NewsArchive**

Назначение:

- Формирование ссылки на страницу с архивом новостей.

Объявление:

- `function NewsArchive()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. URL-адрес страницы архива новостей.

## **\$LINKS->NewsConfirm**

Назначение:

- Формирование ссылки подтверждения подписки на новости.

Объявление:

- function NewsConfirm(\$key, \$lang\_short)

Параметры:

- **string \$key**. Ключ подтверждения из таблицы [proshop\\_news\\_subscribers](#).
- **string \$lang\_short**. Двубуквенное международное обозначение языка, на котором необходимо показать страницу, информирующую об успешности подписки.

Возвращаемое значение:

- Тип **string**. Ссылка подтверждения подписки на новости.

## **\$LINKS->NewsRSS**

Назначение:

- Формирование адреса RSS-канала новостей магазина.

Объявление:

- function NewsRSS()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес RSS-канала новостей.

## **\$LINKS->NewsSubscribe**

Назначение:

- Формирование адреса страницы подписки на новости.

Объявление:

- function NewsSubscribe()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## **\$LINKS->NewsUnsubscribe**

Назначение:

- Формирование адреса страницы отписки от рассылки новостей.

Объявление:

- `function NewsUnsubscribe($key, $lang_short)`

Параметры:

- **string \$key**. Ключ подписки на новости из таблицы [proshop\\_news\\_subscribers](#).
- **string \$lang\_short**. Международное двубуквенное обозначение языка (две английских буквы нижнего регистра) страницы отписки.

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## **\$LINKS->OrderStatus**

Назначение:

- Формирование адреса страницы, на которой пользователь может определить статус заказа по его номеру.

Объявление:

- `function OrderStatus()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## **\$LINKS->Partner**

Назначение:

- Формирование партнерской ссылки, по которой можно привлекать покупателей в магазин за вознаграждение. Для модуля партнерской программы.

Объявление:

- `function Partner($user_id)`

Параметры:

- **int \$user\_id**. Идентификатор пользователя (из таблицы [proshop\\_users](#)), который приглашает других покупателей на сайт.

Возвращаемое значение:

- Тип **string**. Партнерская ссылка.

## **\$LINKS->PriceList**

Назначение:

- Формирование адреса страницы прайс-листа на витрине.

Объявление:

- `function PriceList()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы прайс-листа на витрине.

## **\$LINKS->Product**

Назначение:

- Формирование адреса страницы описания продукта.

Объявление:

- `function Product($prod_info_id, $again = false, $from = "")`

Параметры:

- **array/int \$prod\_info\_id**. Массив с информацией о продукте из таблицы [proshop\\_products](#), либо идентификатор продукта из этой таблицы.
- **bool \$again**. Флаг повторной загрузки страницы. Используется для того, чтобы сохранить введенные пользователем значения (в сессии) до возникновения ошибки, и вывести их на той же самой странице после возникновения ошибки. Обычно равен **false**.
- **string \$from**. Используется для обозначения того, откуда пользователь переходит по ссылке на страницу продукта (из категории, из списка новинок, бестселлеров и т.п.).

Возвращаемое значение:

- Тип **string**. Адрес страницы описания продукта.

## **\$LINKS->Products**

Назначение:

- Формирование адреса страницы со списком продуктов заданной категории.

Объявление:

- `function Products($cat_id, $page_num = 1)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, адрес страницы которой необходимо получить.
- **int \$page\_num**. Номер страницы списка продуктов (при многостраничной навигации).

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## **\$LINKS->ProductsByManuf**

Назначение:

- Формирование адреса страницы со списком продуктов заданного производителя (при применении фильтра по производителям).

Объявление:

- `function ProductsByManuf($cat_id, $manuf_id, $page_num = 1)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, адрес страницы которой необходимо получить.



- **int \$manuf\_if.** Идентификатор производителя.
- **int \$page\_num.** Номер страницы списка продуктов (при многостраничной навигации).

Возвращаемое значение:

- Тип **string.** Адрес страницы.

## **\$LINKS->ProductsByPrice**

Назначение:

- Формирование адреса страницы со списком продуктов заданной категории в заданном ценовом диапазоне.

Объявление:

- `function ProductsByPrice($cat_id, $from_price, $to_price, $page_num = 1)`

Параметры:

- **int \$cat\_id.** Идентификатор категории, адрес страницы которой необходимо получить.
- **int \$from\_price.** Начальное значение ценового диапазона в базовой валюте магазина.
- **int \$to\_price.** Конечное значение ценового диапазона в базовой валюте магазина.
- **int \$page\_num.** Номер страницы списка продуктов (при многостраничной навигации).

Возвращаемое значение:

- Тип **string.** Адрес страницы.

## **\$LINKS->ReviewAdd**

Назначение:

- Формирование адреса страницы добавления отзыва о сайте.

Объявление:

- `function ReviewAdd()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string.** Адрес страницы.

## **\$LINKS->Reviews**

Назначение:

- Формирование ссылки на страницу отзывов о сайте.

Объявление:

- `function Reviews($again = false)`

Параметры:

- **bool \$again.** Флаг повторной загрузки страницы. Используется для того, чтобы сохранить введенные пользователем значения (в сессии) до возникновения ошибки, и вывести их на той же самой странице после возникновения ошибки. Обычно равен **false**.

Возвращаемое значение:

- Тип **string**. Адрес страницы отзывов.

## **\$LINKS->Search**

Назначение:

- Формирование адреса страницы с формой поиска и с результатами поиска продуктов на витрине.

Объявление:

- `function Search($act, $page_num = 1)`

Параметры:

- **string \$act**. Принимает одно из двух значений: **show\_sform** – для показа формы поиска, **search** – для вывода страницы с результатами поиска.
- **int \$page\_num**. Номер страницы со списком найденных продуктов при многостраничном выводе. Данный параметр игнорируется, если значение параметра **\$act** не равно **search**.

## **\$LINKS->SiteMap**

Назначение:

- Формирование адреса страницы карты сайта на витрине.

Объявление:

- `function SiteMap()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## **\$LINKS->Tags**

Назначение:

- Формирование адреса страницы с тэгами, либо адреса страницы, содержащей список продуктов для указанного тэга.

Объявление:

- `function Tags($tag_id = "", $page = 1)`

Параметры:

- **int \$tag\_id**. Идентификатор тэга из таблицы [proshop\\_tags\\_texts](#). Если данный параметр пуст, возвращается адрес страницы, содержащей список тэгов.
- **int \$page**. Номер страницы для постраничного вывода списка продуктов указанного тэга **\$tag\_id**. Данный параметр игнорируется, если **\$tag\_id** равен пустой строке.

Возвращаемое значение:

- Тип **string**. Адрес страницы.

## Класс Location

Предназначен для формирования строки навигации как на витрине, так и в админке.

### `$LOCATION->AddItem`

Назначение:

- Добавление элемента в строку навигации.

Объявление:

- `function AddItem($item)`

Параметры:

- **array \$item**. Массив с информацией о добавляемом элементе. Содержит следующие ключи:
  - **link** – адрес страницы, на которую ссылается добавляемый элемент. Может быть пустой строкой - в этом случае элемент не является ссылкой;
  - **link\_forced** – если равно `true`, элемент отображается как ссылка даже в том случае, если он последний в строке навигации;
  - **caption** – название элемента на текущем языке.

Возвращаемое значение:

- Функция ничего не возвращает.

### `$LOCATION->AppendItem`

Назначение:

- Аналог [\\$LOCATION->AddItem](#), но принимает параметры не в виде массива, а в виде отдельных переменных.

Объявление:

- `function AppendItem($link, $caption, $link_forced = false)`

Параметры:

- **string \$link**. Адрес страницы, на которую ссылается добавляемый элемент. Может быть пустой строкой - в этом случае элемент не является ссылкой.
- **bool \$link\_forced**. Если равно `true`, элемент отображается как ссылка даже в том случае, если он последний в строке навигации.
- **string \$caption**. Название элемента на текущем языке.

Возвращаемое значение:

- Функция ничего не возвращает.

### `$LOCATION->ClearList`

Назначение:

- Очистка списка элементов строки навигации (сброс строки навигации).

Объявление:

- function ClearList()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## Класс Manufacturers

Предназначен для управления производителями продуктов.

### Manufacturers::AddRoot

Назначение:

- Добавление базовой информации о производителе в таблицу [proshop\\_manufacturers](#).

Объявление:

- function AddRoot(\$eng\_name, \$website, \$visible)

Параметры:

- **string \$eng\_name**. Имя производителя в URL (для modRewrite).
- **string \$website**. Адрес сайта производителя.
- **char \$visible**. Флаг видимости производителя (**Y** – видим, **N** - невидим).

Возвращаемое значение:

- Тип **int**. Идентификатор добавленного производителя, либо 0 - в случае ошибки.

### Manufacturers::AddTexts

Назначение:

- Добавление текстовой языкозависимой информации о производителе (название, описание) в таблицу [proshop\\_manufacturers\\_texts](#).

Объявление:

- function AddTexts(\$id, \$lang\_id, \$alt\_text, \$caption, \$descr)

Параметры:

- **int \$id**. Идентификатор производителя из таблицы [proshop\\_manufacturers](#).
- **int \$lang\_id**. Идентификатор языка, для которого добавляется текстовая информация.
- **string \$alt\_text**. Текст атрибута **alt** тэга **img** для изображения производителя на указанном языке. Макс. 254 символа.
- **string \$caption**. Название производителя на указанном языке. Макс. 254 символа.
- **string \$descr**. Описание производителя. Может содержать код HTML. Макс. 65533 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Manufacturers::Delete**

Назначение:

- Удаление указанного производителя из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор производителя из таблицы [proshop\\_manufacturers](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Manufacturers::DeleteImage**

Назначение:

- Удаление изображения указанного производителя.

Объявление:

- `function DeleteImage($id)`

Параметры:

- **int \$id**. Идентификатор производителя из таблицы [proshop\\_manufacturers](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Manufacturers::GetAll**

Назначение:

- Извлечение всех производителей из БД.

Объявление:

- `function GetAll($lang_id = 0)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие производителей (такие, как название и описание). Если равен 0, текстовые составляющие не извлекаются.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_manufacturers](#);
  - **alt\_text** - текст атрибута **alt** тэга **img** для изображения производителя на указанном языке (если параметр **\$lang\_id** задан);
  - **caption** - название производителя на указанном языке (если параметр **\$lang\_id** задан);
  - **descr** - описание производителя на указанном языке (если параметр **\$lang\_id** задан).  
Может содержать код HTML.

Пример:

- ```
$rows = Manufacturers::GetAll($SHOP_LANGUAGE['id']);
foreach ($rows as $row) {
    // $row['id'] – идентификатор производителя
    // $row['caption'] – название производителя
    // ...
}
```

## **Manufacturers::GetAllVisible**

Назначение:

- Извлечение всех видимых на витрине производителей из БД (производителей, у которых значение поля **VISIBLE** таблицы [proshop\\_manufacturers](#) равно Y).

Объявление:

- `function GetAllVisible($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие производителей (такие, как название и описание).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_manufacturers](#);
  - **alt\_text** - текст атрибута **alt** тэга **img** для изображения производителя на указанном языке;
  - **caption** - название производителя на указанном языке;
  - **descr** - описание производителя. Может содержать код HTML.

Пример:

- ```
$rows = Manufacturers::GetAllVisible($SHOP_LANGUAGE['id']);
foreach ($rows as $row) {
    // $row['id'] – идентификатор производителя
    // $row['caption'] – название производителя
}
```

## **Manufacturers::GetIdByName**

Назначение:

- Определение идентификатора производителя по его имени в URL (для modRewrite).

Объявление:

- `function GetIdByName($eng_name)`

Параметры:

- **string \$eng\_name.** Имя в URL производителя из таблицы [proshop\\_manufacturers](#). Данный параметр может также быть идентификатором производителя.

Возвращаемое значение:

- Тип **int**. Идентификатор производителя из таблицы [proshop\\_manufacturers](#) или 0, если производитель не найден.

## **Manufacturers::GetOne**

Назначение:

- Извлечение информации об указанном производителе из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id.** Идентификатор производителя из таблицы [proshop\\_manufacturers](#).
- **int \$lang\_id.** Идентификатор языка, для которого необходимо извлечь текстовые составляющие производителя (его название и описание). Если значение данного параметра равно 0, текстовые составляющие не извлекаются.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе - массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_manufacturers](#);
  - **alt\_text** - текст атрибута **alt** тэга **img** для изображения производителя на указанном языке. Данное значение извлекается, только если параметр **\$lang\_id** больше нуля;
  - **caption** - название производителя на указанном языке. Данное значение извлекается, только если параметр **\$lang\_id** больше нуля;
  - **descr** - описание производителя. Может содержать код HTML. Данное значение извлекается, только если параметр **\$lang\_id** больше нуля.

Пример:

- ```
$row = Manufacturers::GetOne($id, $SHOP_LANGUAGE['id']);  
// $row['id'] – идентификатор производителя  
// $row['caption'] – название производителя  
// ...
```

## **Manufacturers::GetOneByName**

Назначение:

- Извлечение информации об указанном производителе из БД по его имени в URL (для `modRewrite`).

Объявление:

- `function GetOneByName($eng_name, $lang_id)`

Параметры:

- **string \$eng\_name.** Имя в URL производителя из таблицы [proshop\\_manufacturers](#).

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие производителя (его название и описание).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе - массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_manufacturers](#);
  - **alt\_text** - текст атрибута **alt** тэга **img** для изображения производителя на указанном языке;
  - **caption** - название производителя на указанном языке;
  - **descr** - описание производителя. Может содержать код HTML.

Пример:

- ```
$row = Manufacturers::GetOneByName($name, $SHOP_LANGUAGE['id']);  
// $row['id'] – идентификатор производителя  
// $row['caption'] – название производителя  
// ...
```

## **Manufacturers::IsExists**

Назначение:

- Определение, существует ли производитель с заданным идентификатором в таблице [proshop\\_manufacturers](#).

Объявление:

- `function IsExists($id)`

Параметры:

- **int \$id**. Идентификатор производителя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Manufacturers::IsCaptionExists**

Назначение:

- Проверка, не занято ли заданное название производителя для заданного языка.

Объявление:

- `function IsCaptionExists($caption, $lang_id, $id = 0)`

Параметры:

- **string \$caption**. Название производителя на языке **\$lang\_id**.
- **int \$lang\_id**. Идентификатор языка.
- **int \$id**. Идентификатор производителя, которого не нужно учитывать при поиске названия. Если данный параметр равен нулю, учитываются все производители.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.



## **Manufacturers::IsNameExists**

Назначение:

- Проверка, не занято ли заданное имя в URL производителя (для modRewrite).

Объявление:

- `function IsNameExists($name, $id = 0)`

Параметры:

- **string \$name.** Имя в URL производителя, которое необходимо проверить.
- **int \$id.** Идентификатор производителя, которого не нужно учитывать при поиске имени в URL. Если данный параметр равен нулю, учитываются все производители.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Manufacturers::SetCaption**

Назначение:

- Установка названия указанного производителя для указанного языка.

Объявление:

- `function SetCaption($id, $lang_id, $caption)`

Параметры:

- **int \$id.** Идентификатор производителя (из таблицы [proshop\\_manufacturers](#)), для которого необходимо установить название на языке **\$lang\_id**.
- **int \$lang\_id.** Идентификатор языка.
- **string \$caption.** Название производителя. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Manufacturers::SetImage**

Назначение:

- Установка имени файла изображения указанного производителя. Файлы изображений производителей хранятся в директории **images/manufacturers**.

Объявление:

- `function SetImage($id, $img)`

Параметры:

- **int \$id.** Идентификатор производителя из таблицы [proshop\\_manufacturers](#).
- **string \$img.** Имя файла изображения производителя. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Manufacturers::SetPos**

Назначение:

- Установка позиции производителя относительно других производителей (для сортировки в списке).

Объявление:

- `function SetPos($manuf_id, $pos)`

Параметры:

- **int \$manuf\_id**. Идентификатор производителя (из таблицы [proshop\\_manufacturers](#)).
- **int \$pos**. Позиция производителя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Manufacturers::setVisible**

Назначение:

- Установка флага видимости указанного производителя (видимость на витрине).

Объявление:

- `function setVisible($id, $mode)`

Параметры:

- **int \$id**. Идентификатор производителя из таблицы [proshop\\_manufacturers](#).
- **char \$mode**. Значение флага видимости (**Y** – видим, **N** - невидим).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Manufacturers::update**

Назначение:

- Обновление информации об указанном производителе.

Объявление:

- `function update($id, $info, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор производителя из таблицы [proshop\\_manufacturers](#), информацию о котором необходимо обновить.
- **array \$info**. Массив, содержащий следующий ключи:
  - **eng\_name** - имя производителя в URL (для modRewrite);
  - **website** – адрес сайта производителя;
  - **alt\_text** - текст атрибута **alt** тэга **img** для изображения производителя на указанном языке (если параметр **\$lang\_id** задан). Макс. 254 символа;
  - **caption** - название производителя на указанном языке (если параметр **\$lang\_id** задан). Макс. 254 символа;
  - **descr** - описание производителя на указанном языке (если параметр **\$lang\_id** задан). Может содержать код HTML. Макс. 65533 символа;
  - **VISIBLE** - флаг видимости производителя (**Y** – видим, **N** - невидим).

- **int \$lang\_id**. Идентификатор языка, для которого обновляется текстовая информация. Если равен 0, текстовая информация не обновляется.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **Manufacturers::UploadImage**

Назначение:

- Загрузка изображения указанного производителя.

Объявление:

- `function UploadImage($manuf_id, $img, $width)`

Параметры:

- **int \$manuf\_id**. Идентификатор производителя, для которого необходимо загрузить изображение.
- **file \$img**. Файл изображения из **\$\_FILES**. См. пример.
- **int \$width**. Максимальная ширина изображения. Если ширина загружаемого изображения больше данного значения, размеры изображения автоматически уменьшаются до заданного значения с сохранением пропорций.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

Пример:

- `Manufacturers::UploadImage($id, $_FILES['new_img'],  
CONFIG['settings']['manufacturer_img_width']);`

## Класс Messages

Предназначен для управления личными сообщениями между администрацией магазина и покупателями.

### **Messages::AddMessage**

Назначение:

- Добавление сообщения в заданную тему.

Объявление:

- `function AddMessage($topic_id, $message, $type)`

Параметры:

- **int \$topic\_id**. Идентификатор темы сообщений (из таблицы [proshop\\_msgs\\_topics](#)).
- **array \$message**. Массив с информацией о новом сообщении. Должен содержать следующие ключи:
  - **user\_id** – идентификатор пользователя, написавшего сообщение (из таблицы [proshop\\_users](#));

- **user\_name** – имя пользователя, написавшего сообщение;
- **user\_email** – e-mail пользователя, написавшего сообщение;
- **message** – текст сообщения.
- **string \$type**. Тип пользователя, который написал сообщение:
  - **admin** – администратор;
  - **customer** – покупатель.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Messages::AddTopic**

Назначение:

- Добавление темы сообщений в базу данных.

Объявление:

- function AddTopic(\$topic)

Параметры:

- **array \$topic**. Массив с информацией о добавляемой теме. Должен содержать следующие ключи:
  - **user\_id** – идентификатор пользователя (из таблицы [proshop\\_users](#)), создающего тему. Если пользователь не зарегистрирован на сайте, этот параметр должен быть равен 0;
  - **user\_name** – имя пользователя, создающего тему;
  - **user\_email** – e-mail пользователя, создавшего тему;
  - **title** – название темы.

Возвращаемое значение:

- Тип **int/bool**. В случае успеха – идентификатор созданной темы (из таблицы [proshop\\_msgs\\_topics](#)). В случае ошибки – **false**.

## **Messages::GetTopicById**

Назначение:

- Извлечение информации о теме сообщений из базы данных по ее уникальному идентификатору.

Объявление:

- function GetTopicById(\$id)

Параметры:

- **int \$id**. Идентификатор темы из таблицы [proshop\\_msgs\\_topics](#).

Возвращаемое значение:

- Тип **array/bool**. False - в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - ключи, имена которых совпадают с полями таблицы [proshop\\_msgs\\_topics](#).

- **messages** – массив с информацией о сообщениях, которые привязаны к данной теме. Ключи каждого элемента этого массива совпадают с полями таблицы [proshop\\_msgs](#).

## **Messages::GetTopicsByStatus**

Назначение:

- Извлечение из базы данных тем сообщений, которые имеют заданный статус.

Объявление:

- `function GetTopicsByStatus($status)`

Параметры:

- **char \$status**. Статус тем сообщений:
  - **N** – новые темы;
  - **O** – открытые темы;
  - **C** – закрытые темы.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом и содержит ключи, совпадающие с полями таблицы [proshop\\_msgs\\_topics](#).

## **Messages::GetTopicsByUser**

Назначение:

- Извлечение из базы данных тем сообщений, созданных заданным пользователем.

Объявление:

- `function GetTopicsByUser($user_id, $show_closed = true)`

Параметры:

- **int \$user\_id**. Идентификатор пользователя (из таблицы [proshop\\_users](#)).
- **bool \$show\_closed**. Флаг извлечения закрытых тем: **true** – извлекать все темы, включая закрытые; **false** – исключить закрытые темы.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом и содержит ключи, совпадающие с полями таблицы [proshop\\_msgs\\_topics](#).

## **Messages::SetTopicStatus**

Назначение:

- Установка статуса определенной теме сообщений.

Объявление:

- `function SetTopicStatus($topic_id, $status = 'N')`

Параметры:

- **int \$topic\_id**. Идентификатор темы сообщений из таблицы [proshop\\_msgs\\_topics](#).
- **char \$status**. Новый статус указанной темы:

- **N** – новая тема;
- **O** – открытая тема;
- **C** – закрытая тема.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Messages::SetTopicUserRead**

Назначение:

- Установка флага того, что пользователь прочитал последнее сообщение, оставленное администратором в указанной теме.

Объявление:

- `function SetTopicUserRead($topic_id, $user_read)`

Параметры:

- **int \$topic\_id**. Идентификатор темы сообщений из таблицы [proshop\\_msgs\\_topics](#).
- **char \$user\_read**. Флаг прочтения: **Y** – пользователь прочитал последнее сообщение в теме, **N** – еще не прочитал.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс MetaTags**

Предназначен для управления мета-тэгами различных страниц магазина (тэгами для поискового продвижения).

### **MetaTags::AddMeta**

Назначение:

- Добавление мета-тэга в БД.

Объявление:

- `function AddMeta($type_name, $elem_id, $meta_info)`

Параметры:

- **string \$type\_name**. Тип мета-тэга. Определяет, для какой страницы предназначен создаваемый мета-тэг. Список типов содержится в таблице [proshop\\_meta\\_types](#).
- **int \$elem\_id**. Идентификатор элемента, относящегося к указанному типу. Например, если значение параметра **\$type\_name** равно **product**, то **\$elem\_id** будет идентификатором продукта (из таблицы [proshop\\_products](#)).
- **array \$meta\_info**. Массив с информацией о мета-тэге для каждого языка, содержащегося в магазине. См. пример.

Возвращаемое значение:

- Тип **int**. Идентификатор созданного мета-тэга (в таблице [proshop\\_meta\\_tags](#)).

Пример:

- ```
$info = array(  
    // Добавление информации для языка с идентификатором 1  
    1 => array('meta_title' => $title, 'meta_keys' => $keys, 'meta_desc' => $desc),  
    // Добавление информации для языка с идентификатором 2  
    2 => ...  
);  
MetaTags::AddMeta('product', $prod_id, $info);
```

## **MetaTags::DeleteMeta**

Назначение:

- Удаление указанного мета-тэга из БД.

Объявление:

- `function DeleteMeta($type_name, $elem_id = 0)`

Параметры:

- **string \$type\_name**. Тип мета-тэга. Определяет, для какой страницы предназначен удаляемый мета-тэг. Список типов содержится в таблице [proshop\\_meta\\_types](#).
- **int \$elem\_id**. Идентификатор элемента, относящегося к указанному типу. Например, если значение параметра **\$type\_name** равно **product**, то **\$elem\_id** будет идентификатором продукта (из таблицы [proshop\\_products](#)).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **MetaTags::GetMeta**

Назначение:

- Извлечение информации об указанном мета-тэге из БД.

Объявление:

- `function GetMeta($type_name, $elem_id, $lang_id)`

Параметры:

- **string \$type\_name**. Тип мета-тэга. Определяет, для какой страницы предназначен мета-тэг, информацию о котором необходимо извлечь. Список типов содержится в таблице [proshop\\_meta\\_types](#).
- **int \$elem\_id**. Идентификатор элемента, относящегося к указанному типу. Например, если значение параметра **\$type\_name** равно **product**, то **\$elem\_id** будет идентификатором продукта (из таблицы [proshop\\_products](#)).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь информацию о мета-тэге.

Возвращаемое значение:

- Тип **array**. Массив, состоящий из следующих ключей:

- **meta\_title** – содержимое тэга <title> страницы;
- **meta\_keys** – содержимое мета-тэга <meta http-equiv="keywords" /> страницы;
- **meta\_desc** – содержимое мета-тэга <meta http-equiv="description" /> страницы.

## MetaTags::UpdateMeta

Назначение:

- Обновление информации об указанном мета-тэге.

Объявление:

- `function UpdateMeta($type_name, $elem_id, $lang_id, $meta_info)`

Параметры:

- **string \$type\_name**. Тип мета-тэга. Определяет, для какой страницы предназначен мета-тэг, информацию о котором необходимо обновить. Список типов содержится в таблице [proshop\\_meta\\_types](#).
- **int \$elem\_id**. Идентификатор элемента, относящегося к указанному типу. Например, если значение параметра **\$type\_name** равно **product**, то **\$elem\_id** будет идентификатором продукта (из таблицы [proshop\\_products](#)).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо обновить информацию о мета-тэге.
- **array \$meta\_info**. Массив с информацией о мета-тэге, содержащий следующие ключи:
  - **meta\_title** – содержимое тэга <title> страницы;
  - **meta\_keys** – содержимое мета-тэга <meta http-equiv="keywords" /> страницы;
  - **meta\_desc** – содержимое мета-тэга <meta http-equiv="description" /> страницы.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс Modules

Предназначен для управления модулями магазина (извлечение информации о них, включение/отключение).

### Modules::GetAdminActions

Назначение:

- Извлечение всех записей из таблицы [proshop\\_modules\\_admin\\_actions](#) для указанного модуля.

Объявление:

- `function GetAdminActions($module)`

Параметры:

- **string \$module**. Название модуля (из таблицы [proshop\\_modules](#)), записи которого необходимо извлечь из таблицы [proshop\\_modules\\_admin\\_actions](#).

Возвращаемое значение:



- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_modules\\_admin\\_actions](#).

## **Modules::GetAll**

Назначение:

- Извлечение модулей из БД.

Объявление:

- `function GetAll($admin_menu_id = 0, $admin_group_id = 0)`

Параметры:

- **int \$admin\_menu\_id**. Идентификатор блока меню в административной части (из таблицы [proshop\\_admin\\_menus](#)). Используется для извлечения набора модулей для конкретного блока меню админки. Если данный параметр равен 0, извлекаются все модули, независимо от принадлежности тому или иному блоку.
- **int \$admin\_group\_id**. Идентификатор группы администраторов из таблицы [proshop\\_admins\\_groups](#). Если данный параметр не равен нулю, при извлечении модулей учитываются запрещенные модули указанной группы администраторов, которые не должны фигурировать в списке извлеченных модулей.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_modules](#).

## **Modules::GetOne**

Назначение:

- Извлечение информации об указанном модуле из БД (из таблицы [proshop\\_modules](#)).

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор модуля из таблицы [proshop\\_modules](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_modules](#).

## **Modules::GetOneByName**

Назначение:

- Извлечение информации об указанном модуле из БД (из таблицы [proshop\\_modules](#)), по названию модуля.

Объявление:

- `function GetOneByName($name)`

Параметры:

- **string \$name**. Название модуля из таблицы [proshop\\_modules](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_modules](#).

## **Modules::Update**

Назначение:

- Включение/выключение указанного модуля.

Объявление:

- `function Update($id, $avail)`

Параметры:

- **int \$id**. Идентификатор модуля из таблицы [proshop\\_modules](#).
- **char \$avail**. Флаг доступности модуля (**Y** – включен, **N** - выключен).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Newest**

Предназначен для извлечения списка продуктов-новинок из БД.

### **Newest::GetAll**

Назначение:

- Извлечение списка продуктов-новинок из БД.

Объявление:

- `function GetAll($cat_id, $lang_id, $count, $added = true)`

Параметры:

- **int \$cat\_id**. Идентификатор категории, для которой необходимо извлечь продукты-новинки. Если данный параметр равен 0, новинки извлекаются для всего магазина (для любой категории).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие продуктов (название, описание и т.д.).
- **int \$count**. Максимальное количество извлекаемых новинок.
- **bool \$added**. Если значение данного параметра равно **true**, при поиске новинок учитывается дата добавления продукта (т.е. новинками будут считаться последние **\$count** добавленных товаров). Если же значение данного параметра равно **false**, учитывается дата изменения продукта (т.е. новинками будут считаться последние **\$count** измененных или добавленных товаров).

Возвращаемое значение:

- Тип **array**. См. возвращаемое значение функции [Products::Search](#).

### **Newest::GetIds**

Назначение:

- Извлечение идентификаторов продуктов-новинок.

Объявление:

- `function GetIds($cat_id, $count, $added = true)`

Параметры:

- **int \$cat\_id.** Идентификатор категории, продукты-новинки которой (включая все ее подкатегории) будут найдены в результате работы функции. Если категория не задана (параметр равен 0), будут извлечены продукты-новинки для всех категорий магазина.
- **int \$count.** Максимальное кол-во продуктов-новинок, идентификаторы которых нужно извлечь.
- **bool \$added.** Если данный параметр равен **true**, в качестве продуктов-новинок извлекаются последние добавленные продукты. Иначе – последние отредактированные продукты.

Возвращаемое значение:

- Тип **array**. Массив идентификаторов продуктов (идентификаторы из таблицы [proshop\\_products](#)).

## Класс News

Предназначен для управления новостями магазина.

### News : : Add

Назначение:

- Добавление новости в БД.

Объявление:

- `function Add($fact_info, $user_info)`

Параметры:

- **array \$fact\_info.** Массив с информацией о добавляемой новости. Должен содержать следующие ключи:
  - **date** – дата публикации новости (в формате ГГГГ-ММ-ДД);
  - **show\_in\_homepage** – если равен **Y**, анонс новости должен отображаться на главной странице витрины;
  - числовой ключ, обозначающий идентификатор языка, для которого создаются название и текст новости (см. пример ниже).
- **array \$user\_info.** Массив с информацией о пользователе, который редактировал новость. Должен содержать следующие ключи:
  - **id** – идентификатор пользователя из таблицы [proshop\\_users](#);
  - **name** – имя пользователя. Макс. 254 символа;
  - **email** – e-mail пользователя. Макс. 254 символа.

Возвращаемое значение:

- Тип **int/bool**. Идентификатор новости или **false** в случае ошибки.

Пример:

- `// ...`  
`$info = array();`  
`foreach ($LANGUAGES as $lang) {`  
    `$info[$lang['id']]['caption'] = $caption;`  
    `$info[$lang['id']]['content'] = $content;`  
    `$info[$lang['id']]['short_descr'] = $short_descr;`  
`}`  
`News::Add($info);`  
`// ...`

## **News::AddSubscriber**

Назначение:

- Добавление подписчика на новости в БД (в таблицу [proshop\\_news\\_subscribers](#)).

Объявление:

- `function AddSubscriber($email, $lang_id, $confirmed = 'N')`

Параметры:

- **string \$email**. Адрес электронной почты подписчика.
- **int \$lang\_id**. Идентификатор языка, на котором данный подписчик будет читать новости.
- **char \$confirmed**. Подтверждена ли подписка на новости данным подписчиком или нет (**Y** – подтверждена, **N** - нет).

Возвращаемое значение:

- Тип **string/bool**. `False` – в случае ошибки. Иначе – ключ подписки.

## **News::ConfirmSubscriber**

Назначение:

- Подтверждение подписки на новости по указанному ключу.

Объявление:

- `function ConfirmSubscriber($key)`

Параметры:

- **string \$key**. Ключ подписки из таблицы [proshop\\_news\\_subscribers](#).

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

## **News::Delete**

Назначение:

- Удаление указанной новости из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор удаляемой новости из таблицы [proshop\\_news](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **News::DeleteImage**

Назначение:

- Удаление превью-изображения указанной новости.

Объявление:

- function DeleteImage(\$id)

Параметры:

- **int \$id**. Идентификатор новости из таблицы [proshop\\_news](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **News::DelSubscriber**

Назначение:

- Удаление указанного адреса электронной почты из списка подписчиков на новости.

Объявление:

- function DelSubscriber(\$email)

Параметры:

- **string \$email**. Удаляемый адрес электронной почты.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **News::DelSubscriberById**

Назначение:

- Удаление указанного подписчика из таблицы [proshop\\_news\\_subscribers](#).

Объявление:

- function DelSubscriberById(\$id)

Параметры:

- **int \$id**. Идентификатор подписчика из таблицы [proshop\\_news\\_subscribers](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **News::DelSubscriberByKey**

Назначение:

- Удаление подписчика из таблицы [proshop\\_news\\_subscribers](#) по ключу подписки.

Объявление:

- function DelSubscriberByKey(\$key)

Параметры:

- **string \$key**. Ключ подписки из таблицы [proshop\\_news\\_subscribers](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **News::EditHistory**

Назначение:

- Вывод истории редактирования указанной новости (с сортированием по возрастанию даты редактирования).

Объявление:

- `function EditHistory($id)`

Параметры:

- **int \$id**. Идентификатор новости из таблицы [proshop\\_news](#).

Возвращаемое значение:

- Тип **array**. Массив, каждым элементом которого также является массив, содержащий следующие ключи:
  - **update\_date** – дата/время редактирования новости;
  - **user\_id** – идентификатор пользователя (из таблицы [proshop\\_users](#)), отредактировавшего новость, или 0, если пользователя уже не нет в БД;
  - **user\_name** – имя пользователя, отредактировавшего новость;
  - **user\_email** – e-mail пользователя, отредактировавшего новость.

## **News::GetAll**

Назначение:

- Извлечение новостей из БД.

Объявление:

- `function GetAll($lang_id, $limit = 0, $complete_sql = "", $only_captions = false, $show_future = false, $only_homepage = false)`

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие новостей (название, текст).
- **int \$limit**. Максимальное количество извлекаемых новостей (самых свежих по времени). Если данный параметр равен 0, кол-во не ограничено.
- **string \$complete\_sql**. Часть SQL-запроса, отвечающая за сортировку и пейджинг (см. класс [Pager](#)). Может быть равна пустой строке.
- **bool \$only\_captions**. Если данный параметр равен **true**, новости извлекаются только с названием (без текста новости и без текста ее анонса).

- **bool \$show\_future.** Если данный параметр равен **true**, в результирующем списке также могут присутствовать новости, дата публикации которых еще не наступила (дата публикации в будущем по сравнению с текущей датой).
- **bool \$only\_homepage.** Если данный параметр равен **true**, извлекаются только те новости, которые должны отображаться на главной странице витрины.

Возвращаемое значение:

- Тип **array**. Массив, каждым элементом которого также является массив, содержащий следующие ключи:
  - все поля из таблицы [proshop\\_news](#);
  - **caption** – название новости;
  - **content** – содержимое новости;
  - **short\_descr** – текст анонса новости.

## News::GetAllSubscribers

Назначение:

- Извлечение подписчиков на новости из таблицы [proshop\\_news\\_subscribers](#).

Объявление:

- `function GetAllSubscribers($lang_id, $email, $complete_sql = "")`

Параметры:

- **int \$lang\_id.** Идентификатор языка, для которого необходимо извлечь список подписчиков (т.е. извлечь только тех подписчиков, которые читают новости на указанном языке).
- **string \$email.** Фильтр по e-mail адресу. Может содержать часть искомого e-mail адреса.
- **string \$complete\_sql.** Часть SQL-запроса, отвечающая за сортировку и пейджинг (см. класс [Pager](#)). Может быть равна пустой строке.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_news\\_subscribers](#).

## News::GetConfirmedSubscribers

Назначение:

- Извлечение подтвержденных подписчиков на новости (из таблицы [proshop\\_news\\_subscribers](#)).

Объявление:

- `function GetConfirmedSubscribers($lang_id)`

Параметры:

- **int \$lang\_id.** Идентификатор языка, для которого необходимо извлечь список подписчиков (т.е. извлечь только тех подписчиков, которые читают новости на указанном языке).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_news\\_subscribers](#).

## **News::GetOne**

Назначение:

- Извлечение информации об указанной новости из БД.

Объявление:

- `function GetOne($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор новости из таблицы [proshop\\_news](#).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовую информацию о новости (название, содержимое).

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе - массив, содержащий следующие ключи:
  - все поля из таблицы [proshop\\_news](#);
  - **caption** – название новости;
  - **content** – содержимое новости.

## **News::GetOneSubscriber**

Назначение:

- Извлечение информации о подписке из таблицы [proshop\\_news\\_subscribers](#).

Объявление:

- `function GetOneSubscriber($email)`

Параметры:

- **string \$email**. Адрес электронной почты подписчика.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_news\\_subscribers](#).

## **News::IsSubscriberExists**

Назначение:

- Определение, существует ли уже подписчик с заданным e-mail в базе данных.

Объявление:

- `function IsSubscriberExists($email)`

Параметры:

- **string \$email**. Адрес электронной почты подписчика, который необходимо проверить на существование в таблице [proshop\\_news\\_subscribers](#).

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от результата.



## **News::SetCaption**

Назначение:

- Установка заголовка указанной новости для указанного языка.

Объявление:

- `function SetCaption($fact_id, $caption, $lang_id, $user_info)`

Параметры:

- **int \$fact\_id**. Идентификатор новости из таблицы [proshop\\_news](#).
- **string \$caption**. Заголовок новости на языке **\$lang\_id**. Макс. 65533 символа.
- **int \$lang\_id**. Идентификатор языка (хранится в таблице [proshop\\_languages](#)).
- **array \$user\_info**. Массив с информацией о пользователе, который изменил заголовок новости. См. описание одноименного параметра в функции [News::Update](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **News::SetSentFlag**

Назначение:

- Установка флага того, что указанная новость была разослана подписчикам.

Объявление:

- `function SetSentFlag($id, $sent)`

Параметры:

- **int \$id**. Идентификатор новости из таблицы [proshop\\_news](#).
- **char \$sent**. Значение флага (**Y** – новость разослана подписчикам, **N** – новость еще не рассылалась).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **News::SetShowInHomepage**

Назначение:

- Установка флага того, что указанная новость должна отображаться на главной странице витрины (поле **show\_in\_homepage** таблицы [proshop\\_news](#)).

Объявление:

- `function SetShowInHomepage($id, $show_in_homepage)`

Параметры:

- **int \$id**. Идентификатор новости из таблицы [proshop\\_news](#).
- **bool \$show\_in\_homepage**. Если значение данного параметра равно **true**, анонс указанной новости будет отображаться на главной странице витрины. В противном случае анонс на главной странице витрины отображен не будет.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## News::Update

Назначение:

- Обновление информации об указанной новости в БД.

Объявление:

- function Update(\$id, \$fact\_info, \$user\_info, \$lang\_id)

Параметры:

- **int \$id**. Идентификатор новости из таблицы [proshop\\_news](#).
- **array \$fact\_info**. Массив с информацией о новости. Должен содержать следующие ключи:
  - **date** – дата публикации новости (в формате ГГГГ-ММ-ДД);
  - **caption** – заголовок новости. Макс. 65533 символа;
  - **content** – содержимое новости. Макс. 65533 символа;
  - **short\_descr** – текст анонса новости. Макс. 230 символов;
  - **show\_in\_homepage** – если равен Y, анонс новости должен отображаться на главной странице витрины.
- **array \$user\_info**. Массив с информацией о пользователе, который редактировал новость. Должен содержать следующие ключи:
  - **id** – идентификатор пользователя из таблицы [proshop\\_users](#);
  - **name** – имя пользователя. Макс. 254 символа;
  - **email** – e-mail пользователя. Макс. 254 символа.
- **int \$lang\_id**. Идентификатор языка, для которого необходимо установить информацию.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## News::UpdateSubscriber

Назначение:

- Замена e-mail адреса подписчика.

Объявление:

- function UpdateSubscriber(\$old\_email, \$new\_email)

Параметры:

- **string \$old\_email**. E-Mail адрес, который необходимо заменить на новый **\$new\_email**.
- **string \$new\_email**. Новый E-Mail адрес.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## News::UploadImage

Назначение:

- Загрузка превью-изображения указанной новости в магазин.

Объявление:

- function UploadImage(\$fact\_id, \$img, \$width)

Параметры:

- **int \$fact\_id**. Идентификатор новости из таблицы [proshop\\_news](#).
- **file \$img**. Описание загружаемого файла изображения (из **\$\_FILES**).
- **int \$width**. Максимальная ширина загружаемого изображения в пикселях. Если ширина изображения превышает данное значение, оно будет уменьшено по ширине до **\$width** пикселей с сохранением пропорций.

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## Класс Options

Предназначен для хранения индивидуальных настроек модулей в таблице [proshop\\_options](#).

### Options::Get

Назначение:

- Извлечение значения указанной настройки из таблицы [proshop\\_options](#). Вместо данной функции значение настройки можно считать из глобальной переменной: **\$CONFIG['options'][\$name]**.

Объявление:

- function Get(\$name)

Параметры:

- **string \$name**. Имя настройки.

Возвращаемое значение:

- Тип **string**. Значение настройки.

### Options::Set

Назначение:

- Установка значения указанной настройки в таблице [proshop\\_options](#).

Объявление:

- function Set(\$name, \$value)

Параметры:

- **string \$name**. Название настройки.
- **string \$value**. Значение настройки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс Orders

Предназначен для управления заказами.

## Orders : :Add

Назначение:

- Добавление заказа в БД.

Объявление:

- `function Add($cart, $customer)`

Параметры:

- **array \$cart**. Массив с информацией об элементе корзины (`$_SESSION['cart']`), который был сформирован вызовами методов класса [Cart](#). Должен содержать следующие ключи:
  - **order\_id** – идентификатор заказа (из таблицы [proshop\\_orders](#)). Значение данного ключа может быть нулевым, если до вызова данного метода не было ни одной попытки создать заказ. Если значение данного ключа не нулевое, данные о заказе будут обновлены в БД;
  - **payment\_id** – идентификатор способа платежа, выбранного покупателем (из таблицы [proshop\\_payments](#));
  - **shipping\_id** – см. возвращаемое значение метода [Cart::GetShipping](#);
  - **comment** – комментарий к заказу, оставленный покупателем;
  - **products** – см. возвращаемое значение метода [Cart::GetProds](#).
- **array \$customer**. Массив информации о текущем покупателе (`$_SESSION['customer_info']`), ключи которого должны совпадать с полями таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **int/bool**. `False` – в случае ошибки. Иначе – идентификатор созданного заказа (или идентификатор `$cart['order_id']`, если он не нулевой).

## Orders : :AddOrderedProd

Назначение:

- Добавление информации о заказанном продукте в БД (привязка к заказу).

Объявление:

- `function AddOrderedProd($order_id, $info)`

Параметры:

- **int \$order\_id**. Идентификатор заказа (из таблицы [proshop\\_orders](#)), к которому следует привязать заказанный продукт.
- **array \$info**. Массив с информацией о заказанном продукте. Должен содержать следующие ключи:
  - **product\_id** – идентификатор продукта из таблицы [proshop\\_products](#);
  - **variant\_id** – идентификатор варианта продукта (если таковой имеет место быть) из таблицы [proshop\\_products\\_variants](#);
  - **product\_cat** – название категории продукта на текущем языке магазина. Макс. 254 символа;

- **cat\_id** – идентификатор категории (из таблицы [proshop\\_categories](#)), из которой продукт был добавлен в корзину, либо идентификатор главной категории продукта (поле **cat** таблицы [proshop\\_products](#));
- **product\_caption** – наименование продукта на текущем языке магазина. Макс. 254 символа;
- **product\_sku** – артикул продукта. Макс. 254 символа;
- **product\_options** – опции продукта (например, размер обуви или цвет футболки), если таковые имеются. Макс. 65533 символа.
- **product\_qty** – количество экземпляров заказанного продукта;
- **product\_price** – цена одного экземпляра продукта в базовой валюте магазина;
- **product\_weight** – вес одного экземпляра продукта в кг;
- **product\_egood** – является ли продукт скачиваемым (**Y** – да, **N** - нет).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::CancelOrderedProd**

Назначение:

- Помечивание заказанного продукта (из таблицы [proshop\\_orders\\_products](#)) в качестве отмененного – используется при отмене заказа определенного продукта.

Объявление:

- function CancelOrderedProd(\$id)

Параметры:

- **int \$id**. Идентификатор заказанного продукта из таблицы [proshop\\_orders\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::Delete**

Назначение:

- Удаление указанного заказа из БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Orders::DelOrderedProd**

Назначение:

- Удаление заказанного продукта из состава заказа.

Объявление:

- function DelOrderedProd(\$id)

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_orders\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Orders::GetAll

Назначение:

- Извлечение заказов из БД.

Объявление:

- function GetAll(\$status\_id = 0, \$customer\_id = 0, \$only\_ids = false, \$pager\_ending = "")

Параметры:

- **int \$status\_id**. Идентификатор статуса из таблицы [proshop\\_orders\\_statuses](#), заказы с которым необходимо извлечь. Если нужно извлечь заказы с любым статусом, значение данного параметра должно быть равно 0.
- **int \$customer\_id**. Идентификатор покупателя из таблицы [proshop\\_users](#), заказы которого необходимо извлечь. Если нужно извлечь заказы любого покупателя, значение данного параметра должно быть равно 0.
- **bool \$only\_ids**. Если значение данного параметра равно true, извлекаются только идентификаторы заказов. Иначе – все поля таблицы [proshop\\_orders](#).
- **string \$pager\_ending**. Часть SQL-запроса, которую можно «приклеить» к концу основного SQL-запроса (используется пейджингом – см. класс [Pager](#)).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом с ключами, совпадающими с полями таблицы [proshop\\_orders](#). Если параметр **\$only\_ids** имеет значение true, при извлечении учитывается только поле **id** таблицы [proshop\\_orders](#).

## Orders::GetAllByCustomer

Назначение:

- Извлечение всех заказов из БД для заданного покупателя.

Объявление:

- function GetAllByCustomer(\$customer\_id)

Параметры:

- **int \$customer\_id**. Идентификатор покупателя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом с ключами, совпадающими с полями таблицы [proshop\\_orders](#).

## Orders::GetCount

Назначение:

- Определение кол-ва заказов, имеющих указанный статус, и/или связанных с заданным покупателем.

Объявление:

- `function GetCount($status_id, $customer_id)`

Параметры:

- **int \$status\_id**. Идентификатор статуса заказа из таблицы [proshop\\_orders\\_statuses](#). Может быть равен 0 для любого статуса.
- **Int \$customer\_id**. Идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#). Может быть равен 0 для любого покупателя.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – кол-во заказов.

## Orders::GetCustomers

Назначение:

- Извлечение списка всех зарегистрированных покупателей (имеющихся в таблице [proshop\\_users](#)) из таблицы заказов [proshop\\_orders](#).

Объявление:

- `function GetCustomers()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив с информацией о покупателях, каждый элемент которого является массивом, содержащим ключи:
  - **customer\_id** – идентификатор покупателя из таблицы [proshop\\_users](#);
  - **customer\_first\_name** – имя покупателя;
  - **customer\_last\_name** – фамилия покупателя;
  - **customer\_second\_name** – отчество покупателя;
  - **customer\_company** – наименование организации.

## Orders::GetOne

Назначение:

- Извлечение информации об указанном заказе из БД.

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе - массив, ключи которого совпадают с полями таблицы [proshop\\_orders](#).

## **Orders::GetOrderedProd**

Назначение:

- Извлечение информации о заказанном продукте из таблицы [proshop\\_orders\\_products](#).

Объявление:

- function GetOrderedProd(\$id)

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_orders\\_products](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_orders\\_products](#).

## **Orders::GetOrderedProds**

Назначение:

- Извлечение информации о заказанных продуктах из таблицы [proshop\\_orders\\_products](#) для указанного заказа.

Объявление:

- function GetOrderedProds(\$order\_id, \$show\_cancelled = false)

Параметры:

- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **bool \$show\_cancelled**. Учитывать ли при извлечении отмененные продукты (**Y** – учитывать, **N** – не учитывать).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_orders\\_products](#).

## **Orders::GetPendingProds**

Назначение:

- Извлечение из заказов всех товаров, которые заказаны с отложенной отгрузкой.

Объявление:

- function GetPendingProds(\$statuses\_ids)

Параметры:

- **array \$statuses\_ids**. Массив идентификаторов статусов заказов (идентификаторы статусов заказов хранятся в таблице [proshop\\_orders\\_statuses](#)), с которыми необходимо просмотреть все заказы на наличие товаров с отложенной отгрузкой.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:



- **id** – уникальный идентификатор записи в таблице [proshop\\_orders\\_products](#);
- **product\_id** – идентификатор продукта из таблицы [proshop\\_products](#);
- **variant\_id** – идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#). Может быть равен 0, если продукт не имеет вариантов;
- **product\_cat** – название категории, из которой продукт был добавлен в корзину покупателя перед оформлением заказа;
- **cat\_id** – идентификатор категории (из таблицы [proshop\\_categories](#)), из которой продукт был добавлен в корзину покупателя перед оформлением заказа;
- **product\_caption** – название продукта;
- **product\_sku** – артикул продукта. Может быть пустым;
- **product\_qty** – кол-во заказанных экземпляров продукта;
- **product\_options** – опции продукта в текстовом виде. Может быть пустой строкой, если продукт не имеет опций;
- **order\_id** – номер заказа;
- **status\_id** – идентификатор текущего статуса заказа под номером **order\_id**;
- **customer\_first\_name** – имя покупателя. Поле может быть пустым;
- **customer\_last\_name** – фамилия покупателя. Поле может быть пустым;
- **customer\_second\_name** – отчество покупателя. Поле может быть пустым;
- **customer\_company** – название организации покупателя. Поле может быть пустым;
- **customer\_ogrn** – ОГРН организации покупателя. Поле может быть пустым;
- **customer\_inn** – ИНН организации покупателя. Поле может быть пустым;
- **customer\_kpp** – КПП организации покупателя. Поле может быть пустым;
- **customer\_contact\_person** – контактное лицо организации покупателя. Поле может быть пустым;
- **customer\_legal\_address** – юридический адрес организации покупателя. Поле может быть пустым;
- **customer\_phone** – номер(а) телефона(ов) покупателя. Поле может быть пустым;
- **customer\_mobile\_phone** – номер мобильного телефона покупателя. Поле может быть пустым;
- **customer\_fax** – номер факса покупателя. Поле может быть пустым;
- **customer\_address** – адрес покупателя. Поле может быть пустым;
- **customer\_ship\_address** – отдельный адрес доставки. Поле может быть пустым;
- **customer\_zip\_code** – почтовый индекс адреса покупателя. Поле может быть пустым;
- **customer\_ship\_zip\_code** – отдельный почтовый индекс адреса доставки. Поле может быть пустым;
- **customer\_city** – город покупателя. Поле может быть пустым;
- **customer\_ship\_city** – отдельный город адреса доставки. Поле может быть пустым;
- **customer\_country** – страна покупателя. Поле может быть пустым;
- **customer\_ship\_country** – отдельная страна адреса доставки. Поле может быть пустым;

- **customer\_state** – область страны покупателя. Поле может быть пустым;
- **customer\_ship\_state** – отдельная область страны адреса доставки. Поле может быть пустым;
- **customer\_email** – e-mail покупателя;
- **supplier\_notified** – флаг того, что поставщик был уведомлен о заказе данного товара (**Y** – уведомлен, **N** – еще не уведомлен);
- **pqty** – текущее кол-во на складе продукта **product\_id**;
- **vqty** – текущее кол-во на складе варианта продукта **variant\_id**.

## Orders::GetStatuses

Назначение:

- Извлечение информации о возможных статусах заказов из таблицы [proshop\\_orders\\_statuses](#).

Объявление:

- `function GetStatuses($lang_id, $admin_may_set = false)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором нужно извлечь названия статусов.
- **bool \$admin\_may\_set**. Если значение данного параметра равно `true`, извлекаются только те статусы, которые администратор может выставлять вручную при редактировании заказа. Если значение данного параметра равно `false`, извлекаются все статусы.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_orders\\_statuses](#), также содержащим ключ **caption** (название статуса). Если название статуса не задано на указанном языке, вместо него возвращается служебное название статуса.

Пример:

- ```
$statuses = Orders::GetStatuses();
foreach ($statuses as $status_id => $status) {
    // $status_id – идентификатор статуса
    // $status['caption'] – название статуса
    // ...
}
```

## Orders::GetStatusesHistory

Назначение:

- Извлечение истории статусов указанного заказа из таблицы [proshop\\_orders\\_statuses\\_were](#) в хронологическом порядке установки этих статусов.

Объявление:

- `function GetStatusesHistory($order_id)`

Параметры:

- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является ассоциативным массивом, содержащим следующие ключи:
  - **status\_id** – идентификатор статуса из таблицы [proshop\\_orders\\_statuses](#);
  - **timestamp** – дата и время установки статуса.

## **Orders::GetStatusId**

Назначение:

- Определение статуса указанного заказа.

Объявление:

- `function GetStatusId($id)`

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **int**. Идентификатор статуса (из таблицы [proshop\\_orders\\_statuses](#)) указанного заказа.

## **Orders::GetTotalCount**

Назначение:

- Определение общего кол-ва заказов, содержащихся в БД.

Объявление:

- `function GetTotalCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. **False** – в случае ошибки. Иначе – кол-во заказов в таблице [proshop\\_orders](#).

## **Orders::MayCancelStatus**

Назначение:

- Определение, можно ли отменить заказ при его заданном статусе.

Объявление:

- `function MayCancelStatus($status_id)`

Параметры:

- **int \$status\_id**. Идентификатор статуса заказа из таблицы [proshop\\_orders\\_statuses](#).

Возвращаемое значение:

- Тип **bool**. **True**, если любой заказ с заданным статусом может быть отменен. Иначе – **false**.

## **Orders::MayChangedStatus**

Назначение:

- Определение, можно ли менять статус заказа при его заданном статусе.

Объявление:

- `function MayChangedStatus($status_id)`

Параметры:

- **int \$status\_id**. Идентификатор статуса заказа из таблицы [proshop\\_orders\\_statuses](#).

Возвращаемое значение:

- Тип **bool**. True, если у любого заказа с заданным статусом можно сменить статус. Иначе – false.

## **Orders::ReadyForShipment**

Назначение:

- Определение, полностью ли готов указанный заказ к отгрузке со склада.

Объявление:

- `function ReadyForShipment($id)`

Параметры:

- **int \$id**. Номер заказа (идентификатор из таблицы [proshop\\_orders](#)).

Возвращаемое значение:

- Тип **bool**. True, если заказ полностью готов к отгрузке. Иначе – false.

## **Orders::SetChangeEmailed**

Назначение:

- Установка флага отправки уведомления покупателю о том, что состав его заказа был изменен администратором. Применяется после отправки уведомления покупателю по электронной почте.

Объявление:

- `function SetChangeEmailed($id)`

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#), состав которого изменился.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetComment**

Назначение:

- Установка комментария покупателя к заказу.

Объявление:

- `function SetComment($order_id, $comment)`

Параметры:

- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **string \$comment**. Текст комментария на текущем языке магазина. Макс. 65533 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetDiscountPer**

Назначение:

- Установка величины процента скидки на указанный заказ (установка значения поля **discount\_per** таблицы [proshop\\_orders](#)).

Объявление:

- function SetDiscountPer(\$id, \$per)

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **float \$per**. Величина процента скидки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetEmailed**

Назначение:

- Установка флага отправки уведомления покупателю о том, что его заказ был принят к рассмотрению. Применяется после отправки уведомления покупателю по электронной почте.

Объявление:

- function SetEmailed(\$id)

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetProdPending**

Назначение:

- Установка значения флага **pending** для определенного заказанного продукта в таблице [proshop\\_orders\\_products](#).

Объявление:

- function SetProdPending(\$id, \$pending)

Параметры:

- **int \$id**. Идентификатор записи в таблице [proshop\\_orders\\_products](#).
- **bool \$pending**. Значение флага **pending**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetProdQty**

Назначение:

- Установка значения поля **product\_qty** для определенного заказанного продукта в таблице [proshop\\_orders\\_products](#).

Объявление:

- function SetProdQty(\$id, \$qty)

Параметры:

- **int \$id**. Идентификатор записи в таблице [proshop\\_orders\\_products](#).
- **int \$qty**. Значение поля **product\_qty**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetProdSupplierNotified**

Назначение:

- Установка значения поля **supplier\_notified** для указанной записи в таблице [proshop\\_orders\\_products](#).

Объявление:

- function SetProdSupplierNotified(\$id, \$notified)

Параметры:

- **int \$id**. Идентификатор записи в таблице [proshop\\_orders\\_products](#).
- **bool \$notified**. Значение поля **supplier\_notified**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetShippingInfo**

Назначение:

- Установка информации о способе доставки в заказе.

Объявление:

- function SetShippingInfo(\$id, \$info)

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **array \$info**. Массив с информацией о способе доставки. Должен содержать следующие ключи:
  - **method** – название способа доставки. Макс. 254 символа;
  - **cost** – стоимость доставки в базовой валюте магазина.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetStatus**

Назначение:

- Установка статуса указанного заказа.

Объявление:

- `function SetStatus($order_id, $status_id)`

Параметры:

- **int \$order\_id**. Идентификатор заказа (из таблицы [proshop\\_orders](#)), статус которого необходимо установить.
- **int \$status\_id**. Идентификатор статуса из таблицы [proshop\\_orders\\_statuses](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::SetStatusCaption**

Назначение:

- Установка названия указанного статуса на указанном языке (запись в таблицу [proshop\\_orders\\_statuses\\_texts](#)).

Объявление:

- `function SetStatusCaption($id, $lang_id, $caption)`

Параметры:

- **int \$id**. Идентификатор статуса из таблицы [proshop\\_orders\\_statuses](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **string \$caption**. Название статуса на языке **\$lang\_id**. Макс. 60 символов.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Orders::StatusIdToString**

Назначение:

- Конвертирование идентификатора статуса в название статуса.

Объявление:

- `function StatusIdToString($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор статуса из таблицы [proshop\\_orders\\_statuses](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором необходимо извлечь название статуса.

Возвращаемое значение:

- Тип **string**. Название статуса на языке **\$lang\_id**. Если для данного языка название статуса отсутствует, вместо него будет возвращено служебное название статуса.

## Orders :: StatusWas

Назначение:

- Определение, устанавливался ли уже указанный статус указанного заказа ранее.

Объявление:

- `function StatusWas($order_id, $status_id)`

Параметры:

- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **int \$status\_id**. Идентификатор статуса из таблицы [proshop\\_orders\\_statuses](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## Orders :: UpdateCustomerInfo

Назначение:

- Установка информации о покупателе для указанного заказа.

Объявление:

- `function UpdateCustomerInfo($id, $info)`

Параметры:

- **int \$id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **array \$info**. Массив, ключи которого должны совпадать с полями таблицы [proshop\\_orders](#), начинающимися на **customer\_** (кроме **customer\_id**).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Orders :: UpdateOrderedProd

Назначение:

- Обновление информации о заказанном продукте в таблице [proshop\\_orders\\_products](#).

Объявление:

- `function UpdateOrderedProd($id, $info)`

Параметры:

- **int \$id**. Идентификатор заказанного продукта из таблицы [proshop\\_orders\\_products](#).
- **array \$info**. Массив с информацией, содержащий следующие ключи:
  - **price** – цена одного экземпляра продукта в базовой валюте магазина;
  - **weight** – вес одного экземпляра продукта в кг;
  - **qty** – количество заказанных экземпляров продукта;
  - **tax\_percent** – процент налога, который входит в цену одного экземпляра.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.



## Orders::UpdateSum

Назначение:

- Пересчет суммы указанного заказа (и сохранение значения в таблице [proshop\\_orders](#)).

Объявление:

- `function UpdateSum($order_id)`

Параметры:

- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **float/bool**. False – в случае возникновения ошибки. Иначе – сумма заказа (включая налоги) без учета стоимости доставки в базовой валюте магазина.

## Orders::UpdateWeight

Назначение:

- Пересчет суммарного веса указанного заказа (и сохранение его значения в таблице [proshop\\_orders](#)).

Объявление:

- `function UpdateWeight($order_id)`

Параметры:

- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).

Возвращаемое значение:

- Тип **float/bool**. False – в случае возникновения ошибки. Иначе – суммарный вес заказа в кг.

## Класс Pager

Предназначен для организации многостраничной навигации (например, при выводе большого количества продуктов) с возможностью сортировки выводимых списков по заданному полю. Используется некоторыми модулями (**products**, **news**, и т.д.).

### `$pager->getCurrentPage`

Назначение:

- Определение текущей страницы, на которой находится пользователь.

Объявление:

- `function getCurrentPage()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Номер текущей страницы, на которой находится пользователь.

## **`$pager->getNextPage`**

Назначение:

- Определение номера следующей страницы в постраничной навигации.

Объявление:

- `function getNextPage()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Номер следующей страницы.

## **`$pager->getOrderField`**

Назначение:

- Определение имени сортируемого поля по текущему направлению сортировки.

Объявление:

- `function getOrderField($order_field)`

Параметры:

- **string \$order\_field**. Имя сортируемого поля. Если текущее направление сортировки – по возрастанию, данное имя не меняется и возвращается как результат функции. Если же текущее направление сортировки – по убыванию, в конец данного имени прибавляется постфикс **\_re**, означающий, что список элементов в пейджинге нужно вывести по убыванию данного поля.

Возвращаемое значение:

- Тип **string**. Имя сортируемого поля с (или без) префиксом сортировки по убыванию.

## **`$pager->getOrderHref`**

Назначение:

- Формирование ссылки для сортировки элементов списка пейджинга в том или ином порядке (по возрастанию или убыванию) по заданному полю. Применяется в HTML-шаблонах админки.

Объявление:

- `function getOrderHref($href, $order_field, $force_desc = false)`

Параметры:

- **string \$href**. Ссылка, которую необходимо переработать (изменить направление и поле сортировки).
- **string \$order\_field**. Поле, по которому производится сортировка. Может иметь постфикс **\_re** (для сортировки по убыванию).
- **bool \$force\_desc**. Если равен **true**, то включается принудительная сортировка по убыванию.

Возвращаемое значение:

- Тип **string**. Модифицированная ссылка \$href.

## **`$pager->getPagesCount`**

Назначение:

- Определение кол-ва страниц в постраничной навигации.

Объявление:

- `function getPagesCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во страниц.

## **`$pager->getPageSize`**

Назначение:

- Определение максимально возможного кол-ва элементов на одной странице при постраничной навигации.

Объявление:

- `function getPageSize()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Максимально возможное кол-во элементов на одной странице в пейджинге.

## **`$pager->getPaging`**

Назначение:

- Формирование постраничной навигации.

Объявление:

- `function getPaging($href = "")`

Параметры:

- **string \$href**. Адрес текущей страницы в админке. Используется только в админке. Для использования на витрине должен быть пустым.

Возвращаемое значение:

- Тип **array**. Массив с информацией о пейджинге, включающий следующие ключи:
  - **prev\_page** – массив с информацией о предыдущей странице по отношению к текущей. Содержит следующие ключи:
    - **num** – номер предыдущей страницы;
    - **href** – адрес предыдущей страницы (только в админке);
  - **next\_page** – массив с информацией о следующей странице по отношению к текущей. Содержит следующие ключи:
    - **num** – номер следующей страницы;
    - **href** – адрес следующей страницы (только в админке);

- **pages** – массив с информацией о каждой странице пейджинга. Каждый элемент данного массива содержит следующие ключи:
  - **num** – номер страницы;
  - **href** – адрес страницы (только в админке);
  - **dots** – true, если элемент массива не является описанием страницы, а представляет собой многоточие (для разделения групп номеров страниц, когда не все номера страниц возможно вывести).

## **\$pager->getPrevPage**

Назначение:

- Определение номера предыдущей страницы в постраничной навигации.

Объявление:

- function getPrevPage()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Номер предыдущей страницы.

## **\$pager->getSortDir**

Назначение:

- Определение текущего направления сортировки для заданного поля.

Объявление:

- function getSortDir(\$field\_name)

Параметры:

- **string \$field\_name**. Название поля, для которого необходимо определить текущее направление сортировки.

Возвращаемое значение:

- Тип **int**. Возвращается 1, если для заданного поля включена сортировка по возрастанию. Если -1, то включена сортировка по убыванию. Если возвращается 0, то для заданного поля сортировка не включена.

## **\$pager->getSQL**

Назначение:

- Формирование части SQL-запроса, отвечающей за сортировку и вывод части результата запроса SELECT.

Объявление:

- function getSQL(\$enable\_order = true)

Параметры:

- **bool \$enable\_order**. Включение/выключение сортировки в SQL-запросе (оператора ORDER BY). По умолчанию сортировка включена.

Возвращаемое значение:

- Тип **string**. Часть SQL-запроса с операторами ORDER BY и LIMIT.

## **\$pager->Pager**

Назначение:

- Конструктор класса. Подготавливает внутренние переменные класса для работы.

Объявление:

- function Pager()

Параметры:

- Функция не принимает параметров.

## **\$pager->setDefaultSort**

Назначение:

- Установка поля с направлением сортировки, которое используется по умолчанию, если сортировка не задана.

Объявление:

- function setDefaultSort(\$s)

Параметры:

- **string \$s**. Поле с направлением сортировки. Например: **id DESC**. При этом поле **id** должно присутствовать в массиве, установленном с помощью [\\$pager->setSortings](#). Данный параметр может также иметь значение **RAND**, что означает сортировку по любому полю в случайном порядке.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$pager->setPage**

Назначение:

- Установка номера текущей страницы, на которую перешел пользователь в постраничной навигации.

Объявление:

- function setPage(\$i)

Параметры:

- **int \$i**. Номер текущей страницы в пейджинге.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$pager->setPageSize`**

Назначение:

- Установка максимально возможного кол-ва элементов на одной странице пейджинга.

Объявление:

- `function setPageSize($i)`

Параметры:

- **int \$i**. Максимально возможное кол-во элементов. Если данный параметр равен 0, функция [\\$pager->getSQL](#) не будет включать LIMIT в SQL-запрос.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$pager->setRecordCount`**

Назначение:

- Установка общего кол-ва записей, отображаемых в пейджинге, для того, чтобы правильно определить кол-во страниц и составить постраничную навигацию.

Объявление:

- `function setRecordCount($i)`

Параметры:

- **int \$i**. Общее кол-во элементов в пейджинге.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$pager->setSort`**

Назначение:

- Установка поля сортировки для текущего вывода списка.

Объявление:

- `function setSort($s)`

Параметры:

- **string \$s**. Название поля, которое может иметь постфикс `_re`. Например: `caption_re`. При этом поле `caption` должно присутствовать в массиве, установленном с помощью [\\$pager->setSortings](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$pager->setSortings`**

Назначение:

- Установка возможных полей сортировки в пейджинге.

Объявление:

- `function setSortings($s)`

Параметры:

- **array \$s**. Массив, каждый из ключей которого должен соответствовать имени сортируемого поля. Например: `$pager->setSortings(Array('id' => 'a.id', 'caption' => 'at.caption'))`; При этом значениями данных ключей являются соответствующие поля таблиц БД. Для приведенного примера полями являются `id` из таблицы `proshop_articles` и `caption` таблицы `proshop_articles_texts`. В качестве значения может быть также массив с перечислением сортируемых полей таблиц. Например: `$pager->setSortings(Array('email' => array('u.email', 'u.id'), 'first_name' => 'u.first_name'))`; При этом, если будет включена сортировка по полю `email` по возрастанию, результат будет выведен отсортированным по возрастанию полей `u.email` и `u.id`, что соответствует **ORDER BY u.email ASC, u.id ASC**. Если по убыванию – это соответствует **ORDER BY u.email DESC, u.id DESC**.

Возвращаемое значение:

- Функция ничего не возвращает.

## `$pager->setViewPages`

Назначение:

- Установка максимально возможного количества страниц в пейджинге. Если страниц окажется больше, чем установленное значение, будут выведены группы страниц, между которыми будут размещены многоточия. Например: 1, 2, 3 ... 8 ... 13, 14, 15. Этот пример соответствует установленному значению, равному 7.

Объявление:

- `function setViewPages($view_pages)`

Параметры:

- **int \$view\_pages**. Максимально возможное кол-во страниц в пейджинге.

Возвращаемое значение:

- Функция ничего не возвращает.

## Класс `Payments`

Предназначен для управления платежными системами и методами.

### `Payments::AddManual`

Назначение:

- Добавление базовой информации о ручном способе приема платежа в таблицу [proshop\\_payments](#).

Объявление:

- `function AddManual()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданного платежного метода из таблицы [proshop\\_payments](#).

## **Payments::DelManual**

Назначение:

- Удаление ручного способа приема платежа.

Объявление:

- function DelManual(\$id)

Параметры:

- **int \$id**. Идентификатор удаляемого способа оплаты из таблицы [proshop\\_payments](#).

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## **Payments::GetAll**

Назначение:

- Извлечение всех платежных методов из БД в порядке сортировки, определенном полем **pos** таблицы [proshop\\_payments](#).

Объявление:

- function GetAll(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие платежного метода (название, описание, ...).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_payments](#);
  - **caption** – название платежного метода на указанном языке;
  - **descr** – описание платежного метода на указанном языке;
  - **laststep** – текст на последнем шаге заказа на указанном языке.

## **Payments::GetAllActive**

Назначение:

- Извлечение всех включенных (доступных для покупателя) платежных методов из БД в порядке сортировки, определенном полем **pos** таблицы [proshop\\_payments](#).

Объявление:

- function GetAllActive(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие платежного метода (название, описание, ...).



Возвращаемое значение:

- Тип **array**. См. возвращаемое значение функции [Payments::GetAll](#).

## **Payments::GetAllSettings**

Назначение:

- Извлечение всех настроек (из таблицы [proshop\\_payments\\_settings](#)) заданной автоматической платежной системы.

Объявление:

- `function GetAllSettings($id)`

Параметры:

- **int \$id**. Идентификатор платежной системы из таблицы [proshop\\_payments](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом, содержащим ключи, совпадающие с полями таблицы [proshop\\_payments\\_settings](#).

## **Payments::GetCaption**

Назначение:

- Определение названия указанного платежного метода на указанном языке.

Объявление:

- `function GetCaption($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)).

Возвращаемое значение:

- Тип **string**. Название платежного метода на указанном языке, или его служебное имя, если название для указанного языка не задано.

## **Payments::GetOne**

Назначение:

- Извлечение информации об указанном способе оплаты из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор способа оплаты из таблицы [proshop\\_payments](#).
- **int \$lang\_id**. Идентификатор языка, для которого необходимо извлечь текстовые составляющие информации о способе оплаты (название, описание, ...). Если значение данного параметра равно 0, текстовые составляющие не извлекаются.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, содержащий следующие ключи:

- все поля таблицы [proshop\\_payments](#);
- **caption** – название платежного метода на указанном языке;
- **descr** – описание платежного метода на указанном языке;
- **laststep** – текст на последнем шаге заказа на указанном языке.

## **Payments::GetSettingValue**

Назначение:

- Извлечение значения указанной настройки заданной платежной системы (с возможностью автоматического приема платежа).

Объявление:

- `function GetSettingValue($payment_name, $name)`

Параметры:

- **string \$payment\_name**. Уникальное служебное имя платежной системы из таблицы [proshop\\_payments](#).
- **string \$name**. Имя настройки, значение которой необходимо извлечь.

Возвращаемое значение:

- Тип **string**. Значение указанной настройки.

## **Payments::IsNameExists**

Назначение:

- Определение, существует ли заданное служебное имя платежной системы в таблице [proshop\\_payments](#).

Объявление:

- `function IsNameExists($name)`

Параметры:

- **string \$name**. Служебное имя платежной системы, которое необходимо проверить.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SaveTransactionData**

Назначение:

- Сохранение данных о транзакции автоматической платежной системы в таблицу [proshop\\_payments\\_transactions](#). Должна вызываться при поступлении уведомления об оплате от платежной системы.

Объявление:

- `function SaveTransactionData($payment_name, $data)`

Параметры:

- **string \$payment\_name**. Уникальное служебное имя платежной системы из таблицы [proshop\\_payments](#).

- **string \$data**. Данные о транзакции. Макс. 65533 символа.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Payments::SetActive**

Назначение:

- Установка флага активности указанного платежного метода на витрине (включение/выключение платежного метода).

Объявление:

- `function SetActive($id, $is_active)`

Параметры:

- **int \$id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **char \$is\_active**. Флаг активности платежного метода (**Y** – активен, **N** – не активен).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SetAuto**

Назначение:

- Включение/выключение автоматического режима указанной автоматической платежной системы. При выключении автоматического режима платежная система функционирует как ручной метод платежа.

Объявление:

- `function SetAuto($id, $is_auto)`

Параметры:

- **int \$id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **char \$is\_auto**. Флаг включенности автоматического режима (**Y** – включен, **N** – выключен).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SetCaption**

Назначение:

- Установка названия для указанного платежного метода на указанном языке.

Объявление:

- `function SetCaption($id, $caption, $lang_id)`

Параметры:

- **int \$id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **string \$caption**. Название платежного метода. Макс. 254 символа.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого устанавливается название **\$caption**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SetDescr**

Назначение:

- Установка описания для указанного платежного метода на указанном языке.

Объявление:

- `function SetDescr($id, $descr, $lang_id)`

Параметры:

- **int \$id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **string \$descr**. Описание платежного метода. Макс. 65533 символа.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого устанавливается описание **\$descr**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SetLastStepText**

Назначение:

- Установка текста для последней страницы заказа для указанного платежного метода на указанном языке. Например, это может быть текст с реквизитами для оплаты.

Объявление:

- `function SetLastStepText($id, $laststep, $lstep_email_only, $lang_id)`

Параметры:

- **int \$id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **string \$laststep**. Текст на последней странице заказа для платежного метода с идентификатором **\$id**. Макс. 65533 символа. Может содержать HTML.
- **char \$lstep\_email\_only**. Если значение данного параметра равно **Y**, задаваемый текст не будет отображаться на последней странице заказа (вместо этого он будет отправлен покупателю по e-mail).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого устанавливается текст **\$laststep**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SetPos**

Назначение:

- Установка позиции вывода заданной платежной системы относительно других платежных систем (для сортировки вывода).

Объявление:

- function SetPos(\$id, \$pos)

Параметры:

- **int \$id**. Идентификатор платежной системы из таблицы [proshop\\_payments](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Payments::SetSettingValue**

Назначение:

- Установка значения указанной настройки для заданного платежного метода.

Объявление:

- function SetSettingValue(\$payment\_id, \$name, \$value)

Параметры:

- **int \$payment\_id**. Идентификатор платежного метода из таблицы [proshop\\_payments](#).
- **string \$name**. Имя настройки.
- **string \$value**. Значение настройки. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс ProdDetails**

Предназначен для управления параметрами продуктов.

### **ProdDetails::AddDetailRoot**

Назначение:

- Добавление базовой информации о детали (парамetre) указанного продукта.

Объявление:

- function AddDetailRoot(\$prod\_id, \$param\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$param\_id**. Идентификатор параметра категории продукта из таблицы [proshop\\_categories\\_params](#).

Возвращаемое значение:

- Тип **int**. Идентификатор созданного параметра продукта (из таблицы [proshop\\_products\\_details](#)), или 0 – в случае ошибки.

### **ProdDetails::AddDetailValue**

Назначение:

- Добавление значения для указанного параметра продукта в таблицу [proshop\\_products\\_details\\_texts](#).

Объявление:

- `function AddDetailValue($detail_id, $value, $lang_id = 0)`

Параметры:

- **int \$detail\_id**. Идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#).
- **int/string \$value**. Значение параметра продукта. Если параметр **\$lang\_id** равен 0, то значение считается числовым. Если иначе – строковым.
- **int \$lang\_id**. Идентификатор языка, для которого добавляется строковое значение. Для каждого языка может быть свое значение. Если данный параметр равен 0, значение **\$value** считается числовым.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdDetails::AddParamRoot**

Назначение:

- Добавление базовой информации о параметре категории в таблицу [proshop\\_categories\\_params](#).

Объявление:

- `function AddParamRoot($cat_id, $is_numeric, $for_filter)`

Параметры:

- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).
- **char \$is\_numeric**. Если значение данного параметра равно **Y**, параметр категории считается числовым. Иначе – строковым.
- **bool \$for\_filter**. Если значение данного параметра равно **true**, добавляемый параметр категории будет использоваться в фильтре на витрине. Иначе – не будет.

Возвращаемое значение:

- Тип **int**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#), или 0 – в случае ошибки.

## **ProdDetails::AddParamText**

Назначение:

- Добавление названия для указанного параметра категории на указанном языке в таблицу [proshop\\_categories\\_params\\_texts](#).

Объявление:

- `function AddParamText($id, $lang_id, $caption)`

Параметры:

- **int \$id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого добавляется название указанного параметра категории.
- **string \$caption**. Название указанного параметра на языке с идентификатором **\$lang\_id**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdDetails::DelDetail**

Назначение:

- Удаление указанного параметра продукта из БД.

Объявление:

- `function DelDetail($id)`

Параметры:

- **int \$id**. Идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **ProdDetails::DelDetailsByParamID**

Назначение:

- Удаление всех деталей указанного продукта, основанных на указанном параметре категории.

Объявление:

- `function DelDetailsByParamID($prod_id, $param_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$param\_id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdDetails::DelParam**

Назначение:

- Удаление указанного параметра категории из БД.

Объявление:

- `function DelParam($id)`

Параметры:

- **int \$id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **ProdDetails::GetAllDetails**

Назначение:

- Извлечение всех параметров указанного продукта из БД.

Объявление:

- `function GetAllDetails($prod_id, $lang_id = 0)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия и значения параметров продукта. Если данный параметр равен 0, извлекаются только поля из таблицы [proshop\\_products\\_details](#).

Возвращаемое значение:

- Тип **array**. Массив параметров продукта, каждым элементом которого является массив, состоящий из следующих ключей:
  - **id** – идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#);
  - **param\_id** – идентификатор параметра категории продукта из таблицы [proshop\\_categories\\_params](#);
  - **caption** – название параметра (если **\$lang\_id** не равен 0);
  - **value** – значение параметра (если **\$lang\_id** не равен 0).

## **ProdDetails::GetAllNonEmptyDetails**

Назначение:

- Извлечение всех параметров указанного продукта из БД, названия которых заданы для указанного языка.

Объявление:

- `function GetAllNonEmptyDetails($prod_id, $lang_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия и значения параметров продукта.

Возвращаемое значение:

- Тип **array**. Массив параметров продукта, каждым элементом которого является массив, состоящий из следующих ключей:
  - **id** – идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#);
  - **param\_id** – идентификатор параметра категории продукта из таблицы [proshop\\_categories\\_params](#);
  - **caption** – название параметра;
  - **value** – значение параметра.



## **ProdDetails::GetAllParams**

Назначение:

- Извлечение всех параметров указанной категории из БД.

Объявление:

- `function GetAllParams($cat_id, $lang_id = 0, $only_for_filter = false)`

Параметры:

- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия параметров категории. Если данный параметр равен 0, извлекаются только поля из таблицы [proshop\\_categories\\_params](#).
- **bool \$only\_for\_filter**. Если значение данного параметра равно **true**, будут извлечены только те параметры категории, которые должны отображаться в фильтре на витрине. Иначе – будут извлечены все параметры категории.

Возвращаемое значение:

- Тип **array**. Массив, каждым элементом которого является массив, состоящий из следующих ключей:
  - все поля таблицы [proshop\\_categories\\_params](#);
  - **caption** – название параметра категории для указанного языка (если **\$lang\_id** больше нуля).

## **ProdDetails::GetDetail**

Назначение:

- Извлечение информации о параметре продукта из таблицы [proshop\\_products\\_details](#).

Объявление:

- `function GetDetail($id)`

Параметры:

- **int \$id**. Идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_products\\_details](#).

## **ProdDetails::GetDetailValues**

Назначение:

- Извлечение всех значений заданного параметра продукта из таблицы [proshop\\_products\\_details\\_texts](#).

Объявление:

- `function GetDetailValues($id)`

Параметры:

- **int \$id**. Идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#).

Возвращаемое значение:

- Тип **array**. Массив, каждым элементом которого является массив, содержащий ключи, совпадающие с полями таблицы [proshop\\_products\\_details\\_texts](#).

## **ProdDetails::GetOneDetail**

Назначение:

- Извлечение информации об указанном параметре продукта из БД.

Объявление:

- `function GetOneDetail($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор параметра из таблицы [proshop\\_products\\_details](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь строковое значение указанного параметра (зависимое от языка). Если значение **\$lang\_id** равно 0, извлекается только числовое значение параметра.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, состоящий из следующих ключей:
  - все поля таблицы [proshop\\_products\\_details](#);
  - **value** – значение параметра.

## **ProdDetails::GetParam**

Назначение:

- Извлечение информации об указанном параметре категории.

Объявление:

- `function GetParam($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название параметра. Если **\$lang\_id** равен 0, название параметра не извлекается.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_categories\\_params](#);
  - **caption** – название параметра для указанного языка.

## **ProdDetails::GetParamsCount**

Назначение:

- Определение кол-ва параметров указанной категории.

Объявление:

- `function GetParamsCount($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **int**. Кол-во параметров для указанной категории (кол-во записей в таблице [proshop\\_categories\\_params](#) для указанной категории).

## **ProdDetails::IsDetailExists**

Назначение:

- Определение, существует ли параметр для заданного продукта, основанный на заданном параметре категории.

Объявление:

- `function IsDetailExists($prod_id, $param_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$param\_id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **ProdDetails::SetDetailValue**

Назначение:

- Установка значения заданного параметра продукта.

Объявление:

- `function SetDetailValue($id, $value, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор параметра продукта из таблицы [proshop\\_products\\_details](#).
- **int/string \$value**. Значение параметра продукта. Если **\$lang\_id** равен 0, значение считается числовым. Иначе – строковым, заданным для указанного языка.
- **int \$lang\_id**. Идентификатор языка, для которого устанавливается строковое значение параметра. Если данный параметр равен 0, значение **\$value** считается числовым.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdDetails::SetParamCaption**

Назначение:

- Установка названия указанного параметра категории для заданного языка.

Объявление:

- `function SetParamCaption($id, $lang_id, $caption)`

Параметры:

- **int \$id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого устанавливается название указанного параметра.
- **string \$caption**. Название указанного параметра на языке **\$lang\_id**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdDetails::SetParamFilterFlag**

Назначение:

- Установка флага **for\_filter** для заданного параметра категории. Этот флаг определяет, будет ли параметр категории использоваться в фильтре продуктов по параметрам на витрине.

Объявление:

- `function SetParamFilterFlag($id, $for_filter)`

Параметры:

- **int \$id**. Идентификатор параметра категории из таблицы [proshop\\_categories\\_params](#).
- **bool \$for\_filter**. Если значение данного параметра равно **true**, параметр категории **\$id** будет использоваться в фильтре. Иначе – не будет.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdDetails::SetPos**

Назначение:

- Установка позиции указанного параметра продукта по сравнению с другими параметрами этого же продукта. Используется для сортировки параметров продукта.

Объявление:

- `function SetPos($detail_id, $pos)`

Параметры:

- **int \$detail\_id**. Параметр продукта из таблицы [proshop\\_products\\_details](#).
- **int \$pos**. Значение позиции параметра.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс ProdFiles**

Предназначен для обеспечения скачивания файлов электронных продуктов магазина (как платных, так и бесплатных) в многопоточном режиме с сохранением статистики.

### **ProdFiles::ClearActiveUser**

Назначение:

- Удаление информации о пользователе, который скачал файл электронного продукта, из таблицы [proshop\\_products\\_files\\_sessions](#).

Объявление:

- function ClearActiveUser(\$ip, \$uagent)

Параметры:

- **string \$ip**. IP-адрес пользователя.
- **string \$uagent**. Строка User Agent скачивающей программы (или браузера) пользователя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::CreateKey**

Назначение:

- Генерирование ключа (или определение существующего) для скачивания электронного продукта.

Объявление:

- function CreateKey(\$file\_id, \$order\_id, \$order\_prod\_id, \$visitor\_id, \$email)

Параметры:

- **int \$file\_id**. Идентификатор скачиваемого файла из таблицы [proshop\\_products\\_files](#).
- **int \$order\_id**. Идентификатор заказа из таблицы [proshop\\_orders](#).
- **int \$order\_prod\_id**. Идентификатор записи о заказанном продукте в таблице [proshop\\_orders\\_products](#).
- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#), который сделал заказ.
- **string \$email**. Адрес электронной почты пользователя, который сделал заказ.

Возвращаемое значение:

- Тип **string**. Ключ для скачивания электронного продукта из таблицы [proshop\\_products\\_files\\_keys](#), или пустая строка в случае ошибки.

## **ProdFiles::Delete**

Назначение:

- Удаление файла указанного электронного продукта. Файл удаляется только в том случае, если он был загружен на сервер через админку. Файл, загруженный администратором вручную через FTP, не удаляется.

Объявление:

- function Delete(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор электронного продукта из таблицы [proshop\\_products](#), файл которого необходимо удалить.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::DeleteKey**

Назначение:

- Удаление ключа для скачивания электронного продукта из БД.

Объявление:

- function DeleteKey(\$id)

Параметры:

- **int \$id**. Идентификатор ключа из таблицы [proshop\\_products\\_files\\_keys](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **ProdFiles::DelExpiredKeys**

Назначение:

- Удаление просроченных ключей для скачивания электронных продуктов из БД (время создания которых меньше текущего времени на указанное кол-во часов).

Объявление:

- function DelExpiredKeys(\$ttl)

Параметры:

- **int \$ttl**. Срок действия в часах, при превышении которого ключ удаляется.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::FileExists**

Назначение:

- Определение, существует ли файл указанного электронного продукта (не удален ли он случайно).

Объявление:

- function FileExists(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор электронного продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **ProdFiles::Get**

Назначение:

- Извлечение информации о файле указанного электронного продукта.

Объявление:

- function Get(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор электронного продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **array**. Массив с информацией, содержащий следующие ключи:
  - **name** – имя файла или полный путь к нему на диске сервера;
  - **size** – размер файла в Кб;
  - **path** – полный путь к файлу на диске сервера.

## **ProdFiles::GetAllKeys**

Назначение:

- Извлечение всех ключей для скачивания электронных продуктов из таблицы [proshop\\_products\\_files\\_keys](#) в порядке возрастания времени последнего доступа к каждому из них.

Объявление:

- `function GetAllKeys($limit = 0)`

Параметры:

- **int \$limit**. Максимальное кол-во возвращаемых ключей. Если данный параметр равен 0, возвращаются все ключи.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим ключи, совпадающие с полями таблицы [proshop\\_products\\_files\\_keys](#).

## **ProdFiles::GetAllKeysCount**

Назначение:

- Определение кол-ва записей таблицы [proshop\\_products\\_files\\_keys](#).

Объявление:

- `function GetAllKeysCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во записей.

## **ProdFiles::GetIdByProdId**

Назначение:

- Определение идентификатора файла по идентификатору электронного продукта, которому принадлежит этот файл.

Объявление:

- `function GetIdByProd($prod_id)`

Параметры:

- **int \$prod\_id.** Идентификатор электронного продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **int**. Идентификатор файла из таблицы [proshop\\_products\\_files](#).

## **ProdFiles::GetOne**

Назначение:

- Извлечение информации об указанном файле электронного продукта из таблицы [proshop\\_products\\_files](#).

Объявление:

- function GetOne(\$file\_id)

Параметры:

- **int \$file\_id.** Идентификатор записи в таблице [proshop\\_products\\_files](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_products\\_files](#).

## **ProdFiles::GetOneKey**

Назначение:

- Извлечение информации о ключе для скачивания электронного продукта.

Объявление:

- function GetOneKey(\$file\_id, \$key)

Параметры:

- **int \$file\_id.** Идентификатор файла электронного продукта из таблицы [proshop\\_products\\_files](#).
- **string \$key.** Ключ для скачивания электронного продукта.

Возвращаемое значение:

- Тип **array**. Массив, содержащий ключи, совпадающие с полями таблицы [proshop\\_products\\_files\\_keys](#).

## **ProdFiles::GetProdIdById**

Назначение:

- Определение идентификатора электронного продукта по идентификатору его файла.

Объявление:

- function GetProdIdById(\$file\_id)

Параметры:

- **int \$file\_id.** Идентификатор файла из таблицы [proshop\\_products\\_files](#).

Возвращаемое значение:

- Тип **int**. Идентификатор продукта из таблицы [proshop\\_products](#).

## **ProdFiles::GetTimeToAllowFree**

Назначение:



- Определение количества часов, по истечении которого указанному посетителю можно будет скачать любой бесплатный электронный продукт.

Объявление:

- `function GetTimeToAllowFree($visitor_id, $interval, $file_id = 0)`

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).
- **int \$interval**. Разрешенный интервал между скачиваниями, в часах. Берется из настроек магазина (см. [\\$CONFIG](#)).
- **int \$file\_id**. Идентификатор файла электронного продукта из таблицы [proshop\\_products\\_files](#). Должен задаваться только когда пользователь скачивает файл. Когда требуется просто определить кол-во часов, оставшееся до новой возможности скачивания, данный параметр должен быть равен 0.

Возвращаемое значение:

- Тип **float**. Кол-во часов до возможности скачать еще один бесплатный электронный продукт.

### **ProdFiles::IsActiveUser**

Назначение:

- Определение, скачивает ли в данный момент какой-нибудь файл бесплатного электронного продукта пользователь с заданными IP-адресом и строкой User Agent скачивающей программы (или браузера). Применяется для блокировки многопоточности при скачивании бесплатных электронных продуктов.

Объявление:

- `function IsActiveUser($ip, $uagent)`

Параметры:

- **string \$ip**. IP-адрес проверяемого пользователя.
- **string \$uagent**. Строка User Agent программы проверяемого пользователя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

### **ProdFiles::ProlongKey**

Назначение:

- Продление действия указанного ключа скачивания электронного продукта на заданное кол-во часов.

Объявление:

- `function ProlongKey($id, $hours)`

Параметры:

- **int \$id**. Идентификатор ключа из таблицы [proshop\\_products\\_files\\_keys](#).
- **int \$hours**. Кол-во часов, на которое необходимо продлить действие указанного ключа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::SetActiveUser**

Назначение:

- Сохранение информации о скачивающем файл бесплатного электронного продукта пользователе (IP-адрес и строка User Agent скачивающей программы (или браузера)) в таблице [proshop\\_products\\_files\\_sessions](#). Применяется для блокировки многопоточности при скачивании бесплатных электронных продуктов.

Объявление:

- function SetActiveUser(\$ip, \$uagent)

Параметры:

- **string \$ip**. IP-адрес пользователя.
- **string \$uagent**. Строка User Agent скачивающей программы пользователя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::SetKeyAllowed**

Назначение:

- Включение/отключение действия указанного ключа скачивания электронного продукта.

Объявление:

- function SetKeyAllowed(\$id, \$allowed)

Параметры:

- **int \$id**. Идентификатор ключа из таблицы [proshop\\_products\\_files\\_keys](#).
- **char \$allowed**. Флаг включения/отключения (**Y** – включен, **N** - выключен).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::SetKeyLastAccess**

Назначение:

- Установка времени последнего доступа к указанному ключу скачивания заданного файла (устанавливается время в момент вызова) электронного продукта.

Объявление:

- function SetKeyLastAccess(\$file\_id, \$key)

Параметры:

- **int \$file\_id**. Идентификатор файла электронного продукта из таблицы [proshop\\_products\\_files](#).
- **string \$key**. Ключ скачивания.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::SetName**

Назначение:

- Установка имени файла заданного электронного продукта в таблице [proshop\\_products\\_files](#).

Объявление:

- `function SetName($prod_id, $filename)`

Параметры:

- **int \$prod\_id**. Идентификатор электронного продукта из таблицы [proshop\\_products](#).
- **string \$filename**. Имя файла, которое будет записано в поле **name** таблицы [proshop\\_products\\_files](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::SetPath**

Назначение:

- Установка полного пути к файлу (на сервере магазина) заданного электронного продукта в таблице [proshop\\_products\\_files](#).

Объявление:

- `function SetPath($prod_id, $filepath)`

Параметры:

- **int \$prod\_id**. Идентификатор электронного продукта из таблицы [proshop\\_products](#).
- **string \$filepath**. Путь к файлу, который будет записан в поле **path** таблицы [proshop\\_products\\_files](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdFiles::Upload**

Назначение:

- Загрузка файла электронного продукта на сервер магазина.

Объявление:

- `function Upload($prod_id, $file)`

Параметры:

- **int \$prod\_id**. Идентификатор электронного продукта из таблицы [proshop\\_products](#).
- **file \$file**. Описание загружаемого файла (из **\$\_FILES**).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – см. возвращаемое значение функции [ProdFiles::Get](#).

## Класс ProdOptions

Предназначен для управления опциями продуктов. Например, цвет одежды или размер обуви являются опциями.

### ProdOptions::AddException

Назначение:

- Добавление комбинации вариантов опций продукта, которую нельзя использовать для заказа.

Объявление:

- `function AddException($prod_id, $vars)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **string \$vars**. Список вариантов опций продукта, составляющий комбинацию запрещенных вариантов опций. Список должен состоять из идентификаторов вариантов опций (из таблицы [proshop\\_products\\_options\\_vars](#)), разделенных дефисом.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданной записи в таблице [proshop\\_products\\_options\\_exceptions](#).

### ProdOptions::AddRoot

Назначение:

- Добавление базовой информации об опции продукта в таблицу [proshop\\_products\\_options](#).

Объявление:

- `function AddRoot($type_id, $prod_id = 0, $is_global = 'Y')`

Параметры:

- **int \$type\_id**. Идентификатор типа опции из таблицы [proshop\\_products\\_options\\_types](#).
- **int \$prod\_id**. Идентификатор продукта (из таблицы [proshop\\_products](#)), которому должна принадлежать опция. Данный параметр должен быть равен 0, если параметр **\$is\_global** имеет значение Y.
- **char \$is\_global**. Флаг глобальности опции. Если значение данного параметра равно Y, опция считается глобальной, т.е. ее можно клонировать при создании опций для тех или иных продуктов. В этом случае значение параметра **\$prod\_id** игнорируется (считается равным 0).

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор вставленной в таблицу [proshop\\_products\\_options](#) записи.

### ProdOptions::AddTexts

Назначение:

- Добавление названия указанной опции продукта на указанном языке в БД.

Объявление:

- `function AddTexts($id, $lang_id, $caption)`

Параметры:

- **int \$id**. Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо добавить название опции.
- **string \$caption**. Название опции. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::AddVarRoot**

Назначение:

- Добавление варианта указанной опции продукта в таблицу [proshop\\_products\\_options\\_vars](#).

Объявление:

- `function AddVarRoot($id, $price_mod, $price_mod_type)`

Параметры:

- **int \$id**. Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **float \$price\_mod**. Модификатор цены продукта в базовой валюте магазина или в процентах (в зависимости от значения параметра **\$price\_mod\_type**). Может быть положительным, отрицательным или равным 0.
- **string \$price\_mod\_type**. Тип модификатора цены: **abs** – абсолютный (в базовой валюте магазина), **per** – в процентах.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданной в таблице [proshop\\_products\\_options\\_vars](#) записи.

## **ProdOptions::AddVarTexts**

Назначение:

- Добавление названия указанного варианта опции продукта на указанном языке в БД.

Объявление:

- `function AddVarTexts($id, $lang_id, $caption)`

Параметры:

- **int \$id**. Идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо добавить название варианта опции.
- **string \$caption**. Название варианта опции. Макс. 254 символа.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор вставленной в таблицу [proshop\\_products\\_options\\_vars\\_texts](#) записи.

## **ProdOptions::AllowOnlyVariants**

Назначение:

- Установка флага, который разрешает покупателю выбор только вариантов продукта (созданных в таблице [proshop\\_products\\_variants](#)), а не отдельных опций (т.е. все комбинации опций, кроме описанных в вариантах продукта, будут запрещены).

Объявление:

- function AllowOnlyVariants(\$prod\_id, \$only\_variants)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **bool \$only\_variants**. Если значение данного параметра равно true, будут разрешены только варианты продукта. Если значение равно false, будут также разрешены и незапрещенные комбинации опций продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::ClearDefVarsCombination**

Назначение:

- Очистка комбинации вариантов опций по умолчанию для указанного продукта (удаление соответствующей записи из таблицы [proshop\\_products\\_options\\_def\\_combs](#)).

Объявление:

- function ClearDefVarsCombination(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::CloneGlobal**

Назначение:

- Клонирование глобальной опции для указанного продукта (копирование глобальной опции в опцию указанного продукта).

Объявление:

- function CloneGlobal(\$id, \$prod\_id)

Параметры:

- **int \$id**. Идентификатор глобальной опции из таблицы [proshop\\_products\\_options](#).
- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#), для которого необходимо создать копию указанной глобальной опции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::Delete**

Назначение:

- Удаление указанной опции продукта из БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор опции из таблицы [proshop\\_products\\_options](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **ProdOptions::DeleteByProd**

Назначение:

- Удаление всех опций указанного продукта из БД.

Объявление:

- function DeleteByProd(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::DeleteVar**

Назначение:

- Удаление указанного варианта опции продукта из БД.

Объявление:

- function DeleteVar(\$id)

Параметры:

- **int \$id**. Идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::DelException**

Назначение:

- Удаление указанной исключенной комбинации вариантов опций продукта (записи из таблицы [proshop\\_products\\_options\\_exceptions](#)).

Объявление:

- function DelException(\$exc\_id)

Параметры:

- **int \$exc\_id**. Идентификатор записи в таблице [proshop\\_products\\_options\\_exceptions](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::GetAll**

Назначение:

- Извлечение всех опций продуктов из БД.

Объявление:

- `function GetAll($lang_id, $only_global = true, $prod_id = 0)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название опции. Если значение данного параметра равно 0, название не извлекается.
- **bool \$only\_global**. Если значение данного параметра равно true, извлекаются только глобальные опции. При этом значение параметра **\$prod\_id** игнорируется. Если false - извлекаются опции для указанного продукта с идентификатором **\$prod\_id**.
- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#), для которого необходимо извлечь опции. Должен быть равен 0, если **\$only\_global** имеет значение true.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим ключи:
  - все поля таблицы [proshop\\_products\\_options](#);
  - **caption** – название опции (если **\$lang\_id** не равен 0).

## **ProdOptions::GetAllVars**

Назначение:

- Извлечение всех вариантов указанной опции продукта из БД.

Объявление:

- `function GetAllVars($option_id, $lang_id = 0)`

Параметры:

- **int \$option\_id**. Идентификатор опции из таблицы [proshop\\_products\\_options](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия вариантов опции. Если значение данного параметра равно 0, названия вариантов не извлекаются.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим ключи:
  - все поля таблицы [proshop\\_products\\_options\\_vars](#);
  - **caption** – название варианта для указанного языка (если **\$lang\_id** не равен 0).



## **ProdOptions::GetCaption**

Назначение:

- Определение названия указанной опции продукта для указанного языка.

Объявление:

- `function GetCaption($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название указанной опции.

Возвращаемое значение:

- Тип **string**. Название опции продукта.

## **ProdOptions::GetDefVarsCombination**

Назначение:

- Извлечение комбинации вариантов опций по умолчанию для указанного продукта. Комбинация по умолчанию – это комбинация вариантов, которые показаны на витрине в карточке продукта при ее загрузке.

Объявление:

- `function GetDefVarsCombination($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **string**. Строка, состоящая из идентификаторов вариантов опций продукта (из таблицы [proshop\\_products\\_options\\_vars](#)), разделенных дефисом.

## **ProdOptions::GetExceptions**

Назначение:

- Извлечение запрещенных комбинаций вариантов опций продукта (соответствующие записи в таблице [proshop\\_products\\_options\\_exceptions](#)).

Объявление:

- `function GetExceptions($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_products\\_options\\_exceptions](#).

## **ProdOptions::GetIdByCaption**

Назначение:

- Определение идентификатора опции указанного продукта по ее названию на указанном языке. Данная функция может некорректно работать, если у продукта имеются опции с абсолютно одинаковыми названиями.

Объявление:

- `function GetByIdByCaption($prod_id, $caption, $lang_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#), для которого нужно определить идентификатор опции по ее названию.
- **string \$caption**. Название опции на языке **\$lang\_id**.
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **int**. Идентификатор опции из таблицы [proshop\\_products\\_options](#) или 0, если опция с заданным названием не найдена.

## **ProdOptions::GetOne**

Назначение:

- Извлечение информации об указанной опции продукта из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название опции продукта. Если значение данного параметра равно 0, название опции продукта не извлекается.

Возвращаемое значение:

- Тип **array/bool**. **False** – в случае ошибки. Иначе – массив, состоящий из ключей:
  - все поля таблицы [proshop\\_products\\_options](#);
  - **caption** – название опции продукта для указанного языка (если **\$lang\_id** не равен 0).

## **ProdOptions::GetOneVar**

Назначение:

- Извлечение информации об указанном варианте опции продукта из БД.

Объявление:

- `function GetOneVar($var_id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название варианта опции продукта. Если значение данного параметра равно 0, название варианта опции продукта не извлекается.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, состоящий из ключей:
  - все поля таблицы [proshop\\_products\\_options\\_vars](#);
  - **caption** – название варианта опции продукта для указанного языка (если **\$lang\_id** не равен 0).

## **ProdOptions::GetOptionVariants**

Назначение:

- Извлечение вариантов указанной опции продукта с учетом указанных языка и валюты.

Объявление:

- function GetOptionVariants(\$id, \$lang\_id, \$curr\_id)

Параметры:

- **int \$id**. Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия вариантов указанной опции продукта.
- **int \$curr\_id**. Идентификатор валюты (из таблицы [proshop\\_currencies](#)), для которой необходимо преобразовать значения абсолютных модификаторов цен у вариантов продукта (произвести конвертацию с базовой валюты).

Возвращаемое значение:

- Тип **array**. См. возвращаемое значение функции [ProdOptions::GetAllVars](#).

## **ProdOptions::GetTypes**

Назначение:

- Извлечение всех типов отображения вариантов опции продукта из таблицы [proshop\\_products\\_options\\_types](#).

Объявление:

- function GetTypes()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_products\\_options\\_types](#).

## **ProdOptions::GetVarCaption**

Назначение:

- Определение названия указанного варианта опции продукта для указанного языка.

Объявление:

- function GetVarCaption(\$id, \$lang\_id)

Параметры:

- **int \$id.** Идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$lang\_id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название указанного варианта опции.

Возвращаемое значение:

- Тип **string**. Название варианта опции продукта.

## **ProdOptions::GetVarIdByCaption**

Назначение:

- Определение идентификатора варианта указанной опции по его названию на указанном языке. Данная функция может некорректно работать, если у опции имеются варианты с абсолютно одинаковыми названиями (текстами).

Объявление:

- `function GetVarIdByCaption($option_id, $caption, $lang_id)`

Параметры:

- **int \$option\_id.** Идентификатор опции из таблицы [proshop\\_products\\_options](#), для которой нужно определить идентификатор варианта по его названию.
- **string \$caption.** Название варианта на языке **\$lang\_id**.
- **int \$lang\_id.** Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **int**. Идентификатор варианта из таблицы [proshop\\_products\\_options\\_vars](#) или 0, если вариант с заданным названием не найден.

## **ProdOptions::IsExceptioned**

Назначение:

- Определение, является ли указанная комбинация вариантов опций указанного продукта запрещенной.

Объявление:

- `function IsExceptioned($prod_id, $options)`

Параметры:

- **int \$prod\_id.** Идентификатор продукта из таблицы [proshop\\_products](#).
- **array \$options.** Массив с идентификаторами вариантов опций продукта из таблицы [proshop\\_products\\_options\\_vars](#), которые необходимо проверить на запрещенность.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **ProdOptions::IsOnlyVariantsAllowed**

Назначение:

- Определение, включен ли флаг доступности только комбинаций вариантов опций указанного продукта, принадлежащих вариантам продукта (из таблицы [proshop\\_products\\_variants](#)). См. также описание функции [ProdOptions::AllowOnlyVariants](#).

Объявление:

- `function IsOnlyVariantsAllowed($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **ProdOptions::IsOptionsCorrect**

Назначение:

- Определение, является ли заданный набор опций корректным (существуют заданные ли варианты опций в базе данных).

Объявление:

- `function IsOptionsCorrect($options)`

Параметры:

- **array \$options**. Массив с идентификаторами вариантов опций продукта из таблицы [proshop\\_products\\_options\\_vars](#), которые необходимо проверить на корректность.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **ProdOptions::ModifProdByOptions**

Назначение:

- Модифицирование информации о продукте, взятой из таблицы [proshop\\_products](#), в соответствии с заданным набором вариантов опций продукта. Модифицируются вес, артикул, доступное кол-во на складе, изображение, если заданный набор вариантов опций продукта соответствует какому-либо варианту продукта из таблицы [proshop\\_products\\_variants](#).

Объявление:

- `function ModifProdByOptions($prod, $options, &$variant_id = 0)`

Параметры:

- **array \$prod**. Массив (с информацией о продукте), ключи которого совпадают с полями таблицы [proshop\\_products](#).
- **array \$options**. Массив идентификаторов вариантов опций продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$variant\_id**. Необязательная переменная, в которую следует записать идентификатор варианта продукта (из таблицы [proshop\\_products\\_variants](#)), если ему соответствует указанный набор опций **\$options**.

Возвращаемое значение:

- Тип **array**. Модифицированный (или не измененный) массив **\$prod**.

## **ProdOptions::ModifProdPriceByOptions**

Назначение:

- Модификация цены продукта в соответствии с указанными вариантами опций продукта, которые могут иметь модификаторы цен.

Объявление:

- `function ModifProdPriceByOptions($price, $options)`

Параметры:

- **float \$price**. Цена продукта из таблицы [proshop\\_products](#) в базовой валюте магазина.
- **array \$options**. Массив идентификаторов вариантов опций продукта (из таблицы [proshop\\_products\\_options\\_vars](#)).

Возвращаемое значение:

- Тип **float**. Модифицированная (или не измененная) цена продукта **\$price**.

## **ProdOptions::SetDefVarsCombination**

Назначение:

- Установка комбинации вариантов опций по умолчанию для указанного продукта.

Объявление:

- `function SetDefVarsCombination($prod_id, $options)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **array \$options**. Массив идентификаторов вариантов опций продукта (из таблицы [proshop\\_products\\_options\\_vars](#)).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::SetPos**

Назначение:

- Установка позиции указанной опции продукта по сравнению с остальными опциями этого же продукта. Используется для установки порядка вывода опций продукта.

Объявление:

- `function SetPos($option_id, $pos)`

Параметры:

- **int \$option\_id**. Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::SetVarPos**

Назначение:

- Установка позиции указанного варианта опции продукта по сравнению с остальными вариантами этой же опции. Используется для установки порядка вывода вариантов опции продукта.

Объявление:

- `function SetVarPos($var_id, $pos)`

Параметры:

- **int \$var\_id.** Идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$pos.** Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::Update**

Назначение:

- Обновление информации об указанной опции продукта в БД.

Объявление:

- `function Update($id, $lang_id, $info)`

Параметры:

- **int \$id.** Идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **int \$lang\_id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо установить название опции продукта.
- **array \$info.** Массив с информацией, содержащий следующие ключи:
  - **type\_id** – идентификатор типа опции продукта (из таблицы [proshop\\_products\\_options\\_types](#));
  - **caption** – название опции продукта на указанном языке.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdOptions::UpdateVar**

Назначение:

- Обновление информации об указанном варианте опции продукта в БД.

Объявление:

- `function UpdateVar($id, $lang_id, $info)`

Параметры:

- **int \$id.** Идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **int \$lang\_id.** Идентификатор языка (из таблицы [proshop\\_languages](#)).

- **array \$info**. Массив с информацией об указанном варианте опции продукта. Должен содержать следующие ключи:
  - **caption** – название варианта опции продукта на указанном языке **\$lang\_id**;
  - **price\_mod** - модификатор цены продукта в базовой валюте магазина или в процентах (в зависимости от значения параметра **price\_mod\_type**). Может быть положительным, отрицательным или равным 0;
  - **price\_mod\_type** - тип модификатора цены: **abs** – абсолютный (в базовой валюте магазина), **per** – в процентах.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс ProdPictures

Предназначен для управления дополнительными изображениями товаров.

### **ProdPictures::AddRoot**

Назначение:

- Добавление базовой информации о дополнительном изображении товара в таблицу [proshop\\_products\\_pictures](#).

Объявление:

- function AddRoot(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **int**. Идентификатор добавленного изображения из таблицы [proshop\\_products\\_pictures](#).

### **ProdPictures::AddText**

Назначение:

- Добавление текста атрибута ALT для указанного изображения товара на указанном языке.

Объявление:

- function AddText(\$pic\_id, \$lang\_id, \$alt)

Параметры:

- **int \$pic\_id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого добавляется текст атрибута ALT изображения товара.
- **string \$alt**. Текст атрибута ALT на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.



## **ProdPictures::Delete**

Назначение:

- Удаление указанного дополнительного изображения товара.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **ProdPictures::GetAll**

Назначение:

- Извлечение всех дополнительных изображений для указанного продукта.

Объявление:

- `function GetAll($prod_id, $lang_id = 0)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь тексты атрибутов ALT дополнительных изображений. Если значение данного параметра равно 0, тексты атрибутов ALT не извлекаются.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого также является массивом, состоящим из следующих ключей:
  - все поля таблицы [proshop\\_products\\_pictures](#);
  - **alt** – текст атрибута ALT дополнительного изображения продукта на указанном языке (если **\$lang\_id** не равен 0).

## **ProdPictures::GetAlt**

Назначение:

- Извлечение текста атрибута ALT указанного изображения на указанном языке.

Объявление:

- `function GetAlt($pic_id, $lang_id)`

Параметры:

- **int \$pic\_id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь текст атрибута ALT указанного изображения.

Возвращаемое значение:

- Тип **string**. Текст атрибута ALT.

## **ProdPictures::LoadImage**

Назначение:

- Загрузка дополнительного изображения продукта в магазин.

Объявление:

- function LoadImage(\$pic\_id, \$file, \$tmpfile)

Параметры:

- **int \$pic\_id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).
- **file \$file**. Описание загружаемого файла (из **\$\_FILES**).
- **string \$tmpfile**. Путь к временному файлу изображения, загруженному с помощью браузера.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – имя созданного файла дополнительного изображения продукта (файлы изображений продуктов содержатся в директории **images/products**).

## **ProdPictures::SetFilename**

Назначение:

- Установка имени файла для указанного дополнительного изображения продукта в таблице [proshop\\_products\\_pictures](#).

Объявление:

- function SetFilename(\$pic\_id, \$filename)

Параметры:

- **int \$pic\_id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).
- **string \$filename**. Имя файла дополнительного изображения продукта. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdPictures::SetPos**

Назначение:

- Установка позиции дополнительного изображения продукта относительно других дополнительных изображений этого же продукта.

Объявление:

- function SetPos(\$pic\_id, \$pos)

Параметры:

- **int \$pic\_id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdPictures::UpdateAlt**

Назначение:

- Обновление текста атрибута ALT для указанного дополнительного изображения продукта на указанном языке.

Объявление:

- `function UpdateAlt($pic_id, $lang_id, $alt)`

Параметры:

- **int \$pic\_id**. Идентификатор изображения из таблицы [proshop\\_products\\_pictures](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы `proshop_languages`), для которого необходимо обновить текст атрибута ALT.
- **string \$alt**. Текст атрибута ALT. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс ProdPins**

Предназначен для управления секретными кодами цифровых продуктов (пин-кодами).

### **ProdPins::Add**

Назначение:

- Добавление секретного кода к указанному цифровому продукту.

Объявление:

- `function Add($prod_id, $code)`

Параметры:

- **int \$prod\_id**. Идентификатор цифрового продукта из таблицы [proshop\\_products](#).
- **string \$code**. Значение секретного кода. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **ProdPins::Clear**

Назначение:

- Удаление всех секретных кодов указанного цифрового продукта из таблицы [proshop\\_products\\_pins](#).

Объявление:

- `function Clear($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор цифрового продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdPins::Count**

Назначение:

- Подсчет кол-ва секретных кодов указанного цифрового продукта в таблице [proshop\\_products\\_pins](#).

Объявление:

- `function Count($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор цифрового продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **int**. Кол-во секретных кодов, привязанных к заданному продукту **\$prod\_id**.

## **ProdPins::GetAll**

Назначение:

- Извлечение всех секретных кодов указанного цифрового продукта.

Объявление:

- `function GetAll($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор цифрового продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **array**. Массив значений секретных кодов.

## **ProdPins::Pop**

Назначение:

- Извлечение очередного секретного кода из таблицы [proshop\\_products\\_pins](#) для указанного продукта, а также удаление этого секретного кода из этой таблицы.

Объявление:

- `function Pop($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор цифрового продукта из таблицы [proshop\\_products](#), секретный код которого необходимо получить.

Возвращаемое значение:

- Тип **string**. Возвращает значение секретного кода или пустую строку, если для указанного цифрового продукта секретные коды отсутствуют или закончились.

## **Класс ProdsDiscuss**

Предназначен для управления отзывами о продуктах.

## **ProdsDiscuss::Add**

Назначение:

- Добавление отзыва об указанном продукте в БД.

Объявление:

- `function Add($prod_id, $visitor_id, $user_id, $author, $email, $rate, $comment, $lang_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#). Во время работы магазина можно взять из `$_SESSION['visitor_id']`.
- **int \$user\_id**. Идентификатор зарегистрированного пользователя (из таблицы [proshop\\_users](#)), который оставляет отзыв. Может быть 0, если отзыв оставляет незарегистрированный пользователь.
- **string \$author**. Имя автора отзыва. Макс. 254 символа.
- **string \$email**. Адрес электронной почты автора отзыва. Макс. 254 символа. Может быть пустой строкой.
- **int \$rate**. Оценка продукта по 5-балльной шкале.
- **string \$comment**. Комментарий пользователя к оценке продукта. Макс. 65533 символа. Может быть пустым.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), который был выбран в магазине в момент создания комментария.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdsDiscuss::Approve**

Назначение:

- Одобрение указанного отзыва о продукте модератором.

Объявление:

- `function Approve($id)`

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_products\\_discuss](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdsDiscuss::Decline**

Назначение:

- Запрет публикации указанного отзыва о продукте. Операция, обратная одобрению отзыва.

Объявление:

- `function Decline($id)`

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_products\\_discuss](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdsDiscuss::Delete**

Назначение:

- Удаление указанного отзыва о продукте из БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_products\\_discuss](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdsDiscuss::GetAll**

Назначение:

- Извлечение всех отзывов о продуктах из БД для указанного языка. Извлекаются только отзывы, имеющие ненулевую длину комментария.

Объявление:

- function GetAll(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь отзывы.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_products\\_discuss](#).

## **ProdsDiscuss::GetAllApproved**

Назначение:

- Извлечение всех одобренных отзывов об указанном продукте из БД для указанного языка. Извлекаются только отзывы, имеющие ненулевую длину комментария.

Объявление:

- function GetAllApproved(\$prod\_id, \$lang\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором необходимо извлечь отзывы.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_products\\_discuss](#).

## **ProdsDiscuss::GetAllInapproved**

Назначение:

- Извлечение всех неодобренных отзывов о продуктах из БД для указанного языка.

Объявление:

- `function GetAllInapproved($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь отзывы.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_products\\_discuss](#).

## **ProdsDiscuss::GetCount**

Назначение:

- Определение кол-ва отзывов о продуктах или об определенном продукте (включая простые голосования без комментариев).

Объявление:

- `function GetCount($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#). Если данный параметр имеет значение 0, учитываются все продукты.

Возвращаемое значение:

- Тип **int**. Кол-во отзывов.

## **ProdsDiscuss::GetInapprovedCount**

Назначение:

- Определение кол-ва неодобренных отзывов о продуктах на указанном языке.

Объявление:

- `function GetInapprovedCount($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)).

Возвращаемое значение:

- Тип **int**. Кол-во неодобренных отзывов.

## **ProdsDiscuss::GetTopRated**

Назначение:

- Извлечение N самых рейтинговых продуктов из БД для указанной категории.

Объявление:

- `function GetTopRated($cat_id, $lang_id, $num)`

Параметры:

- **int \$cat\_id.** Идентификатор категории из таблицы [proshop\\_categories](#). Если данный параметр равен 0, возвращается массив самых рейтинговых товаров для всего магазина.
- **int \$lang\_id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь текстовые составляющие каждого продукта (название и т.п.). Параметр может быть равен 0, что означает не извлекать текстовые составляющие.
- **int \$num.** Максимальное кол-во самых рейтинговых извлекаемых товаров.

Возвращаемое значение:

- Тип **array**. См. возвращаемое значение функции [Products::Search](#).

## **ProdsDiscuss::IsDiscussed**

Назначение:

- Определение, оставлял ли уже указанный пользователь (посетитель) отзыв об указанном продукте. Применяется для защиты от накрутки рейтинга продукта.

Объявление:

- `function IsDiscussed($prod_id, $visitor_id, $user_id)`

Параметры:

- **int \$prod\_id.** Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$visitor\_id.** Идентификатор посетителя из таблицы [proshop\\_visitors](#). Можно также использовать `$_SESSION['visitor_id']`.
- **int \$user\_id.** Идентификатор зарегистрированного пользователя из таблицы [proshop\\_users](#), или 0, если пользователь не зарегистрирован (или не авторизован).

Возвращаемое значение:

- Тип **bool**. True, если указанный пользователь уже оставлял отзыв об указанном продукте. Иначе – false.

## **ProdsDiscuss::SetComment**

Назначение:

- Установка текста комментария указанного отзыва о продукте.

Объявление:

- `function SetComment($id, $comment)`

Параметры:

- **int \$id.** Идентификатор отзыва из таблицы [proshop\\_products\\_discuss](#).
- **string \$comment.** Текст комментария. Макс. 65533 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.



## Класс Products

### Products::Add

Назначение:

- Добавление продукта в БД.

Объявление:

- function Add(\$cat, \$prod)

Параметры:

- **int \$cat**. Идентификатор категории из таблицы [proshop\\_categories](#), в которую необходимо добавить продукт.
- **array \$info**. Массив с информацией о продукте. Должен содержать следующие ключи:
  - **eng\_name** – имя в URL (для modRewrite) продукта. Макс. 254 символа. Может быть пустым;
  - **sku** – артикул продукта. Макс. 254 символа. Может быть пустым;
  - **price** – цена одного экземпляра продукта в базовой валюте;
  - **prev\_price** – старая (зачеркнутая) цена продукта в базовой валюте;
  - **qty** – доступное кол-во экземпляров продукта на складе. Если значение – пустая строка, считается, что кол-во неограничено. Если 0 – продукта нет на складе;
  - **min\_order\_qty** – минимальное кол-во продукта, которое может заказать покупатель (не может быть меньше 1);
  - **sold\_qty** – кол-во проданных экземпляров продукта за все время его размещения в магазине;
  - **weight** – вес продукта в килограммах;
  - **manuf\_id** – идентификатор производителя продукта (из таблицы [proshop\\_manufacturers](#)). Может быть 0;
  - **is\_visible** – флаг доступности продукта на витрине (**Y** – продукт виден покупателю, **N** – невидим);
  - **egood** – если **Y**, то продукт является электронным (скачиваемым). Для обычных продуктов должен быть равен **N**;
  - **is\_free** – флаг бесплатности электронного продукта (**Y** – электронный продукт бесплатный, **N** – не бесплатный);
  - идентификатор языка. Значением данного ключа является массив, состоящий из следующих ключей (см. пример):
    - **caption** – название продукта на указанном языке. Макс. 254 символа;
    - **descr** – описание продукта на указанном языке. Макс. 65533 символа;
    - **short\_descr** – краткое описание продукта на указанном языке. Макс. 65533 символа;
    - **alt** – альтернативный текст изображения продукта (атрибут ALT) на указанном языке. Макс. 254 символа;

- **if\_null\_qty** – текст на указанном языке, выводимый покупателю в карточке товара, если данного товара нет на складе. Макс. 50 символов.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданного продукта в таблице [proshop\\_products](#).

Пример:

- ```
$info = array(
    'eng_name' => $eng_name,
    'price' => $price,
    // ...
);
foreach ($LANGUAGES as $lang) {
    $info[$lang['id']] = array(
        'caption' => $caption,
        'descr' => $descr,
        'short_descr' => $short_descr,
        'alt' => $alt
    );
}
Products::Add($cat, $info);
```

## **Products::AddAdditionalCat**

Назначение:

- Привязка указанного продукта к дополнительной указанной категории.

Объявление:

- `function AddAdditionalCat($prod_id, $cat_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::DecQty**

Назначение:

- Уменьшение кол-ва доступных экземпляров на складе указанного продукта.

Объявление:

- `function DecQty($prod_id, $qty = 1)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

- **int \$qty**. Значение, на которое необходимо уменьшить кол-во доступных экземпляров указанного продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::DelAdditionalCat**

Назначение:

- Операция, обратная привязке указанного продукта к дополнительной указанной категории (удаление связи продукта и категории).

Объявление:

- `function DelAdditionalCat($prod_id, $cat_id = 0)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#). Если равен 0, удаляются все связи продукта **\$prod\_id** с его дополнительными категориями из таблицы [proshop\\_products\\_categories](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::DelAll**

Назначение:

- Удаление всех продуктов указанной категории.

Объявление:

- `function DelAll($cat)`

Параметры:

- **int \$cat**. Идентификатор категории (из таблицы [proshop\\_categories](#)), продукты которой необходимо удалить.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::Delete**

Назначение:

- Удаление указанного продукта.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Products::DelImage**

Назначение:

- Удаление основного изображения указанного продукта.

Объявление:

- `function DelImage($id)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::DelThumb**

Назначение:

- Удаление уменьшенного варианта основного изображения указанного продукта.

Объявление:

- `function DelThumb($id)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::GetAll**

Назначение:

- Извлечение всех продуктов указанной категории из БД.

Объявление:

- `function GetAll($cat, $lang_id, $complete_sql = "", $only_parent_cat = false, $min_info = false)`

Параметры:

- **int \$cat**. Идентификатор категории (из таблицы [proshop\\_categories](#)), продукты которой необходимо извлечь.
- **int \$lang\_id**. Идентификатор языка (из таблицы `proshop_languages`), для которого необходимо извлечь текстовые составляющие каждого продукта (название, описание и т.п.).
- **string \$complete\_sql**. Часть SQL-запроса, которая может быть пристыкована к концу основного SQL-запроса, формируемого функцией.
- **bool \$only\_parent\_cat**. Если значение данного параметра равно `true`, учитываются только те продукты, главная категория которых совпадает с `$cat`. Иначе – учитываются все продукты указанной категории (в т.ч. и второстепенные продукты, когда `$cat` является дополнительной категорией продукта).
- **bool \$min\_info**. Если значение данного параметра равно `true`, функция не будет извлекать текстовые поля (краткое/полное описание продукта и т.п., кроме названия) для экономии времени и памяти.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого описывает продукт и также является массивом, содержащим следующие ключи:
  - все поля из таблицы [proshop\\_products](#) (кроме некоторых текстовых, если параметр **\$min\_info** равен **true**);
  - **pos** – позиция продукта в указанной категории относительно других продуктов этой же категории;
  - **caption** – название продукта на указанном языке;
  - **descr** – описание продукта на указанном языке (если параметр **\$min\_info** равен **false**);
  - **short\_descr** – краткое описание продукта на указанном языке (если параметр **\$min\_info** равен **false**);
  - **alt** – альтернативный текст изображения продукта (атрибут ALT) на указанном языке (если параметр **\$min\_info** равен **false**);
  - **if\_null\_qty** – текст, выводимый покупателю в карточке товара, если данного товара нет на складе (если параметр **\$min\_info** равен **false**).

## **Products::GetCatIds**

Назначение:

- Извлечение списка идентификаторов категорий, в которые помещен указанный продукт.

Объявление:

- `function GetCatIds($prod_id, $only_additional = false)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **bool \$only\_additional**. Флаг, определяющий, извлекать ли только дополнительные категории продукта: **true** – извлекать только дополнительные категории, **false** – извлекать все категории (главную и дополнительные).

Возвращаемое значение:

- Тип **array**. Массив идентификаторов категорий продукта.

## **Products::GetDiscount**

Назначение:

- Определение размера скидки в базовой валюте магазина для указанного продукта.

Объявление:

- `function GetDiscount($prod_id, $price = "")`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

- **float \$price**. Значение цены экземпляра продукта в базовой валюте без скидки, но с налогом. Если данное значение не задано, используется значение поля **price** из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **float**. Размер скидки в базовой валюте магазина для данного продукта( и заданной цены).

## **Products::GetFinalPrice**

Назначение:

- Определение цены указанного продукта с учетом налога и всех действующих на данный момент скидок.

Объявление:

- `function GetFinalPrice($prod_id, $price = "")`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **float \$price**. Значение цены экземпляра продукта в базовой валюте. Если данное значение не задано, используется значение поля **price** из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **float**. Цена экземпляра продукта с учетом налога и скидок в базовой валюте.

## **Products::GetIdByName**

Назначение:

- Определение идентификатора продукта по его имени в URL (для modRewrite).

Объявление:

- `function GetIdByName($prod_name)`

Параметры:

- **string/int \$prod\_name**. Имя в URL (для modRewrite) продукта или его идентификатор.

Возвращаемое значение:

- Тип **int**. Идентификатор продукта из таблицы [proshop\\_products](#), или 0, если продукт не найден.

## **Products::GetNameById**

Назначение:

- Определение имени в URL продукта (для modRewrite) по его идентификатору.

Объявление:

- `function GetNameById($id)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **string**. Имя в URL продукта или идентификатор продукта, если его имя не задано.

## **Products::GetOne**

Назначение:

- Извлечение информации о продукте из БД.

Объявление:

- `function GetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь текстовые составляющие продукта (название, описание и т.д.). Если значение данного параметра равно 0, текстовые составляющие не извлекаются.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_products](#);
  - **caption** – название продукта на указанном языке;
  - **descr** – описание продукта на указанном языке;
  - **short\_descr** – краткое описание продукта на указанном языке;
  - **alt** – текст атрибута ALT изображения продукта на указанном языке;
  - **if\_null\_qty** – текст на указанном языке, выводимый покупателю в карточке товара, если данного товара нет на складе.

## **Products::GetTax**

Назначение:

- Определение размера налога в базовой валюте магазина для указанного продукта.

Объявление:

- `function GetTax($prod_id, $price = "")`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **float \$price**. Значение цены экземпляра продукта в базовой валюте. Если данное значение не задано, используется значение поля **price** из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **float**. Размер налога в базовой валюте магазина для данного продукта (и заданной цены).

## **Products::GetTexts**

Назначение:

- Извлечение текстовых составляющих (название, описание, ...) указанного продукта на указанном языке.

Объявление:

- function GetTexts(\$prod\_id, \$lang\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - **caption** – название продукта на указанном языке;
  - **descr** – описание продукта на указанном языке;
  - **short\_descr** – краткое описание продукта на указанном языке;
  - **alt** – текст атрибута ALT изображения продукта на указанном языке;
  - **if\_null\_qty** – текст на указанном языке, выводимый покупателю в карточке товара, если данного товара нет на складе.

## **Products::GetTotalCount**

Назначение:

- Определение общего кол-ва продуктов в таблице [proshop\\_products](#).

Объявление:

- function GetTotalCount()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – кол-во всех продуктов в магазине.

## **Products::IdBySKU**

Назначение:

- Определение идентификатора продукта по его артикулу.

Объявление:

- function IdBySKU(\$sku)

Параметры:

- **string \$sku**. Артикул продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **int**. Идентификатор продукта из таблицы [proshop\\_products](#) или 0, если артикул не соответствует ни одному продукту.

## **Products::IdToCaption**

Назначение:

- Определение названия продукта по его идентификатору на заданном языке.

Объявление:



- `function IdToCaption($prod_id, $lang_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string**. Название продукта на указанном языке. Если название не задано, возвращается имя в URL (для `modRewrite`) продукта. Если имя в URL не задано, возвращается идентификатор продукта.

## **Products::IncQty**

Назначение:

- Увеличение кол-ва доступных экземпляров продукта на складе.

Объявление:

- `function IncQty($prod_id, $qty = 1)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$qty**. Значение, на которое необходимо увеличить кол-во экземпляров.

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

## **Products::IncSoldQty**

Назначение:

- Увеличение кол-ва проданных экземпляров указанного продукта.

Объявление:

- `function IncSoldQty($prod_id, $qty = 1)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$qty**. Значение, на которое необходимо увеличить кол-во проданных экземпляров.

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

## **Products::IsExists**

Назначение:

- Определение, существует ли продукт с заданным идентификатором в таблице [proshop\\_products](#).

Объявление:

- `function IsExists($id)`

Параметры:

- **int \$id**. Идентификатор продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Products::IsNameExists**

Назначение:

- Определение, не занято ли уже указанное имя в URL (для modRewrite) продукта.

Объявление:

- `function IsNameExists($name, $id = 0)`

Параметры:

- **string \$name**. Имя в URL продукта.
- **int \$id**. Идентификатор продукта (из таблицы [proshop\\_products](#)), который необходимо игнорировать при поиске имени в URL. Если данный параметр равен 0, учитываются все продукты.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Products::IsSKUExists**

Назначение:

- Определение, не занят ли уже указанный артикул продукта.

Объявление:

- `function IsSKUExists($sku, $prod_id = 0)`

Параметры:

- **string \$sku**. Артикул продукта.
- **int \$prod\_id**. Идентификатор продукта (из таблицы [proshop\\_products](#)), который необходимо игнорировать при поиске артикула. Если данный параметр равен 0, учитываются все продукты.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Products::IsVisible**

Назначение:

- Определение, видим ли указанный продукт на витрине.

Объявление:

- `function IsVisible($prod_id, $consider_qty = true)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **bool \$consider\_qty**. Флаг, определяющий, принимать ли во внимание кол-во на складе продукта: **true** – принимать (если продукт отсутствует на складе, он будет считаться невидимым), **false** – не принимать.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Products::GetQty**

Назначение:

- Определение кол-ва доступных экземпляров указанного продукта на складе.

Объявление:

- `function GetQty($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **string**. Кол-во доступных экземпляров на складе (может быть отрицательным при возможном «отрицательном учете», а также может быть нулем). Если возвращается пустая строка, то считается, что учет данного товара не ведется и его кол-во бесконечно.

## **Products::LoadImage**

Назначение:

- Загрузка основного изображения продукта в магазин.

Объявление:

- `function LoadImage($id, $pic, $tmpfile, $make_thumb = false, $thumb_width = 105, $thumb_height = 96)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **file \$pic**. Описание загружаемого файла изображения (из **\$\_FILES**).
- **string \$tmpfile**. Путь к временному файлу изображения, загруженному с помощью браузера.
- **bool \$make\_thumb**. Если значение данного параметра равно **true**, для заданного продукта также автоматически формируется уменьшенный вариант основного изображения продукта на основе загружаемого изображения.
- **int \$thumb\_width**. Ширина уменьшенного изображения продукта. Данный параметр игнорируется, если **\$make\_thumb** равен **false**.
- **int \$thumb\_height**. Высота уменьшенного изображения продукта. Данный параметр игнорируется, если **\$make\_thumb** равен **false**.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – сформированное имя файла загруженного изображения продукта.

## **Products::LoadThumbnail**

Назначение:

- Загрузка уменьшенного варианта основного изображения продукта в магазин.

Объявление:

- function LoadThumbnail(\$id, \$pic, \$tmpfile, \$width = 105, \$height = 96)

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **file \$pic**. Описание загружаемого файла изображения (из **\$\_FILES**).
- **string \$tmpfile**. Путь к временному файлу изображения, загруженному с помощью браузера.
- **int \$width**. Ширина уменьшенного изображения продукта.
- **int \$height**. Высота уменьшенного изображения продукта.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – сформированное имя файла загруженного уменьшенного варианта изображения продукта.

## **Products::MakeThumbnail**

Назначение:

- Генерирование уменьшенного варианта основного изображения указанного продукта.

Объявление:

- function MakeThumbnail(\$prod\_id, \$width, \$height)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$width**. Ширина уменьшенного изображения продукта.
- **int \$height**. Высота уменьшенного изображения продукта.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе - сформированное имя файла уменьшенного варианта изображения продукта.

## **Products::ModPrice**

Назначение:

- Модификация цены (и старой/зачеркнутой цены) указанного товара на заданный процент (как в большую, так и в меньшую сторону).

Объявление:

- function ModPrice(\$id, \$price\_mod)

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **float \$price\_mod**. Значение процента, на который необходимо изменить цену товара.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::Search**

Назначение:

- Поиск продуктов по заданным правилам (фильтру) в магазине.

Объявление:

- `function Search($rules, $lang_id, $only_visible = true, $only_ids = false, $pfields = 'p.*', $complete_sql = "")`

Параметры:

- **array \$rules.** Массив правил поиска. Правила поиска задаются с помощью ключей массива и соответствующих значений. Описание каждого правила, которое может содержаться в массиве **\$rules**:
  - **\$rule['type' => 'by\_name', 'only\_caption' => \$only\_caption, 'name' => \$name]** – поиск по ключевым словам. Строка **\$name** может содержать несколько слов, разделенных пробелом, которые могут содержаться в названии продукта, в его описании, в кратком описании, в имени URL (для modRewrite) или в артикуле. Если флаг **\$only\_caption** равен **Y**, строка **\$name** ищется только в названии продукта.
  - **\$rule['type' => 'only\_egoods', 'value' => 'Y']** – поиск только среди электронных продуктов.
  - **\$rule['type' => 'no\_egoods', 'value' => 'Y']** – поиск среди всех продуктов, кроме электронных.
  - **\$rule['type' => 'no\_zero\_qty', 'value' => 'Y']** – исключить продукты, кол-во на складе которых равно 0.
  - **\$rule['type' => 'by\_price', 'include\_tax' => \$include\_tax, 'include\_discount' => \$include\_discount, 'from\_price' => \$from\_price, 'to\_price' => \$to\_price]** – поиск продуктов по диапазону цен. Если параметр **\$include\_tax** равен **Y**, при поиске цен учитываются налоги продуктов. Если параметр **\$include\_discount** равен **Y**, при поиске цен учитываются скидки. Параметр **\$from\_price** задает начальное значение диапазона цен в базовой валюте магазина (если данный параметр не задан, начальное значение диапазона считается равным нулю). Параметр **\$to\_price** задает конечное значение диапазона цен в базовой валюте магазина (если данный параметр не задан, конечное значение диапазона считается равным бесконечности).
  - **\$rule['type' => 'by\_weight', 'from\_weight' => \$from\_weight, 'to\_weight' => \$to\_weight]** – поиск продуктов по заданному диапазону веса в кг. Параметр **\$from\_weight** задает начальное значение диапазона (если данный параметр не задан, начальное значение считается равным нулю). Параметр **\$to\_weight** задает конечное значение диапазона (если данный параметр не задан, конечное значение считается равным бесконечности).
  - **\$rule['type' => 'by\_manuf', 'manuf\_id' => \$manuf\_id]** – поиск продуктов по производителю (по идентификатору из таблицы [proshop\\_manufacturers](#)).
  - **\$rule['type' => 'by\_cat', 'cat' => \$cat, 'search\_in\_subcats' => \$search\_in\_subcats]** – поиск в заданной категории. Параметр **\$cat** должен содержать идентификатор категории из таблицы [proshop\\_categories](#). Параметр **\$search\_in\_subcats** определяет, искать продукт во всех подкатегориях указанной категории (**Y**) или нет (**N**).

- **\$rule['type' => 'by\_tag', 'tag\_id' => \$tag\_id]** – поиск продуктов, принадлежащих заданному тэгу. Параметр **\$tag\_id** должен быть идентификатором тэга из таблицы [proshop\\_tags\\_texts](#).
- **\$rule['type' => 'by\_params', 'params\_list' => \$params\_list]** – поиск продуктов по параметрам. Массив **\$params\_list** должен описывать параметры, по которым должен производиться поиск, и каждый элемент данного массива должен состоять из следующих ключей:
  - **id** – идентификатор параметра из таблицы [proshop\\_categories\\_params](#);
  - **is\_numeric** – является ли параметр числовым (**Y**) или строковым (**N**);
  - **from\_value** – начальное значение диапазона поиска по числовому параметру (если **is\_numeric** равен **Y**). Может быть не задано, что будет означать 0;
  - **to\_value** – конечное значение диапазона поиска по числовому параметру (если **is\_numeric** равен **N**). Может быть не задано, что будет означать бесконечность;
  - **value** – строковое значение параметра (если **is\_numeric** не равен **Y**), которое необходимо найти.
- **bool \$only\_visible**. Данный параметр должен быть true, если требуется найти только видимые (активные) на витрине продукты. Если данный параметр равен false, будут искаться все продукты.
- **bool \$only\_ids**. Если значение данного параметра равно true, функцией будут возвращены только идентификаторы найденных продуктов.
- **string \$pfields**. Перечисление полей таблицы [proshop\\_products](#), значения которых должны быть возвращены функцией. Например: **p.eng\_name, p.sku**. По умолчанию возвращаются все поля (**p.\***).
- **string \$complete\_sql**. Часть SQL-запроса, которую можно пристыковать к концу основного SQL-запроса, формируемого функцией (например, при использовании пейджинга, см. класс [Pager](#)).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с описанием найденного продукта, содержащим следующие ключи:
  - **id** – идентификатор продукта;
  - все поля таблицы [proshop\\_products](#), перечисленные в параметре **\$pfields**;
  - **pos** – позиция продукта по отношению к другим продуктам указанной категории, если поиск производится строго в пределах определенной категории;
  - **tax** – величина налога для продукта в базовой валюте (если при поиске задействованы налоги);
  - **discount** – величина скидки для продукта в базовой валюте (если при поиске задействованы скидки);
  - **final\_price** – итоговая цена продукта в базовой валюте с учетом налога и скидок;

- **caption** – название продукта на заданном языке (если параметр **\$lang\_id** не равен 0);
- **short\_descr** – краткое описание продукта на заданном языке (если параметр **\$lang\_id** не равен 0);
- **alt** – альтернативный текст основного изображения продукта на заданном языке (если параметр **\$lang\_id** не равен 0);
- **if\_null\_qty** – текст на заданном языке, выводимый покупателю в карточке товара, если данного товара нет на складе (если параметр **\$lang\_id** не равен 0).

Пример:

- ```
$rules = array();
$rule = array(
    'type' => 'by_price',
    'include_tax' => Taxes::GetCount() > 0 ? 'Y' : 'N',
    'include_discount' => Discounts::GetActiveCount() > 0 ? 'Y' : 'N'
);
if (!empty($from_price)) {
    $rule['from_price'] = $from_price;
}
if (!empty($to_price)) {
    $rule['to_price'] = $to_price;
}
// ...
$rules[] = $rule;
$products = Products::Search($rules, $SHOP_LANGUAGE['id'], true, false);
foreach ($products as $prod) {
    // $prod['id'] – идентификатор продукта;
    // $prod['caption'] – название продукта;
    // ...
}
```

### **Products::SearchCount**

Назначение:

- Определение кол-ва продуктов, которые будут найдены по заданным правилам с помощью функции [Products::Search](#).

Объявление:

- `function SearchCount($rules, $lang_id, $only_visible = true, $complete_sql = "")`

Параметры:

- см. описание параметров [Products::Search](#).

Возвращаемое значение:

- Тип **int**. Кол-во найденных продуктов.

## **Products::SetCaption**

Назначение:

- Установка названия указанного продукта на указанном языке.

Объявление:

- `function SetCaption($prod_id, $caption, $lang_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **string \$caption**. Название продукта на указанном языке. Макс. 254 символа.
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetCatDiscount**

Назначение:

- Установка указанной скидки на все продукты указанной категории.

Объявление:

- `function SetCatDiscount($discount_id, $cat_id)`

Параметры:

- **int \$discount\_id**. Идентификатор скидки из таблицы [proshop\\_discounts](#) (тип скидки должен быть **products**).
- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetCatTax**

Назначение:

- Установка указанного налога на все продукты указанной категории.

Объявление:

- `function SetCatTax($tax_id, $cat_id)`

Параметры:

- **int \$tax\_id**. Идентификатор налога из таблицы [proshop\\_taxes](#).
- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetDiscount**

Назначение:



- Установка скидки на указанный продукт.

Объявление:

- `function SetDiscount($discount_id, $prod_id)`

Параметры:

- **int \$discount\_id**. Идентификатор скидки из таблицы [proshop\\_discounts](#) (тип скидки должен быть **products**).
- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetImage**

Назначение:

- Установка имени файла основного изображения указанного продукта. Файлы изображений продуктов хранятся в директории **images/products**.

Объявление:

- `function SetImage($id, $img)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **string \$img**. Имя файла основного изображения продукта. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetMainCat**

Назначение:

- Установка главной категории указанного продукта (смена у продукта главной категории). После применения данной функции необходимо пересчитать показатели категорий вызовом функции [Categories::RecalcProductsCount\(\)](#).

Объявление:

- `function SetMainCat($id, $cat_id)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$cat\_id**. Идентификатор новой главной категории продукта из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetPosition**

Назначение:

- Установка позиции указанного продукта в указанной категории относительно других продуктов этой же категории.

Объявление:

- `function SetPosition($prod_id, $cat_id, $pos)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetPrice**

Назначение:

- Установка цены указанного продукта в базовой валюте магазина.

Объявление:

- `function SetPrice($prod_id, $price)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **float \$price**. Значение цены.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetQty**

Назначение:

- Установка кол-ва доступных экземпляров указанного продукта на складе.

Объявление:

- `function SetQty($prod_id, $qty)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **string \$qty**. Кол-во доступных экземпляров. Если значение данного параметра – пустая строка, доступное кол-во считается бесконечным. Если 0 – доступных экземпляров нет.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetSKU**

Назначение:

- Установка артикула указанного продукта.

Объявление:

- `function SetSKU($prod_id, $sku)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **string \$sku**. Артикул. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetTax**

Назначение:

- Установка налога на указанный продукт.

Объявление:

- `function SetTax($tax_id, $prod_id)`

Параметры:

- **int \$tax\_id**. Идентификатор налога из таблицы [proshop\\_taxes](#).
- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetThumb**

Назначение:

- Установка имени файла уменьшенного варианта основного изображения указанного продукта. Файлы изображений продуктов хранятся в директории **images/products**.

Объявление:

- `function SetThumb($id, $thumb)`

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **string \$thumb**. Имя файла уменьшенного варианта основного изображения продукта. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetVisibility**

Назначение:

- Установка видимости указанного продукта на витрине (включение/выключение отображения продукта).

Объявление:

- `function SetVisibility($prod_id, $visible)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **bool \$visible**. Флаг видимости продукта: **true** – продукт видим, **false** – невидим.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::SetWeight**

Назначение:

- Установка веса указанного продукта (в кг).

Объявление:

- function SetWeight(\$prod\_id, \$weight)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **float \$weight**. Значение веса.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Products::Update**

Назначение:

- Обновление информации об указанном продукте в БД.

Объявление:

- function Update(\$id, \$prod)

Параметры:

- **int \$id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **array \$prod**. См. описание соответствующего параметра для функции [Products::Add](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс ProdVariants**

Предназначен для управления вариантами продуктов.

### **ProdVariants::Add**

Назначение:

- Добавление варианта указанного продукта в БД.

Объявление:

- function Add(\$prod\_id, \$weight, \$sku, \$qty, \$pos, \$options)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **float \$weight**. Вес варианта указанного продукта.
- **string \$sku**. Артикул варианта указанного продукта.

- **string \$qty**. Кол-во доступных экземпляров данного варианта продукта. Если значение данного параметра – пустая строка, доступное кол-во считается бесконечным. Если 0 – доступных экземпляров нет.
- **int \$pos**. Позиция данного варианта для сортировки вывода вариантов в пределах заданного продукта **\$prod\_id**.
- **array \$options**. Массив идентификаторов вариантов опций продукта из таблицы [proshop\\_products\\_options\\_vars](#).

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданного варианта продукта из таблицы [proshop\\_products\\_variants](#).

## **ProdVariants::DecQty**

Назначение:

- Уменьшение кол-ва доступных экземпляров на складе указанного варианта продукта.

Объявление:

- `function DecQty($variant_id, $qty = 1)`

Параметры:

- **int \$variant\_id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **int \$qty**. Значение, на которое необходимо уменьшить кол-во доступных экземпляров указанного варианта продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::Delete**

Назначение:

- Удаление указанного варианта продукта.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **ProdVariants::DeleteByProd**

Назначение:

- Удаление всех вариантов указанного продукта.

Объявление:

- `function DeleteByProd($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::DelImage**

Назначение:

- Удаление изображения указанного варианта продукта.

Объявление:

- `function DelImage($id)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::DelThumb**

Назначение:

- Удаление уменьшенного варианта изображения указанного варианта продукта.

Объявление:

- `function DelThumb($id)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::GetAll**

Назначение:

- Извлечение всех вариантов указанного продукта из БД.

Объявление:

- `function GetAll($lang_id, $prod_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#), на котором будут извлечены варианты опций продукта, привязанные к варианту продукта.
- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#), варианты которого необходимо извлечь.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого также является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_products\\_variants](#);
  - **options** – массив, каждый элемент которого является массивом, содержащим ключи:

- **option\_caption** – название опции продукта на указанном языке **\$lang\_id**;
- **var\_caption** – название варианта опции продукта на указанном языке **\$lang\_id**;
- **var\_price\_mod** - модификатор цены продукта в базовой валюте магазина или в процентах (в зависимости от значения ключа **var\_price\_mod\_type**). Может быть положительным, отрицательным или равным 0;
- **var\_price\_mod\_type** - тип модификатора цены: **abs** – абсолютный (в базовой валюте магазина), **per** – в процентах.

## **ProdVariants::GetIdByOptions**

Назначение:

- Определение идентификатора варианта продукта по набору вариантов опций продукта.

Объявление:

- `function GetIdByOptions($options)`

Параметры:

- **array \$options**. Массив идентификаторов вариантов опций продукта из таблицы [proshop\\_products\\_options\\_vars](#).

Возвращаемое значение:

- Тип **int**. Идентификатор варианта продукта (из таблицы [proshop\\_products\\_variants](#)), соответствующего данному набору вариантов опций, или 0, если ни один вариант ни одного продукта не соответствует данному набору вариантов опций продукта.

## **ProdVariants::GetOne**

Назначение:

- Извлечение информации об указанном варианте продукта из БД.

Объявление:

- `function GetOne($id)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_products\\_variants](#).

## **ProdVariants::GetQty**

Назначение:

- Определение кол-ва доступных экземпляров указанного варианта продукта на складе.

Объявление:

- `function GetQty($id)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **string**. Кол-во доступных экземпляров на складе (может быть отрицательным при возможном «отрицательном учете», а также может быть нулем). Если возвращается пустая строка, то считается, что учет данного товара не ведется и его кол-во бесконечно.

## **ProdVariants::GetSKU**

Назначение:

- Определение артикула указанного варианта продукта.

Объявление:

- function GetSKU(\$id)

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **string**. Артикул.

## **ProdVariants::GetVarsCombs**

Назначение:

- Извлечение комбинации вариантов опций продукта для указанного варианта продукта.

Объявление:

- function GetVarsCombs(\$id)

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_products\\_options\\_vars\\_combs](#).

## **ProdVariants::IncQty**

Назначение:

- Увеличение кол-ва доступных экземпляров варианта продукта на складе.

Объявление:

- function IncQty(\$variant\_id, \$qty = 1)

Параметры:

- **int \$variant\_id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **int \$qty**. Значение, на которое необходимо увеличить кол-во экземпляров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::LoadImage**

Назначение:



- Загрузка изображения варианта продукта в магазин. Изображения вариантов продуктов хранятся в директории **images/products**.

Объявление:

- `function LoadImage($id, $name, $tmp_name, $make_thumb = false, $thumb_width = 105, $thumb_height = 96)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **string \$name**. Имя загружаемого файла изображения (из `$_FILES['name']`).
- **string \$tmp\_name**. Имя временного файла загружаемого изображения (из `$_FILES['tmp_name']`).
- **bool \$make\_thumb**. Если значение данного параметра равно **true**, для заданного варианта продукта также автоматически формируется уменьшенный вариант изображения на основе загружаемого изображения.
- **int \$thumb\_width**. Ширина уменьшенного изображения варианта продукта. Данный параметр игнорируется, если **\$make\_thumb** равен **false**.
- **int \$thumb\_height**. Высота уменьшенного изображения варианта продукта. Данный параметр игнорируется, если **\$make\_thumb** равен **false**.

Возвращаемое значение:

- Тип **string/bool**. **False** – в случае ошибки. Иначе – сформированное имя файла загруженного изображения варианта продукта.

### **ProdVariants::LoadThumbnail**

Назначение:

- Загрузка уменьшенного изображения варианта продукта в магазин. Изображения вариантов продуктов хранятся в директории **images/products**.

Объявление:

- `function LoadThumbnail($id, $name, $tmp_name, $width = 105, $height = 96)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **string \$name**. Имя загружаемого файла изображения (из `$_FILES['name']`).
- **string \$tmp\_name**. Имя временного файла загружаемого изображения (из `$_FILES['tmp_name']`).
- **int \$width**. Ширина уменьшенного изображения варианта продукта.
- **int \$height**. Высота уменьшенного изображения варианта продукта.

Возвращаемое значение:

- Тип **string/bool**. **False** – в случае ошибки. Иначе – сформированное имя файла загруженного уменьшенного изображения продукта.

## **ProdVariants::MakeThumb**

Назначение:

- Генерирование уменьшенного изображения указанного варианта продукта.

Объявление:

- `function MakeThumb($id, $width, $height)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **int \$width**. Ширина уменьшенного изображения.
- **int \$height**. Высота уменьшенного изображения.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе - сформированное имя файла уменьшенного варианта изображения продукта.

## **ProdVariants::SetImage**

Назначение:

- Установка имени файла изображения указанного варианта продукта. Файлы изображений продуктов хранятся в директории **images/products**.

Объявление:

- `function SetImage($id, $img)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **string \$img**. Имя файла изображения варианта продукта. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::SetQty**

Назначение:

- Установка кол-ва доступных экземпляров указанного варианта продукта на складе.

Объявление:

- `function SetQty($id, $qty)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **string \$qty**. Кол-во доступных экземпляров. Если значение данного параметра – пустая строка, доступное кол-во считается бесконечным. Если 0 – доступных экземпляров нет.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::SetSKU**

Назначение:

- Установка артикула указанного варианта продукта.

Объявление:

- `function SetSKU($id, $sku)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **string \$sku**. Артикул. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::SetThumb**

Назначение:

- Установка имени файла уменьшенного изображения указанного варианта продукта. Файлы изображений продуктов хранятся в директории **images/products**.

Объявление:

- `function SetThumb($id, $thumb)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **string \$thumb**. Имя файла уменьшенного изображения. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::SetWeight**

Назначение:

- Установка веса указанного варианта продукта (в кг).

Объявление:

- `function SetWeight($id, $weight)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).
- **float \$weight**. Значение веса.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **ProdVariants::Update**

Назначение:

- Обновление информации об указанном варианте продукта в БД.

Объявление:

- `function Update($id, $weight, $sku, $qty, $pos)`

Параметры:

- **int \$id**. Идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#).

- **float \$weight**. Вес варианта продукта (в кг).
- **string \$sku**. Артикул варианта продукта. Макс. 254 символа.
- **string \$qty**. Кол-во доступных экземпляров. Если значение данного параметра – пустая строка, доступное кол-во считается бесконечным. Если 0 – доступных экземпляров нет.
- **int \$pos**. Позиция данного варианта для сортировки вывода вариантов в пределах продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс **RecommendedProds**

Предназначен для управления рекомендуемыми продуктами (продуктами, которые могут быть рекомендованы к продаже совместно с другими продуктами). Рекомендуемый продукт (назовем **продукт2**) отображается в карточке продукта (назовем **продукт1**). При этом **продукт1** не отображается в карточке **продукта2**.

### **RecommendedProds : :Add**

Назначение:

- Привязка рекомендуемого продукта к указанному продукту.

Объявление:

- function Add(\$prod\_id, \$rprod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$rprod\_id**. Идентификатор рекомендуемого продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **RecommendedProds : :Delete**

Назначение:

- Удаление указанного рекомендуемого продукта из списка рекомендуемых в БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор рекомендуемого продукта из таблицы [proshop\\_products\\_recommended](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **RecommendedProds : :GetAll**

Назначение:

- Извлечение списка рекомендованных продуктов для указанного продукта.

Объявление:

- `function GetAll($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_products\\_recommended](#).

## **RecommendedProds :: SetPos**

Назначение:

- Установка позиции указанного рекомендуемого товара относительно других рекомендуемых товаров.

Объявление:

- `function SetPos($rprod_id, $pos)`

Параметры:

- **int \$rprod\_id**. Идентификатор рекомендуемого товара из таблицы [proshop\\_products\\_recommended](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс RelatedProds**

Предназначен для управления родственными продуктами. Родственный продукт (назовем **продукт1**) отображается в карточке продукта (назовем **продукт2**) в соответствующем разделе, и наоборот, **продукт2** отображается в карточке **продукта1**.

### **RelatedProds :: Add**

Назначение:

- Привязка родственного продукта к указанному продукту.

Объявление:

- `function Add($prod_id, $rprod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$rprod\_id**. Идентификатор родственного продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **RelatedProds::Delete**

Назначение:

- Удаление указанного родственного продукта из списка родственников в БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор родственного продукта из таблицы [proshop\\_products\\_related](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **RelatedProds::GetAll**

Назначение:

- Извлечение списка родственников продуктов для указанного продукта.

Объявление:

- `function GetAll($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_products\\_related](#).

## **RelatedProds::SetPos**

Назначение:

- Установка позиции указанного родственного товара относительно других родственников товаров.

Объявление:

- `function SetPos($rprod_id, $pos)`

Параметры:

- **int \$rprod\_id**. Идентификатор рекомендуемого товара из таблицы [proshop\\_products\\_related](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс RememberMe**

Предназначен для хранения информации о пользователе, необходимой для автоматического захода (логина) в магазин, в cookie.

## **RememberMe::ClearAdmin**

Назначение:

- Удаление cookie, связанного с хранением информации об администраторе магазина.

Объявление:

- `function ClearAdmin()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **RememberMe::ClearCustomer**

Назначение:

- Удаление cookie, связанного с хранением информации о зарегистрированном покупателе (пользователе) магазина.

Объявление:

- `function ClearCustomer()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

## **RememberMe::GetAdminAuth**

Назначение:

- Извлечение информации из cookie, установленном ранее с помощью функции [RememberMe::SetAdminAuth](#).

Объявление:

- `function GetAdminAuth()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Первый элемент массива является идентификатором администратора в таблице [proshop\\_users](#). Второй элемент – хэш-значением пароля.

## **RememberMe::GetAdminLang**

Назначение:

- Определение идентификатора языка административной части магазина, сохраненного ранее в cookie с помощью [RememberMe::SetAdminLang](#).

Объявление:

- `function GetAdminLang()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Идентификатор языка из таблицы [proshop\\_languages](#).

## **RememberMe::GetCustomerAuth**

Назначение:

- Извлечение информации из cookie, установленном ранее с помощью функции [RememberMe::SetCustomerAuth](#).

Объявление:

- `function GetCustomerAuth()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Первый элемент массива является идентификатором покупателя (пользователя) в таблице [proshop\\_users](#). Второй элемент – хэш-значением пароля.

## **RememberMe::IsAdminAuth**

Назначение:

- Определение, установлен ли на клиентском компьютере cookie, сохраненный ранее с помощью вызова функции [RememberMe::SetAdminAuth](#).

Объявление:

- `function IsAdminAuth()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **RememberMe::IsAdminLang**

Назначение:

- Определение, установлен ли на клиентском компьютере cookie, сохраненный ранее с помощью вызова функции [RememberMe::SetAdminLang](#).

Объявление:

- `function IsAdminLang()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.



## **RememberMe::IsCustomerAuth**

Назначение:

- Определение, установлен ли на клиентском компьютере cookie, сохраненный ранее с помощью вызова функции [RememberMe::SetCustomerAuth](#).

Объявление:

- `function IsCustomerAuth()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **RememberMe::SetAdminAuth**

Назначение:

- Сохранение cookie с информацией для автоматического входа администратора магазина.

Объявление:

- `function SetAdminAuth($user_id, $hash)`

Параметры:

- **int \$user\_id**. Идентификатор администратора из таблицы [proshop\\_users](#).
- **string \$hash**. Хэш-значение пароля администратора.

Возвращаемое значение:

- Функция ничего не возвращает.

## **RememberMe::SetAdminLang**

Назначение:

- Сохранение идентификатора языка админки в cookie на клиентском компьютере администратора.

Объявление:

- `function SetAdminLang($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## **RememberMe::SetCustomerAuth**

Назначение:

- Сохранение cookie с информацией для автоматического входа (логина) зарегистрированного покупателя в магазин.

Объявление:

- `function SetCustomerAuth($user_id, $hash)`

Параметры:

- **int \$user\_id**. Идентификатор покупателя из таблицы [proshop\\_users](#).
- **string \$hash**. Хэш-значение пароля покупателя.

Возвращаемое значение:

- Функция ничего не возвращает.

## Класс Request

Предназначен для хранения временной информации заданного модуля в сессии.

Используется вместо «голового» **\$\_SESSION**.

### **\$req->DelAllParams**

Назначение:

- Удаление всех параметров текущего объекта из сессии (очистка памяти).

Объявление:

- `function DelAllParams()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Функция ничего не возвращает.

Пример:

- ```
$req = new Request($module, 'list');  
$req->DelAllParams();
```

### **\$req->DelParamValue**

Назначение:

- Удаление указанного параметра текущего объекта из сессии (очистка памяти).

Объявление:

- `function DelParamValue($name)`

Параметры:

- **string \$name**. Название параметра.

Возвращаемое значение:

- Функция ничего не возвращает.

Пример:

- ```
$req = new Request($module, 'list');  
$req->DelParamValue('name');
```

### **\$req->GetParamValue**

Назначение:

- Извлечение значения указанного параметра текущего объекта из сессии.

Объявление:

- `function GetParamValue($name)`

Параметры:

- **string \$name**. Название параметра.

Возвращаемое значение:

- Тип **mixed**. Значение параметра.

Пример:

- ```
$req = new Request($module, 'list');
$name = $req->GetParamValue('name');
```

## **`$req->ParamValueExists`**

Назначение:

- Определение, задано ли значение указанного параметра текущего объекта в сессии.

Объявление:

- `function ParamValueExists($name)`

Параметры:

- **string \$name**. Название параметра.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

Пример:

- ```
$req = new Request($module, 'list');
if ($req->ParamValueExists('name')) {
    // ...
}
```

## **`$req->Request`**

Назначение:

- Конструктор класса.

Объявление:

- `function Request($module, $act, $id = 0)`

Параметры:

- **string \$module**. Название модуля, для которого создается экземпляр класса.
- **string \$act**. Действие модуля, для которого создается экземпляр класса.
- **int \$id**. Дополнительный идентификатор для указанного действия (для уникальности). Может быть равен нулю.

Пример:

- ```
$req = new Request($module, 'list');
```

## `$req->SetParamValue`

Назначение:

- Установка значения указанного параметра текущего объекта в сессии.

Объявление:

- `function SetParamValue($name, $value)`

Параметры:

- **string \$name**. Название параметра.
- **mixed \$value**. Значение параметра.

Возвращаемое значение:

- Функция ничего не возвращает.

Пример:

- ```
$req = new Request($module, 'list');  
$req->SetParamValue('name', 'value');
```

## Класс `Reviews`

Предназначен для управления отзывами о магазине (гостевой книгой).

### `Reviews::Add`

Назначение:

- Добавление нового отзыва в таблицу [proshop\\_reviews](#).

Объявление:

- `function Add($author, $email, $comment, $lang_id)`

Параметры:

- **string \$author**. Имя автора отзыва. Макс. 254 символа.
- **string \$email**. Адрес электронной почты автора отзыва. Макс. 254 символа.
- **string \$comment**. Текст комментария. Макс. 65533 символа.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), который выбран в магазине в момент сохранения отзыва.

Возвращаемое значение:

- Тип **int/bool**. `False` – в случае ошибки. Иначе – идентификатор вставленной в таблицу [proshop\\_reviews](#) записи.

### `Reviews::Approve`

Назначение:

- Одобрение указанного отзыва (отзыв становится видимым на витрине магазина).

Объявление:

- `function Approve($id)`

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_reviews](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Reviews::Decline**

Назначение:

- Операция, обратная одобрению отзыва (отзыв становится невидимым на витрине магазина).

Объявление:

- function Decline(\$id)

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_reviews](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Reviews::Delete**

Назначение:

- Удаление указанного отзыва из таблицы [proshop\\_reviews](#).

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_reviews](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Reviews::GetAll**

Назначение:

- Извлечение всех отзывов на указанном языке из таблицы [proshop\\_reviews](#).

Объявление:

- function GetAll(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_reviews](#).

## **Reviews::GetAllApproved**

Назначение:

- Извлечение всех одобренных отзывов на указанном языке из таблицы [proshop\\_reviews](#).

Объявление:

- function GetAllApproved(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_reviews](#).

## **Reviews::GetAllInapproved**

Назначение:

- Извлечение всех неодобренных отзывов на указанном языке из таблицы [proshop\\_reviews](#).

Объявление:

- function GetAllInapproved(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_reviews](#).

## **Reviews::GetInapprovedCount**

Назначение:

- Определение кол-ва неодобренных отзывов на указанном языке в таблице [proshop\\_reviews](#).

Объявление:

- function GetInapprovedCount(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **int**. Кол-во неодобренных отзывов.

## **Reviews::SetComment**

Назначение:

- Установка текста комментария указанного отзыва.

Объявление:

- function SetComment(\$id, \$comment)

Параметры:

- **int \$id**. Идентификатор отзыва из таблицы [proshop\\_reviews](#).
- **string \$comment**. Текст комментария. Макс. 65533 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс RSS

Предназначен для формирования и вывода XML-содержимого RSS-канала.

### `$rss->AddItem`

Назначение:

- Добавление элемента в вывод RSS.

Объявление:

- `function AddItem($url, $title, $description, $date, $author = "", $category = "")`

Параметры:

- **string \$url**. Адрес страницы элемента.
- **string \$title**. Заголовок элемента.
- **string \$description**. Текст описания элемента.
- **string \$date**. Дата создания элемента.
- **string \$author**. Автор контента.
- **string \$category**. Категория контента.

Возвращаемое значение:

- Функция ничего не возвращает.

### `$rss->Get`

Назначение:

- Формирование XML-содержимого RSS-канала, подготовленного ранее с помощью вызовов методов данного класса.

Объявление:

- `function Get()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. XML-содержимое канала.

### `$rss->GetChannelDescription`

Назначение:

- Определение описания RSS-канала, установленного ранее.

Объявление:

- `function GetChannelDescription()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Описание RSS-канала.

## **`$rss->GetChannelTitle`**

Назначение:

- Определение названия RSS-канала, установленного ранее.

Объявление:

- `function GetChannelTitle()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Название RSS-канала.

## **`$rss->GetChannelURL`**

Назначение:

- Определение URL-адреса RSS-канала, установленного ранее.

Объявление:

- `function GetChannelURL()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. URL-адрес RSS-канала.

## **`$rss->GetDate`**

Назначение:

- Определение даты формирования RSS-канала, установленной ранее.

Объявление:

- `function GetDate()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Дата формирования RSS.

## **`$rss->GetImage`**

Назначение:

- Определение информации об изображении RSS-канала, установленном ранее с помощью [\\$rss->SetImage](#).

Объявление:

- `function GetImage()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:



- Тип **array**. Массив, описывающий изображение и состоящий из следующих ключей:
  - **path** – URL-адрес файла изображения;
  - **title** – название изображения;
  - **url** – URL-адрес ссылки изображения (куда оно ссылается).

## **\$rss->GetItems**

Назначение:

- Извлечение массива элементов RSS-канала, установленных ранее путем вызова [\\$rss->AddItem](#).

Объявление:

- `function GetItems()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив элементов, каждый из которых является массивом, состоящим из следующих ключей:
  - **url** - адрес страницы элемента;
  - **title** - заголовок элемента;
  - **description** - текст описания элемента;
  - **date** - дата создания элемента;
  - **author** - автор контента;
  - **category** - категория контента.

## **\$rss->GetLang**

Назначение:

- Определение двубуквенного обозначения языка RSS-контента, установленного ранее путем вызова [\\$rss->SetLang](#) или вызова конструктора класса.

Объявление:

- `function GetLang()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Двубуквенное обозначение языка.

## **\$rss->GetRSSVersion**

Назначение:

- Определение версии RSS.

Объявление:

- `function GetRSSVersion()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Всегда возвращается **2.0**.

## **`$rss->GetXMLVersion`**

Назначение:

- Определение версии XML.

Объявление:

- `function GetXMLVersion()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Всегда возвращается **1.0**.

## **`$rss->Publish`**

Назначение:

- Вывод XML-контента RSS-канала в браузер.

Объявление:

- `function Publish($content = "")`

Параметры:

- **string \$content**. XML-контент, который необходимо вывести. Если не задан, используется возвращаемое значение функции [\\$rss->Get](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$rss->RSS`**

Назначение:

- Конструктор класса.

Объявление:

- `function RSS($lang, $url, $title, $description, $date = null)`

Параметры:

- **string \$lang**. Двубуквенное обозначение языка формируемого RSS-контента.
- **string \$url**. Путь к RSS-каналу.
- **string \$title**. Заголовок RSS-канала.
- **string \$description**. Описание RSS-канала.
- **string \$date**. Дата формирования RSS-канала. Если не задана, используется текущая дата сервера.

## **`$rss->SetChannelInfo`**

Назначение:

- Установка информации о формируемом RSS-канале.

Объявление:

- `function SetChannelInfo($url, $title, $description, $date = null)`

Параметры:

- См. описание соответствующих параметров конструктора [\\$rss->RSS](#).

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$rss->SetImage`**

Назначение:

- Установка информации об изображении RSS-канала.

Объявление:

- `function SetImage($path, $title, $url)`

Параметры:

- **string \$path**. URL-адрес файла изображения;
- **string \$title**. Название изображения;
- **string \$url**. URL-адрес ссылки изображения (куда оно ссылается).

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$rss->SetLang`**

Назначение:

- Установка двубуквенного обозначения языка контента RSS-канала.

Объявление:

- `function SetLang($lang)`

Параметры:

- **string \$lang**. Двубуквенное обозначение языка (две английские буквы нижнего регистра).

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс RusPost**

Предназначен для расчета стоимости доставки заказа через Почту России.

### **`RusPost::GetAllOptions`**

Назначение:

- Извлечение всех настроек службы доставки Почта России из таблицы [proshop\\_ruspost\\_options](#).

Объявление:

- function GetAllOptions()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False - в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_ruspost\\_options](#).

## **RusPost::GetAllRegions**

Назначение:

- Извлечение информации о магистральных поясах РФ из таблицы [proshop\\_ruspost\\_regions](#).

Объявление:

- function GetAllRegions()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе - массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_ruspost\\_regions](#).

## **RusPost::GetOption**

Назначение:

- Извлечение значения указанной опции службы Почты России из таблицы [proshop\\_ruspost\\_options](#).

Объявление:

- function GetOption(\$name)

Параметры:

- **string \$name**. Название опции из таблицы [proshop\\_ruspost\\_options](#).

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – значение указанной опции.

## **RusPost::GetRegionByNum**

Назначение:

- Извлечение информации о заданном магистральном поясе РФ из таблицы [proshop\\_ruspost\\_regions](#).

Объявление:

- function GetRegionByNum(\$num)

Параметры:

- **int \$num.** Номер магистрального пояса.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив с ключами, совпадающими с полями таблицы [proshop\\_ruspost\\_regions](#).

## **RusPost::SetAvailAvia**

Назначение:

- Установка флага авиадоставки для указанного региона России.

Объявление:

- function SetAvailAvia(\$state\_id, \$avail)

Параметры:

- **string \$state\_id.** Идентификатор области РФ из таблицы [proshop\\_states](#).
- **char \$avail.** Если данный параметр равен Y, при расчете стоимости доставки будет предполагаться, что доставка в заданную область России будет осуществлена воздушным способом.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **RusPost::SetOption**

Назначение:

- Установка значения указанной опции Почты России в таблице [proshop\\_ruspost\\_options](#).

Объявление:

- function SetOption(\$name, \$value)

Параметры:

- **string \$name.** Название опции из таблицы [proshop\\_ruspost\\_options](#).
- **string \$value.** Значение опции. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **RusPost::SetRegion**

Назначение:

- Привязка заданной области РФ к заданному магистральному поясу.

Объявление:

- function SetRegion(\$state\_id, \$region\_id)

Параметры:

- **int \$state\_id.** Идентификатор области из таблицы [proshop\\_states](#).
- **int \$region\_id.** Номер магистрального пояса из таблицы [proshop\\_ruspost\\_regions](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **RusPost::SetRegionAddCost**

Назначение:

- Установка размера дополнительной платы за каждые полные/неполные 500 г. массы посылки, если масса заказа превышает 500 г., в базовой валюте магазина в таблице [proshop\\_ruspost\\_regions](#) для указанного магистрального пояса.

Объявление:

- `function SetRegionAddCost($region, $add_cost)`

Параметры:

- **int \$region**. Номер магистрального пояса в таблице [proshop\\_ruspost\\_regions](#).
- **float \$add\_cost**. Значение суммы дополнительной платы в базовой валюте магазина.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **RusPost::SetRegionCost**

Назначение:

- Установка стоимости пересылки заказа для указанного магистрального пояса, если масса заказа не превышает 500 г., в базовой валюте магазина в таблице [proshop\\_ruspost\\_regions](#).

Объявление:

- `function SetRegionCost($region, $cost)`

Параметры:

- **int \$region**. Номер магистрального пояса в таблице [proshop\\_ruspost\\_regions](#).
- **float \$cost**. Значение стоимости пересылки в базовой валюте магазина.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Sberbank**

Предназначен для генерирования квитанции Сбербанка для оплаты в формате PDF.

### **Sberbank::GenerateReceipt**

Назначение:

- Генерирование квитанции в формате PDF.

Объявление:

- `function GenerateReceipt($info, $name = "")`

Параметры:

- **array \$info**. Массив с информацией для генерирования квитанции. Должен содержать следующие ключи:
  - **pay\_recipient\_name** – наименование получателя платежа;

- **pay\_recipient\_inn** – ИНН получателя платежа;
- **pay\_recipient\_acc** – номер счета получателя платежа;
- **pay\_recipient\_bank** – наименование банка получателя платежа;
- **pay\_recipient\_bik** – БИК;
- **pay\_recipient\_loro** – к/с банка получателя платежа;
- **pay\_name** – наименование платежа;
- **payer\_name** – ФИО плательщика;
- **payer\_address** – адрес плательщика;
- **pay\_summ** – сумма платежа в формате **руб.коп.**
- **string \$name**. Имя файла генерируемой квитанции. Если задано, сгенерированный файл выбрасывается в браузер. Иначе – его содержимое возвращается функцией.

Возвращаемое значение:

- Если параметр **\$name** не задан, возвращается содержимое сгенерированного PDF-файла квитанции. Иначе – ничего не возвращается.

## Класс Security

Предназначен для фильтрации числовых и текстовых данных перед их вставкой в SQL-запрос и после их извлечения с помощью SELECT-запроса.

### Security::ToInt

Назначение:

- Фильтрация целочисленного значения указанной переменной с помощью **intval** перед его вставкой в SQL-запрос.

Объявление:

- `function ToInt($var)`

Параметры:

- **array/int \$var**. Данный параметр может быть как целым числом, так и массивом целых чисел. Если задан массив, фильтруется каждое его значение.

Возвращаемое значение:

- Тип **array/int**. Тип возвращаемого значения зависит от типа параметра **\$var**.

Пример:

- `$id = Security::ToInt($id);`  
`$sql = 'SELECT * FROM '.$TABLES['products'].' WHERE id = '.$id;`

### Security::ToSqlStr

Назначение:

- Фильтрация строки (добавление слешей и вырезание HTML-тэгов при необходимости) перед ее вставкой в SQL-запрос.

Объявление:

- `function ToSqlStr($str, $type = "")`

Параметры:

- **array/string \$str**. Данный параметр может быть как строкой, так и массивом строк. Если задан массив, фильтруется каждое его значение.
- **string \$type**. Если данный параметр равен **html**, из параметра **\$str** не вырезаются HTML-теги.

Возвращаемое значение:

- Тип **array/string**. Тип возвращаемого значения зависит от типа параметра **\$str**.

Пример:

- ```
$caption = Security::ToSqlStr($caption);  
$sql = 'UPDATE '.$TABLES['products_texts'].' SET caption = "'.$caption.'" WHERE id = '.$id;
```

## **Security::ToStr**

Назначение:

- Фильтрация строки после ее извлечения из БД с помощью SELECT-запроса.

Объявление:

- `function ToStr($str, $type = "")`

Параметры:

- **array/string \$str**. Данный параметр может быть как строкой, так и массивом строк. Если задан массив, фильтруется каждое его значение.
- **string \$type**. Если данный параметр равен **html**, из параметра **\$str** не вырезаются HTML-теги.

Возвращаемое значение:

- Тип **array/string**. Тип возвращаемого значения зависит от типа параметра **\$str**.

Пример:

- ```
$sql = 'SELECT caption FROM '.$TABLES['products_texts'].' WHERE id = '.$id;  
$row = $DB->GetRow($sql);  
$caption = Security::ToStr($row['caption']);
```

## **Класс Slider**

Предназначен для управления слайдером продуктов.

### **Slider::Add**

Назначение:

- Добавление продукта в слайдер.

Объявление:

- `function Add($prod_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).



Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Slider::Delete**

Назначение:

- Удаление продукта из слайдера.

Объявление:

- function Delete(\$prod\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Slider::GetAll**

Назначение:

- Извлечение списка продуктов слайдера в порядке их сортировки, определенном полем pos таблицы [proshop\\_slider](#).

Объявление:

- function GetAll(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#), на котором необходимо извлечь названия продуктов.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - **id** – идентификатор продукта;
  - **eng\_name** – имя в URL продукта (для modRewrite);
  - **thumb** – см. описание одноименного поля таблицы [proshop\\_products](#);
  - **alt** – альтернативный текст изображения продукта на языке **\$lang\_id**;
  - **caption** – название продукта на языке **\$lang\_id**.

## **Slider::SetPos**

Назначение:

- Установка позиции отображения заданного продукта относительно других продуктов в слайдере.

Объявление:

- function SetPos(\$prod\_id, \$pos)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).

- **int \$pos**. Номер позиции указанного продукта.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс Shippings

Предназначен для управления службами доставок, способами доставок заказа, а также для управления привязками способов доставок к способам оплаты (для возможности выбора только определенных способов оплаты при определенном выбранном способе доставки).

### **Shippings::CustomGetAll**

Назначение:

- Извлечение всех способов доставок типа **custom** (таких, как Почта России, EMS) из БД.

Объявление:

- `function CustomGetAll($lang_id, $active_only = true)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия способов доставок.
- **bool \$active\_only**. Если данный параметр равен **true**, извлекаются только активные методы доставки типа **custom**. Иначе – извлекаются все методы доставки типа **custom**.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - все поля таблицы [proshop\\_cshippings](#);
  - **caption** – название службы доставки на указанном языке. Если название для службы доставки не найдено, используется служебное название доставки из таблицы [proshop\\_cshippings](#).

### **Shippings::CustomGetOne**

Назначение:

- Извлечение информации о доставке типа **custom** (такой, как Почта России) из БД.

Объявление:

- `function CustomGetOne($id, $lang_id = 0)`

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_cshippings](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название указанной доставки. Если данный параметр равен 0, название не извлекается.

Возвращаемое значение:

- Тип **array/bool**. **False** – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_cshippings](#);
  - **caption** – название службы доставки на указанном языке. Если название для службы доставки не найдено, используется служебное название доставки из таблицы [proshop\\_cshippings](#).

## **Shippings::CustomGetOneByName**

Назначение:

- Извлечение информации о доставке типа **custom** (такой, как Почта России) из БД по ее служебному имени.

Объявление:

- `function CustomGetOneByName($name, $lang_id = 0)`

Параметры:

- **string \$name**. Служебное имя доставки из таблицы [proshop\\_cshippings](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название указанной доставки. Если данный параметр равен 0, название не извлекается.

Возвращаемое значение:

- См. возвращаемое значение функции [Shippings::GetOne](#).

## **Shippings::CustomSetDeliveryTime**

Назначение:

- Установка диапазона времени, в течение которого осуществляется доставка типа **custom** (такая, как Почта России), для указанной доставки.

Объявление:

- `function CustomSetDeliveryTime($id, $delivery_time1, $delivery_time2)`

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_cshippings](#).
- **int \$delivery\_time1**. Минимальное кол-во дней, в течение которого осуществляется доставка.
- **int \$delivery\_time2**. Максимальное кол-во дней, в течение которого осуществляется доставка.

Возвращаемое значение:

- Тип **bool**. **True/false** – в зависимости от успешности операции.

## **Shippings::CustomSetIsActive**

Назначение:

- Установка флага активности указанного метода доставки.

Объявление:

- function CustomSetIsActive(\$id, \$is\_active)

Параметры:

- **int \$id**. Идентификатор метода доставки из таблицы [proshop\\_cshippings](#).
- **bool \$is\_active**. Флаг активности: **true** – метод доставки активен, **false** – неактивен.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::CustomSetTexts**

Назначение:

- Установка указанного названия доставки типа **custom** (такой, как Почта России) на указанном языке.

Объявление:

- function CustomSetTexts(\$id, \$lang\_id, \$caption)

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_cshippings](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **string \$caption**. Название доставки на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::PaymentBindAdd**

Назначение:

- Привязка указанных способа оплаты и доставки друг к другу.

Объявление:

- function PaymentBindAdd(\$shipping\_id, \$shipping\_type, \$payment\_id)

Параметры:

- **int \$shipping\_id**. Идентификатор способа доставки. К какой таблице БД относится данный идентификатор, зависит от значения параметра **\$shipping\_type**.
- **char \$shipping\_type**. Тип способа доставки. Может принимать одно из следующих значений:
  - **r** – realtime-доставки (такие, как FedEx, InterShipper и т.д.) – берутся из таблицы [proshop\\_rtshippings](#);
  - **s** – статические способы доставки – берутся из таблицы [proshop\\_shippings](#);
  - **c** – custom-доставки (такие, как Почта России, EMS) – берутся из таблицы [proshop\\_cshippings](#).
- **int \$payment\_id**. Идентификатор способа оплаты из таблицы [proshop\\_payments](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::PaymentBindDelete**

Назначение:

- Удаление указанной связи способа доставки со способом оплаты из таблицы [proshop\\_shippings\\_payments](#).

Объявление:

- function PaymentBindDelete(\$id)

Параметры:

- **int \$id**. Идентификатор записи из таблицы [proshop\\_shippings\\_payments](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::PaymentBindGetAll**

Назначение:

- Извлечение всех связей способов доставки и способов оплаты (всех записей из таблицы [proshop\\_shippings\\_payments](#)).

Объявление:

- function PaymentBindGetAll()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_shippings\\_payments](#).

## **Shippings::RTGetAll**

Назначение:

- Извлечение из БД списка служб доставок, которые имеют возможность подсчета стоимости доставки в режиме реального времени.

Объявление:

- function RTGetAll()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - все поля таблицы [proshop\\_rtshippings\\_groups](#);
  - **services** – массив, состоящий из элементов-массивов с ключами, совпадающими с полями таблицы [proshop\\_rtshippings](#).

## **Shippings::RTGetAllActive**

Назначение:

- Извлечение всех включенных (активных, видимых на витрине) способов доставки, которые имеют возможность расчета стоимости доставки в режиме реального времени, из таблицы [proshop\\_rtshippings](#).

Объявление:

- `function RTGetAllActive()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_rtshippings](#).

## **Shippings::RTGetCaption**

Назначение:

- Извлечение названия указанного способа доставки типа `realtime` из таблицы [proshop\\_rtshippings](#).

Объявление:

- `function RTGetCaption($id)`

Параметры:

- **int \$id**. Идентификатор способа доставки из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **string**. Название способа доставки.

## **Shippings::RTGetGroup**

Назначение:

- Извлечение информации о группе способов доставки, которые имеют возможность подсчета стоимости доставки в режиме реального времени, из БД.

Объявление:

- `function RTGetGroup($id)`

Параметры:

- **int \$id**. Идентификатор группы из таблицы [proshop\\_rtshippings\\_groups](#).

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе - массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - все поля таблицы [proshop\\_rtshippings\\_groups](#);
  - **settings** – массив настроек группы, который состоит из массивов с ключами, совпадающими с полями таблицы [proshop\\_rtshippings\\_groups\\_settings](#).

## **Shippings::RTGetGroupSettingByName**

Назначение:

- Извлечение значения указанной настройки указанной группы realtime-доставок (доставок, которые имеют возможность подсчета стоимости доставки в режиме реального времени) из таблицы [proshop\\_rtshippings\\_groups\\_settings](#).

Объявление:

- function RTGetGroupSettingByName(\$group\_id, \$name)

Параметры:

- **int \$group\_id**. Идентификатор группы realtime-доставок из таблицы [proshop\\_rtshippings\\_groups](#).
- **string \$name**. Название настройки из таблицы [proshop\\_rtshippings\\_groups\\_settings](#).

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – значение указанной настройки.

## **Shippings::RTGetOne**

Назначение:

- Извлечение информации об указанной realtime-доставке из таблицы [proshop\\_rtshippings](#).

Объявление:

- function RTGetOne(\$id)

Параметры:

- **int \$id**. Идентификатор realtime-доставки из таблицы [proshop\\_rtshippings](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив с ключами, совпадающими с полями таблицы [proshop\\_rtshippings](#).

## **Shippings::RTUpdate**

Назначение:

- Обновление информации об указанной realtime-доставке в таблице [proshop\\_rtshippings](#).

Объявление:

- function RTUpdate(\$id, \$is\_active, \$delivery\_time1, \$delivery\_time2)

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_rtshippings](#).
- **char \$is\_active**. Флаг доступности доставки на витрине (**Y** – доступна, **N** - недоступна).
- **int \$delivery\_time1**. Минимально возможное время доставки в днях.
- **int \$delivery\_time2**. Максимально возможное время доставки в днях.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::RTUpdateSetting**

Назначение:

- Обновление значения указанной настройки realtime-доставки в таблице [proshop\\_rtshippings\\_groups\\_settings](#).

Объявление:

- `function RTUpdateSetting($id, $value)`

Параметры:

- **int \$id**. Идентификатор настройки в таблице [proshop\\_rtshippings\\_groups\\_settings](#).
- **string \$value**. Новое значение настройки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::StaticAddRoot**

Назначение:

- Добавление базовой информации о статической доставке (доставке с фиксированной стоимостью, которая может зависеть от суммы заказа) в таблицу [proshop\\_shippings](#).

Объявление:

- `function StaticAddRoot($is_active, $delivery_time1, $delivery_time2, $pos)`

Параметры:

- **char \$is\_active**. Флаг доступности доставки на витрине магазина (**Y** – доступна, **N** - недоступна).
- **int \$delivery\_time1**. Минимально возможное время доставки, в днях.
- **int \$delivery\_time2**. Максимально возможное время доставки, в днях.
- **int \$pos**. Позиция для сортировки вывода статических доставок.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданной доставки в таблице [proshop\\_shippings](#).

## **Shippings::StaticAddTexts**

Назначение:

- Добавление названия указанной статической доставки для указанного языка в таблицу [proshop\\_shippings\\_texts](#). Статическая доставка – это доставка с фиксированной стоимостью, которая может зависеть от суммы заказа.

Объявление:

- `function StaticAddTexts($id, $caption, $lang_id)`

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_shippings](#).
- **string \$caption**. Название доставки. Макс. 254 символа.



- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого добавляется название.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::StaticDelete**

Назначение:

- Удаление указанной статической доставки из БД. Статическая доставка – это доставка с фиксированной стоимостью, которая может зависеть от суммы заказа.

Объявление:

- `function StaticDelete($id)`

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_shippings](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Shippings::StaticGetAll**

Назначение:

- Извлечение всех статических доставок из БД. Статическая доставка – это доставка с фиксированной стоимостью, которая может зависеть от суммы заказа.

Объявление:

- `function StaticGetAll($lang_id, $active_only = false)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором необходимо извлечь названия доставок.
- **bool \$active\_only**. Если значение данного параметра равно **true**, из таблицы [proshop\\_shippings](#) извлекаются только активные (доступные на витрине) способы доставки.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_shippings](#);
  - **caption** – название доставки на указанном языке.

## **Shippings::StaticGetBands**

Назначение:

- Извлечение всех записей из таблицы [proshop\\_shippings\\_costs](#) для заданной доставки по возрастанию значений поля **low\_band** этой таблицы.

Объявление:

- `function StaticGetBands($id)`

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_shippings](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_shippings\\_costs](#) (**low\_band**, **cost**, **per**).

## **Shippings::StaticGetCaption**

Назначение:

- Извлечение названия указанного статического способа доставки для заданного языка из БД.

Объявление:

- `function StaticGetCaption($id, $lang_id)`

Параметры:

- **int \$id**. Идентификатор способа доставки из таблицы [proshop\\_shippings](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string**. Название способа доставки.

## **Shippings::StaticGetCost**

Назначение:

- Определение стоимости указанной доставки при заданной сумме заказа.

Объявление:

- `function StaticGetCost($id, $sum, $weight = null)`

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_shippings](#).
- **float \$sum**. Сумма заказа в базовой валюте.
- **float \$weight**. Вес заказа в килограммах. Если данный параметр равен **null**, то расчет стоимости доставки ведется на основе суммы заказа. Иначе – на основе заданного веса.

Возвращаемое значение:

- Тип **float**. Значение стоимости доставки в базовой валюте. Если возвращенное значение меньше нуля, то считается, что стоимость доставки для заданных условий определить не удалось.

## **Shippings::StaticGetOne**

Назначение:

- Извлечение информации о статической доставке из БД. Статическая доставка – это доставка с фиксированной стоимостью, которая может зависеть от суммы заказа.

Объявление:

- `function StaticGetOne($id, $lang_id = 0)`

Параметры:

- **int \$id.** Идентификатор доставки из таблицы [proshop\\_shippings](#).
- **int \$lang\_id.** Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором необходимо извлечь название доставки. Если данный параметр равен 0, название доставки не извлекается.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_shippings](#);
  - **caption** – название доставки на указанном языке.

## **Shippings::StaticSetBands**

Назначение:

- Установка новых значений диапазонов стоимости доставки в таблице [proshop\\_shippings\\_costs](#) для заданной доставки.

Объявление:

- `function StaticSetBands($id, $bands)`

Параметры:

- **int \$id.** Идентификатор доставки из таблицы [proshop\\_shippings](#).
- **array \$bands.** Массив, содержащий описание диапазонов. Должен состоять из массивов, содержащих следующие ключи:
  - **low** – нижняя граница диапазона суммы заказа в единицах базовой валюты, либо диапазона веса заказа в килограммах;
  - **cost** – стоимость доставки заказа, сумма (вес) которого превышает значение **low**, в единицах базовой валюты (в килограммах);
  - **per** – стоимость доставки заказа, сумма (вес) которого превышает значение **low**, в процентах от суммы заказа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::StaticSetTexts**

Назначение:

- Установка названия указанной статической доставки для указанного языка в таблице [proshop\\_shippings\\_texts](#). Статическая доставка – это доставка с фиксированной стоимостью, которая может зависеть от суммы заказа.

Объявление:

- `function StaticSetTexts($id, $caption, $lang_id)`

Параметры:

- **int \$id.** Идентификатор доставки из таблицы [proshop\\_shippings](#).
- **string \$caption.** Название доставки. Макс. 254 символа.

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого устанавливается название доставки.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Shippings::StaticUpdate**

Назначение:

- Обновление базовой информации указанной статической доставки в таблице [proshop\\_shippings](#). Статическая доставка – это доставка с фиксированной стоимостью, которая может зависеть от суммы заказа.

Объявление:

- function StaticUpdate(\$id, \$is\_active, \$delivery\_time1, \$delivery\_time2, \$pos)

Параметры:

- **int \$id**. Идентификатор доставки из таблицы [proshop\\_shippings](#).
- См. описание остальных параметров в описании функции [Shippings::StaticAddRoot](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Shortcuts**

Предназначен для управления ярлыками в панели ярлыков в административной части магазина.

### **Shortcuts::Add**

Назначение:

- Добавление ярлыка в панель ярлыков в админке.

Объявление:

- function Add(\$user\_id, \$url, \$caption)

Параметры:

- **int \$user\_id**. Идентификатор администратора (из таблицы [proshop\\_users](#)), для которого добавляется ярлык.
- **string \$url**. URL-адрес, на который ссылается ярлык. Макс. 2000 символов.
- **string \$caption**. Название ярлыка. Макс. 254 символа.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданного ярлыка в таблице [proshop\\_shortcuts](#).

### **Shortcuts::Delete**

Назначение:

- Удаление указанного ярлыка из панели.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор удаляемого ярлыка из таблицы [proshop\\_shortcuts](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Shortcuts::GetAll**

Назначение:

- Извлечение всех ярлыков, созданных ранее указанным администратором, из таблицы [proshop\\_shortcuts](#).

Объявление:

- function GetAll(\$user\_id)

Параметры:

- **int \$user\_id**. Идентификатор администратора (из таблицы [proshop\\_users](#)), ярлыки которого необходимо извлечь.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_shortcuts](#).

## **Класс SMSGates**

Предназначен для управления СМС-шлюзами, использующимися для оповещения администратора о поступлении заказа в магазине.

### **SMSGates::GetActive**

Назначение:

- Извлечение текущего активного смс-шлюза из таблицы [proshop\\_smsgates](#).

Объявление:

- function GetActive()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_smsgates](#).

### **SMSGates::GetAll**

Назначение:

- Извлечение всех смс-шлюзов из таблицы [proshop\\_smsgates](#).

Объявление:

- function GetAll()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, ключи которого совпадают с полями таблицы [proshop\\_smsgates](#).

## **SMSSGates::GetAllSettings**

Назначение:

- Извлечение всех настроек указанного смс-шлюза из таблицы [proshop\\_smsgates\\_settings](#).

Объявление:

- function GetAllSettings(\$id)

Параметры:

- **int \$id**. Идентификатор шлюза из таблицы [proshop\\_smsgates](#).

Возвращаемое значение:

- Тип **array**. Массив, ключи которого совпадают с полями таблицы [proshop\\_smsgates\\_settings](#).

## **SMSSGates::GetOne**

Назначение:

- Извлечение информации об указанном смс-шлюзе из таблицы [proshop\\_smsgates](#).

Объявление:

- function GetOne(\$id)

Параметры:

- **int \$id**. Идентификатор смс-шлюза из таблицы [proshop\\_smsgates](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, ключи которого совпадают с полями таблицы [proshop\\_smsgates](#).

## **SMSSGates::GetSettingValue**

Назначение:

- Извлечение значения указанной настройки указанного смс-шлюза.

Объявление:

- function GetSettingValue(\$gate\_name, \$name)

Параметры:

- **string \$gate\_name**. Служебное имя смс-шлюза из таблицы [proshop\\_smsgates](#).
- **string \$name**. Имя настройки из таблицы [proshop\\_smsgates\\_settings](#).

Возвращаемое значение:

- Тип **string**. Значение указанной настройки.

## **SMSTGates::SetActive**

Назначение:

- Установка указанного смс-шлюза в качестве текущего активного.

Объявление:

- `function SetActive($id)`

Параметры:

- **int \$id**. Идентификатор смс-шлюза из таблицы [proshop\\_smsgates](#), который необходимо сделать активным.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **SMSTGates::SetSettingValue**

Назначение:

- Установка значения указанной настройки указанного смс-шлюза.

Объявление:

- `function SetSettingValue($gate_id, $name, $value)`

Параметры:

- **int \$gate\_id**. Идентификатор смс-шлюза из таблицы [proshop\\_smsgates](#).
- **string \$name**. Имя настройки из таблицы [proshop\\_smsgates\\_settings](#).
- **string \$value**. Значение настройки. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Stats**

Предназначен для ведения различной статистической информации в магазине.

### **Stats::GetCatViews**

Назначение:

- Определение кол-ва просмотров указанной категории.

Объявление:

- `function GetCatViews($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **int**. Кол-во просмотров.

## Stats::GetLoginCount

Назначение:

- Определение кол-ва попыток входа в аккаунт заданного пользователя в заданную дату.

Объявление:

- `function GetLoginCount($date, $user_id, $email, $success)`

Параметры:

- **string \$date**. Дата в формате ГГГГ-ММ-ДД.
- **int \$user\_id**. Идентификатор зарегистрированного покупателя (пользователя) из таблицы [proshop\\_users](#).
- **string \$email**. E-Mail адрес пользователя.
- **bool \$success**. Если значение данного параметра равно true, определяется кол-во успешных попыток входа. Если false – кол-во неуспешных.

Возвращаемое значение:

- Тип **int**. Кол-во попыток входа.

## Stats::GetLoginHistory

Назначение:

- Извлечение журнала авторизаций пользователей (в т.ч. администраторов) для заданного диапазона дат.

Объявление:

- `function GetLoginHistory($from_date = "", $to_date = "")`

Параметры:

- **string \$from\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД. Если не задана, диапазон считается не заданным и возвращается весь журнал для всего периода времени.
- **string \$to\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД. Если не задана, диапазон считается не заданным и возвращается весь журнал для всего периода времени.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_login\\_history](#).

## Stats::GetOrderedProdsStats

Назначение:

- Извлечение статистики продаж продуктов заданной категории из БД. Проданными считаются только те продукты, статус заказа которых – **Оплачен** или **Завершен**.

Объявление:

- `function GetOrderedProdsStats($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:



- Тип **array**. Массив, содержащий следующие ключи:
  - **qty** – кол-во проданных экземпляров продуктов заданной категории;
  - **price** – сумма всех проданных экземпляров продуктов заданной категории в базовой валюте магазина.

## **Stats::GetOrderedSum**

Назначение:

- Определение суммы совершенных заказов для заданного диапазона дат.

Объявление:

- `function GetOrderedSum($start_date, $end_date, $status_id = 0)`

Параметры:

- **string \$start\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД.
- **string \$end\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД.
- **int \$status\_id**. Статус заказов, для которых необходимо определить общую сумму. Если не задан, будут учитываться заказы со статусами **Оплачен** и **Завершен**.

Возвращаемое значение:

- Тип **int**. Целочисленное значение суммы заказов (с учетом налогов и скидок) в базовой валюте магазина.

## **Stats::GetOrdersCount**

Назначение:

- Определение кол-ва совершенных заказов для заданного диапазона дат.

Объявление:

- `function GetOrdersCount($start_date, $end_date, $status_id = 0)`

Параметры:

- **string \$start\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД.
- **string \$end\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД.
- **int \$status\_id**. Статус заказов, для которых необходимо определить общее кол-во. Если не задан, будут учитываться заказы со статусами **Оплачен** и **Завершен**.

Возвращаемое значение:

- Тип **int**. Кол-во совершенных заказов за данный промежуток времени.

## **Stats::GetProdsCount**

Назначение:

- Определение кол-ва добавленных в магазин продуктов для заданного диапазона дат.

Объявление:

- `function GetProdsCount($start_date, $end_date)`

Параметры:

- **string \$start\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД.

- **string \$end\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД.

Возвращаемое значение:

- Тип **int**. Кол-во добавленных продуктов.

## **Stats::GetSales**

Назначение:

- Формирование статистики продаж для заданного диапазона дат.

Объявление:

- `function GetSales($from_date, $to_date, $cat_id, $prod_id, $bfields)`

Параметры:

- **string \$from\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД.
- **string \$to\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД.
- **int \$cat\_id**. Идентификатор категории (из таблицы [proshop\\_categories](#)), для которой необходимо определить проданные продукты. Данный параметр игнорируется, если параметр **\$prod\_id** не равен 0. Если данный параметр равен 0, статистика формируется для всех категорий.
- **int \$prod\_id**. Идентификатор продукта (из таблицы [proshop\\_products](#)), статистику продаж которого необходимо сформировать. Должен быть равен 0, если требуется сформировать статистику продаж для определенной категории или для всех категорий. Если не равен 0, параметр **\$cat\_id** игнорируется.
- **array \$bfields**. Поля информации, которые необходимо вывести в качестве статистических данных. Может содержать следующие значения (порядок полей в данном массиве имеет значение – в этом порядке будут отсортированы данные):
  - **caption** – название продукта;
  - **sku** – артикул продукта;
  - **data** – дата заказа;
  - **order\_id** – номер заказа (из таблицы [proshop\\_orders](#)).

Возвращаемое значение:

- Тип **array**. Массив, содержащий следующие ключи:
  - **list** – массив, каждый элемент которого является массивом, содержащим следующие ключи:
    - **order\_id** – номер заказа;
    - **product\_id** – идентификатор заказанного продукта;
    - **product\_caption** – название заказанного продукта (если параметр **\$bfields** содержал **caption**);
    - **product\_sku** – артикул заказанного продукта (если параметр **\$bfields** содержал **sku**);
    - **date** – дата заказа (если параметр **\$bfields** содержал **date**);
  - **orders\_count** – кол-во найденных заказов.

## **Stats::GetSearchStats**

Назначение:

- Извлечение статистики поиска продуктов посетителями для заданного диапазона дат.

Объявление:

- `function GetSearchStats($from_date = "", $to_date = "")`

Параметры:

- **string \$from\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД. Если не задана, диапазон считается не заданным и возвращается весь журнал для всего периода времени.
- **string \$to\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД. Если не задана, диапазон считается не заданным и возвращается весь журнал для всего периода времени.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_stats\\_search](#). Статистика выводится по убыванию даты задания поискового запроса.

## **Stats::GetUsersCount**

Назначение:

- Определение кол-ва добавленных (зарегистрированных) пользователей магазина (в т.ч. администраторов) за определенный период времени.

Объявление:

- `function GetUsersCount($start_date, $end_date)`

Параметры:

- **string \$start\_date**. Дата начала диапазона в формате ГГГГ-ММ-ДД.
- **string \$end\_date**. Дата конца диапазона в формате ГГГГ-ММ-ДД.

Возвращаемое значение:

- Тип **int**. Кол-во добавленных пользователей.

## **Stats::RegisterCatView**

Назначение:

- Фиксирование факта просмотра указанной категории. Не учитывается просмотр пользователем, IP-адрес которого – 127.0.0.1.

Объявление:

- `function RegisterCatView($cat_id)`

Параметры:

- **int \$cat\_id**. Идентификатор категории из таблицы [proshop\\_categories](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Stats::RegisterLogin**

Назначение:

- Фиксирование попытки пользователя зайти в свой аккаунт (в таблице [proshop\\_login\\_history](#)). Если IP-адрес пользователя – 127.0.0.1, попытка не фиксируется.

Объявление:

- `function RegisterLogin($email, $user_id, $ip, $is_success)`

Параметры:

- **string \$email**. E-Mail адрес пользователя.
- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **string \$ip**. IP-адрес пользователя в момент попытки входа.
- **char \$is\_success**. Если данный параметр имеет значение **Y**, то попытка входа удачна и пользователь смог войти. Иначе – **N**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Stats::RegisterSearch**

Назначение:

- Фиксирование факта поиска продукта пользователем в таблице [proshop\\_stats\\_search](#) с сохранением поискового запроса и даты его задания. Если IP-адрес пользователя – 127.0.0.1, поиск не фиксируется.

Объявление:

- `function RegisterSearch($search_text)`

Параметры:

- **string \$search\_text**. Поисковый запрос. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Класс Tabs**

Предназначен для управления вкладками модулей в административной части магазина.

### **\$TABS->GetActive**

Назначение:

- Определение номера текущей активной вкладки (начиная от 1) для заданного модуля.

Объявление:

- `function GetActive($mod)`

Параметры:

- **string \$mod**. Название модуля, для которого определяется номер текущей вкладки.

Возвращаемое значение:

- Тип **int**. Номер вкладки или 0, если текущая вкладка не задана.

## **\$TABS->IsActive**

Назначение:

- Определение, является ли указанная вкладка активной в данный момент.

Объявление:

- `function IsActive($module, $tab_num)`

Параметры:

- **string \$module**. Название модуля, для которого определяется активность указанной вкладки.
- **int \$tab\_num**. Номер вкладки (начиная от 1), активность которой нужно определить.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **\$TABS->SetActive**

Назначение:

- Установка активной вкладки заданного модуля.

Объявление:

- `function SetActive($module, $tab_num)`

Параметры:

- **string \$module**. Название модуля, для которого нужно установить номер активной вкладки.
- **int \$tab\_num**. Номер вкладки, которую необходимо сделать активной.

Возвращаемое значение:

- Функция ничего не возвращает.

## **Класс Tags**

Предназначен для управления тэгами (для облака тэгов).

### **Tags : :Add**

Назначение:

- Добавление тэга в таблицу [proshop\\_tags\\_texts](#).

Объявление:

- `function Add($name, $lang_id)`

Параметры:

- **string \$name**. Текст тэга на указанном языке **\$lang\_id**. Макс. 254 символа.
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **int**. Идентификатор записи, вставленной в таблицу [proshop\\_tags\\_texts](#), или 0 в случае ошибки.

## **Tags::AddProdTag**

Назначение:

- Привязка указанного тэга из таблицы [proshop\\_tags\\_texts](#) к указанному продукту.

Объявление:

- `function AddProdTag($prod_id, $tag_id)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$tag\_id**. Идентификатор тэга из таблицы [proshop\\_tags\\_texts](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Tags::Delete**

Назначение:

- Удаление указанного тэга из БД.

Объявление:

- `function Delete($id)`

Параметры:

- **int \$id**. Идентификатор тэга из таблицы [proshop\\_tags\\_texts](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Tags::DelProdTags**

Назначение:

- Удаление всех тэгов на заданном языке, привязанных к заданному продукту.

Объявление:

- `function DelProdTags($prod_id, $lang_id = 0)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка, тэги на котором необходимо удалить. Если язык не задан, удаляются тэги для всех языков.

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Tags::GetAll**

Назначение:

- Извлечение всех тэгов на заданном языке из таблицы [proshop\\_tags\\_texts](#).

Объявление:

- `function GetAll($lang_id, $tags_count = 0)`

Параметры:

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **int \$tags\_count**. Максимально возможное кол-во извлекаемых тэгов. Если равен 0, извлекаются все тэги для заданного языка в порядке возрастания их весов. Вес тэга – кол-во связей с другими элементами (например, с продуктами). Если же данный параметр не равен 0, извлекается заданное кол-во тэгов в порядке убывания их весов, т.е. извлекается заданное кол-во самых весомых тэгов.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_tags\\_texts](#).

## **Tags : : GetProdTags**

Назначение:

- Извлечение всех тэгов, привязанных к заданному продукту, на указанном языке.

Объявление:

- `function GetProdTags($prod_id, $lang_id = 0)`

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$lang\_id**. Идентификатор языка, тэги на котором необходимо извлечь. Если язык не задан, извлекаются тэги для всех языков.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_tags\\_texts](#).

## **Tags : : IdToName**

Назначение:

- Определение текста тэга по его идентификатору на заданном языке.

Объявление:

- `function IdToName($tag_id, $lang_id)`

Параметры:

- **int \$tag\_id**. Идентификатор тэга из таблицы [proshop\\_tags\\_texts](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string/bool**. False – В случае ошибки. Иначе – текст тэга.

## **Tags : : NameToId**

Назначение:

- Определение идентификатора тэга по его тексту на заданном языке.

Объявление:

- `function NameTold($name, $lang_id)`

Параметры:

- **string \$name.** Текст тэга.
- **int \$lang\_id.** Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор тэга из таблицы [proshop\\_tags\\_texts](#).

## Класс Taxes

Предназначен для управления налогами в магазине.

### **Taxes : :AddRoot**

Назначение:

- Добавление базовой информации о налоге в БД.

Объявление:

- `function AddRoot($type, $charge, $value, $no_vat)`

Параметры:

- **string \$type.** Тип налога (**per** – в процентах, **abs** – абсолютный, в базовой валюте магазина).
- **char \$charge.** Если данный параметр равен **Y**, налог считается начисляемым. Если **N** – налог выделяемый. Выделяемый налог – это налог, который уже входит в цену любого продукта, заданного в таблице [proshop\\_products](#).
- **float \$value.** Значение налога в процентах или в базовой валюте магазина, в зависимости от значения параметра **\$type**.
- **char \$no\_vat.** Если данный параметр равен **Y**, форма этого налога – «Без НДС». Значение **\$value** при этом должно быть равно 0.

Возвращаемое значение:

- Тип **int/bool**. False – В случае ошибки. Иначе – идентификатор созданного налога в таблице [proshop\\_taxes](#).

### **Taxes : :AddText**

Назначение:

- Добавление названия указанного налога для указанного языка в БД.

Объявление:

- `function AddText($lang_id, $tax_id, $caption)`

Параметры:

- **int \$lang\_id.** Идентификатор языка из таблицы [proshop\\_languages](#).
- **int \$tax\_id.** Идентификатор налога из таблицы [proshop\\_taxes](#).



- **string \$caption**. Название налога на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Taxes::Delete**

Назначение:

- Удаление указанного налога из БД.

Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор налога из таблицы [proshop\\_taxes](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Taxes::GetAll**

Назначение:

- Извлечение всех налогов из БД.

Объявление:

- function GetAll(\$lang\_id)

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия налогов.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_taxes](#);
  - **caption** – название налога для указанного языка.

## **Taxes::GetCount**

Назначение:

- Извлечение кол-ва всех налогов в таблице [proshop\\_taxes](#).

Объявление:

- function GetCount()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Кол-во налогов или 0, если модуль налогов не включен или налоги не созданы.

## **Taxes::GetOne**

Назначение:

- Извлечение информации об указанном налоге из БД.

Объявление:

- `function GetOne($tax_id, $lang_id = 0)`

Параметры:

- **int \$tax\_id**. Идентификатор налога из таблицы [proshop\\_taxes](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название налога. Если данный параметр равен 0, название налога не извлекается.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_taxes](#);
  - **caption** – название налога для указанного языка (если параметр **\$lang\_id** не равен 0).

## **Taxes::IsTextExists**

Назначение:

- Определение, не занято ли указанное название налога на заданном языке.

Объявление:

- `function IsTextExists($caption, $lang_id, $tax_id = 0)`

Параметры:

- **string \$caption**. Название налога, которое необходимо проверить.
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **int \$tax\_id**. Идентификатор налога из таблицы [proshop\\_taxes](#), который не следует учитывать при поиске. Если данный параметр равен 0, учитываются все налоги.

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

## **Taxes::Update**

Назначение:

- Обновление информации о налоге в БД.

Объявление:

- `function Update($lang_id, $info)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо обновить название налога.
- **array \$info**. Массив с информацией о налоге. Должен содержать следующие ключи:
  - **type** – тип налога (**abs** – абсолютный в базовой валюте магазина, **per** – в процентах);

- **charge** - если данный параметр равен **Y**, налог считается начисляемым. Если **N** – налог выделяемый. Выделяемый налог – это налог, который уже входит в цену любого продукта, заданного в таблице [proshop\\_products](#);
- **value** – значение налога в процентах или в базовой валюте магазина, в зависимости от значения **type**;
- **no\_vat** – если данный параметр равен **Y**, форма этого налога – «Без НДС». Значение **value** при этом должно быть равно 0;
- **caption** – название налога на указанном языке;
- **id** – идентификатор налога в таблице [proshop\\_taxes](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс TplEditor

Предназначен для чтения/записи (редактирования) шаблонов дизайна магазина.

### TplEditor::CreateDir

Назначение:

- Создание директории по заданному полному пути в файловой системе сервера.

Объявление:

- function CreateDir(\$dir)

Параметры:

- **string \$dir**. Полный путь к создаваемой директории.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### TplEditor::GetContent

Назначение:

- Чтение содержимого файла по указанному пути на сервере.

Объявление:

- function GetContent(\$path)

Параметры:

- **string \$path**. Полный путь к файлу, содержимое которого необходимо прочитать.

Возвращаемое значение:

- Тип **string/bool**. False – в случае ошибки. Иначе – содержимое указанного текстового файла.

### TplEditor::GetDir

Назначение:

- Извлечение содержимого (списка файлов и директорий) указанной директории.

Объявление:

- `function GetDir($root)`

Параметры:

- **string \$root**. Полный путь к директории, содержимое которой необходимо извлечь.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, каждый элемент которого описывает найденный объект файловой системы и является массивом, состоящим из следующих ключей:
  - **name** – имя файла/директории;
  - **md5name** – значение md5-хеши от имени файла/директории;
  - **atime** – время последнего доступа к файлу (в формате ДД-ММ-ГГГГ ЧЧ:ММ:СС);
  - **mtime** – время последней модификации файла (в формате ДД-ММ-ГГГГ ЧЧ:ММ:СС);
  - **is\_dir** – является ли объект директорией (**true** – является, **false** – не является);
  - **type** – расширение файла в верхнем регистре (если объект является файлом);
  - **size** – размер файла в Кб (если объект является файлом).

## **TplEditor::PutContent**

Назначение:

- Сохранение текста указанного файла.

Объявление:

- `function PutContent($path, $content)`

Параметры:

- **string \$path**. Полный путь к сохраняемому файлу. Если файл существует, он будет перезаписан.
- **string \$content**. Текстовое содержимое файла.

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

## **TplEditor::UploadFile**

Назначение:

- Загрузка файла на сервер в указанную директорию.

Объявление:

- `function UploadFile($dir, $file)`

Параметры:

- **string \$dir**. Полный путь к директории на сервере, в которой необходимо сохранить загружаемый файл.
- **file \$file**. Описание загружаемого файла (из **\$\_FILES**).

Возвращаемое значение:

- Тип **bool**. `True/false` – в зависимости от успешности операции.

Пример:

- `TplEditor::UploadFile($dir, $_FILES['upload_file']);`

## Класс `UrlNameHistory`

Предназначен для добавления имени в URL объекта в историю (при смене имени) – в таблицу [proshop\\_url\\_name\\_history](#), удаления имени в URL из истории и поиска имени в URL для объекта заданного типа.

### `UrlNameHistory::Add`

Назначение:

- Добавление имени в URL заданного объекта в историю.

Объявление:

- `function Add($obj_id, $obj_type, $url_name)`

Параметры:

- **int \$obj\_id**. Идентификатор объекта, имя в URL которого необходимо добавить (например, для объекта «Продукт» - это идентификатор продукта из таблицы [proshop\\_products](#)).
- **string \$obj\_type**. Тип объекта, идентификатор которого указан в параметре **\$obj\_id**. Обозначения типов объектов см. в описании таблицы [proshop\\_url\\_name\\_history](#).
- **string \$url\_name**. Имя в URL объекта, которое нужно добавить в историю. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### `UrlNameHistory::Delete`

Назначение:

- Удаление информации об истории имени в URL для заданного объекта.

Объявление:

- `function Delete($obj_id, $obj_type)`

Параметры:

- **int \$obj\_id**. Идентификатор объекта, имя в URL которого необходимо удалить из истории (например, для объекта «Продукт» - это идентификатор продукта из таблицы [proshop\\_products](#)).
- **string \$obj\_type**. Тип объекта, идентификатор которого указан в параметре **\$obj\_id**. Обозначения типов объектов см. в описании таблицы [proshop\\_url\\_name\\_history](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### `UrlNameHistory::Find`

Назначение:

- Поиск идентификатора объекта заданного типа в истории имен по заданному имени в URL.

Объявление:

- function Find(\$obj\_type, \$url\_name)

Параметры:

- **string \$obj\_type**. Тип объекта, идентификатор которого необходимо найти в таблице [proshop\\_url\\_name\\_history](#). Обозначения типов объектов см. в описании таблицы [proshop\\_url\\_name\\_history](#).
- **string \$url\_name**. Имя в URL объекта, идентификатор которого необходимо найти.

Возвращаемое значение:

- Тип **int**. Идентификатор найденного объекта или 0, если объект не найден.

## Класс Users

Предназначен для управления пользователями магазина.

### Users::Add

Назначение:

- Добавление пользователя в таблицу [proshop\\_users](#).

Объявление:

- function Add(\$user\_info)

Параметры:

- **array \$user\_info**. Массив с информацией о пользователе. Должен содержать следующие ключи:
  - **type** – тип добавляемого пользователя (**CUSTOMER** или **ADMIN**);
  - **email** – адрес электронной почты пользователя (в качестве логина). Макс. 254 символа;
  - **first\_name** – имя пользователя. Макс. 254 символа;
  - **group\_id** – идентификатор группы администраторов (если добавляется администратор) из таблицы [proshop\\_admins\\_groups](#);
  - **title** – обращение к пользователю (**MR** – мистер, **MS** - мисс) – для аккаунта типа **CUSTOMER**;
  - **last\_name** – фамилия пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
  - **second\_name** – отчество пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
  - **gender** – пол пользователя (M – мужской, F - женский). Действительно только для аккаунта типа **CUSTOMER**;
  - **company\_name** – название организации пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;

- **ogrn** – ОГРН организации пользователя. Макс. 60 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **inn** – ИНН организации пользователя. Макс. 60 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **kpp** – КПП организации пользователя. Макс. 60 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **contact\_person** – контактное лицо организации. Макс. 100 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **legal\_address** – юридический адрес организации. Макс. 200 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **bank\_name** – наименование банка пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **bank\_bik** – БИК банка организации пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **bank\_rs** – р/с банка организации пользователя. Макс. 100 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **bank\_ks** – к/с банка организации пользователя. Макс. 100 символов. Действительно только для аккаунта типа **CUSTOMER**;
- **phone** – номер (или номера) телефона пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **mobile\_phone** – номер мобильного телефона пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **fax** – номер факса пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **website** – адрес сайта пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **address** – строка адреса пользователя с названием улицы. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **zip\_code** – индекс адреса пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **city** – город пользователя. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **country** – идентификатор страны пользователя (из таблицы [proshop\\_countries](#)). Действительно только для аккаунта типа **CUSTOMER**;
- **state** – идентификатор области указанной страны пользователя (из таблицы [proshop\\_states](#)). Действительно только для аккаунта типа **CUSTOMER**;
- **partner\_id** – идентификатор партнера (из таблицы [proshop\\_users](#)), который пригласил создаваемого пользователя на сайт. Действительно только для аккаунта типа **CUSTOMER**;

- **lang\_id** – идентификатор предпочитаемого языка (из таблицы [proshop\\_languages](#)) создаваемого пользователя. Если данный параметр не задан, он задается при входе пользователя в свой аккаунт. Действительно только для аккаунта типа **CUSTOMER**;
- **curr\_id** - идентификатор предпочитаемой валюты (из таблицы [proshop\\_currencies](#)) создаваемого пользователя. Если данный параметр не задан, он задается при входе пользователя в свой аккаунт. Действительно только для аккаунта типа **CUSTOMER**;
- **person\_type** – тип покупателя: **A** – юридическое лицо, **N** – физическое лицо. Действительно только для аккаунта типа **CUSTOMER**;
- **ship\_address** – строка адреса доставки товара пользователю с названием улицы. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **ship\_zip\_code** – индекс адреса доставки товара пользователю. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **ship\_city** – город адреса доставки товара пользователю. Макс. 254 символа. Действительно только для аккаунта типа **CUSTOMER**;
- **ship\_country** – идентификатор страны адреса доставки товара пользователю (из таблицы [proshop\\_countries](#)). Действительно только для аккаунта типа **CUSTOMER**;
- **ship\_state** – идентификатор области указанной страны адреса доставки товара пользователю (из таблицы [proshop\\_states](#)). Действительно только для аккаунта типа **CUSTOMER**.

Возвращаемое значение:

- Тип **int/bool**. **False** – в случае ошибки. Иначе – идентификатор созданного пользователя в таблице [proshop\\_users](#).

## **Users::AddMoney**

Назначение:

- Начисление указанной суммы к балансу указанного партнера (партнерское вознаграждение).

Объявление:

- `function AddMoney($partner_id, $sum)`

Параметры:

- **int \$partner\_id**. Идентификатор партнера (который приглашает пользователей на сайт) из таблицы [proshop\\_users](#).
- **float \$sum**. Сумма в базовой валюте, которую необходимо прибавить к балансу указанного партнера.

Возвращаемое значение:

- Тип **bool**. **True/false** – в зависимости от успешности операции.

## **Users::Delete**

Назначение:

- Удаление указанного пользователя из БД.



Объявление:

- function Delete(\$id)

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Users::GenPwdSalt**

Назначение:

- Генерирование «соли» к паролю пользователя.

Объявление:

- function GenPwdSalt()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Соль длиной 16 символов к паролю пользователя.

## **Users::GetAll**

Назначение:

- Извлечение всех пользователей из таблицы [proshop\\_users](#).

Объявление:

- function GetAll(\$type\_id, \$only\_ids = false, \$complete\_sql = "")

Параметры:

- **int \$type\_id**. Фильтр по типу пользователей. Идентификатор из таблицы [proshop\\_users\\_types](#). Если данный параметр равен 0, фильтр по типу не применяется.
- **bool \$only\_ids**. Если данный параметр равен true, функция вернет только идентификаторы пользователей.
- **string \$complete\_sql**. Часть SQL-запроса, которую можно пристыковать к концу основного SQL-запроса, формируемого функцией. Используется для пейджинга (см. класс [Pager](#)).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_users](#) (или только поле **id**, если параметр **\$only\_ids** равен **true**);
  - **type** – тип пользователя (**ADMIN** – администратор, **CUSTOMER** – зарегистрированный покупатель).

## **Users::GetEmail**

Назначение:

- Определение адреса электронной почты указанного пользователя.

Объявление:

- `function GetEmail($user_id)`

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **string**. E-Mail указанного пользователя.

## **Users::GetFields**

Назначение:

- Извлечение информации о возможных полях профайла зарегистрированного покупателя из таблицы [proshop\\_users\\_fields](#).

Объявление:

- `function GetFields()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, содержащий ключи, совпадающие со значениями поля **name** таблицы [proshop\\_users\\_fields](#). Значением каждого ключа является массив, с ключами, соответствующими полям таблицы [proshop\\_users\\_fields](#). См. пример.

Пример:

- ```
$fields = Users::GetFields();  
// $fields['c_first_name']['avail'] – доступность поля c_first_name  
// ...
```

## **Users::GetIdByType**

Назначение:

- Определение идентификатора типа пользователя по его названию из таблицы [proshop\\_users\\_types](#).

Объявление:

- `function GetIdByType($type)`

Параметры:

- **string \$type**. Тип пользователя (**CUSTOMER** или **ADMIN**).

Возвращаемое значение:

- Тип **int**. Идентификатор указанного типа пользователя из таблицы [proshop\\_users\\_types](#).

## **Users::GetLangID**

Назначение:

- Определение идентификатора предпочитаемого указанным пользователем языка.

Объявление:

- function GetLangID(\$user\_id)

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **int**. Идентификатор предпочитаемого указанным пользователем языка из таблицы [proshop\\_languages](#). Если предпочитаемый язык для указанного пользователя не задан, возвращается идентификатор языка магазина по умолчанию.

## **Users::GetOne**

Назначение:

- Извлечение информации об указанном пользователе из таблицы [proshop\\_users](#).

Объявление:

- function GetOne(\$id)

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - все поля таблицы [proshop\\_users](#);
  - **type** – тип пользователя (**ADMIN** – администратор, **CUSTOMER** – зарегистрированный покупатель).

## **Users::GetPartners**

Назначение:

- Извлечение информации о партнерах магазина из таблицы [proshop\\_users](#) (пользователи с положительным партнерским балансом).

Объявление:

- function GetPartners()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_users](#).

## **Users::GetPwdHash**

Назначение:

- Генерирование хэш-значения пароля для пользователя с применением «соли» по алгоритму SHA-256.

Объявление:

- `function GetPwdHash($password, $salt1, $salt2)`

Параметры:

- **string \$password**. Строка пароля.
- **string \$salt1**. Соль №1.
- **string \$salt2**. Соль №2.

Возвращаемое значение:

- Тип **string**. Хэш-значение пароля, сгенерированное по алгоритму SHA-256.

## **Users::GetPwdSalt1**

Назначение:

- Извлечение «соли» №1 к паролю указанного пользователя из таблицы [proshop\\_users](#).

Объявление:

- `function GetPwdSalt1($user_id)`

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **string**. Значение «соли» (строка символов).

## **Users::GetPwdSalt2**

Назначение:

- Извлечение «соли» №2 к паролю указанного пользователя из таблицы [proshop\\_users](#).

Объявление:

- `function GetPwdSalt2($user_id)`

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **string**. Значение «соли» (строка символов).

## **Users::GetTypeCount**

Назначение:

- Определение кол-ва пользователей заданного типа в таблице [proshop\\_users](#).

Объявление:

- `function GetTypeCount($type_id)`

Параметры:

- **int \$type\_id**. Идентификатор типа пользователя из таблицы [proshop\\_users\\_types](#).

Возвращаемое значение:

- Тип **int**. Кол-во пользователей указанного типа.

## **Users::GetTypes**

Назначение:

- Извлечение возможных типов пользователей из таблицы [proshop\\_users\\_types](#).

Объявление:

- `function GetTypes()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим ключи, совпадающие с полями таблицы [proshop\\_users\\_types](#).

## **Users::GetUserId**

Назначение:

- Определение идентификатора пользователя из таблицы [proshop\\_users](#) по его E-Mail и типу.

Объявление:

- `function GetUserId($email, $type)`

Параметры:

- **string \$email**. E-Mail адрес пользователя из таблицы [proshop\\_users](#).
- **string \$type**. Тип пользователя (**ADMIN** – администратор, **CUSTOMER** - зарегистрированный покупатель).

Возвращаемое значение:

- Тип **int**. Идентификатор пользователя из таблицы [proshop\\_users](#) или 0, если заданный пользователь в таблице не найден.

## **Users::IdByActivationKey**

Назначение:

- Определение идентификатора пользователя по ключу его активации из таблицы [proshop\\_users](#).

Объявление:

- `function IdByActivationKey($key)`

Параметры:

- **string \$key**. Строка ключа активации пользователя. Ключ не должен быть пустым.

Возвращаемое значение:

- Тип **int**. Идентификатор пользователя из таблицы [proshop\\_users](#) или 0, если пользователь, соответствующий данному ключу, не найден.

## **Users::IdByPwdRecoverKey**

Назначение:

- Определение идентификатора пользователя по ключу восстановления его пароля из таблицы [proshop\\_users](#).

Объявление:

- `function IdByPwdRecoverKey($key)`

Параметры:

- **string \$key**. Строка ключа восстановления пароля. Ключ не должен быть пустым.

Возвращаемое значение:

- Тип **int**. Идентификатор пользователя из таблицы [proshop\\_users](#) или 0, если пользователь, соответствующий данному ключу, не найден.

## **Users::IncTotalOrdersSum**

Назначение:

- Увеличение значения поля **total\_orders\_sum** таблицы [proshop\\_users](#) для указанного зарегистрированного покупателя.

Объявление:

- `function IncTotalOrdersSum($id, $sum)`

Параметры:

- **int \$id**. Идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#).
- **float \$sum**. Сумма, которую нужно прибавить к полю **total\_orders\_sum** таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Users::IsActivated**

Назначение:

- Определение, активирован ли указанный пользователь.

Объявление:

- `function IsActivated($id)`

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Users::IsExists**

Назначение:

- Определение, существует ли пользователь с указанным идентификатором в таблице [proshop\\_users](#).

Объявление:

- `function IsExists($id)`

Параметры:

- **int \$id**. Идентификатор пользователя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Users::IsFieldAvail**

Назначение:

- Определение, доступно ли для заполнения указанное поле профайла покупателя.

Объявление:

- function IsFieldAvail(\$name)

Параметры:

- **string \$name**. Имя поля из таблицы [proshop\\_users\\_fields](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Users::IsFieldRequired**

Назначение:

- Определение, обязательно ли для заполнения указанное поле профайла покупателя.

Объявление:

- function IsFieldRequired(\$name)

Параметры:

- **string \$name**. Имя поля из таблицы [proshop\\_users\\_fields](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Users::IsUser**

Назначение:

- Определение, существует ли указанный пользователь с указанным хэш-значением пароля в таблице [proshop\\_users](#).

Объявление:

- function IsUser(\$id, \$hash)

Параметры:

- **int \$id**. Идентификатор пользователя в таблице [proshop\\_users](#).
- **string \$hash**. Хэш-значение пароля пользователя.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## **Users::PayPartner**

Назначение:

- Выплата указанному партнеру указанной суммы по партнерской программе, т.е. уменьшение виртуального баланса партнера на размер указанной суммы.

Объявление:

- `function PayPartner($partner_id, $sum)`

Параметры:

- **int \$partner\_id**. Идентификатор пользователя-партнера из таблицы [proshop\\_users](#).
- **float \$sum**. Сумма выплаты в базовой валюте магазина.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **Users::SaveField**

Назначение:

- Сохранение доступности и обязательности указанного поля профайла покупателя в таблице [proshop\\_users\\_fields](#).

Объявление:

- `function SaveField($name, $savail, $required)`

Параметры:

- **string \$name**. Название поля из таблицы [proshop\\_users\\_fields](#).
- **char \$savail**. Флаг доступности поля для его заполнения покупателем (**Y** – доступно, **N** - недоступно).
- **char \$required**. Флаг обязательности поля для его заполнения покупателем (**Y** – обязательно, **N** - необязательно). Данный флаг имеет смысл только при **\$savail** равном **Y**.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **Users::SetActivationKey**

Назначение:

- Установка ключа активации указанного пользователя в таблице [proshop\\_users](#).

Объявление:

- `function SetActivationKey($id, $key)`

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **string \$key**. Ключ активации пользователя. Последовательность английских букв и цифр длиной не более 60 символов.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **Users::SetDiscount**

Назначение:



- Установка указанной скидки для указанного зарегистрированного покупателя.

Объявление:

- function SetDiscount(\$user\_id, \$discount\_id)

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **int \$discount\_id**. Идентификатор скидки для пользователей из таблицы [proshop\\_discounts](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Users::SetLangID**

Назначение:

- Установка предпочитаемого языка для указанного зарегистрированного покупателя в таблице [proshop\\_users](#).

Объявление:

- function SetLangID(\$user\_id, \$lang\_id)

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **int \$lang\_id**. Идентификатор предпочитаемого языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Users::SetPwdHash**

Назначение:

- Установка нового хэш-значения пароля указанного пользователя в таблице [proshop\\_users](#), а также «соли» №1 и «соли» №2 для увеличения стойкости хэш-значения к перебору.

Объявление:

- function SetPwdHash(\$id, \$hash, \$salt1, \$salt2)

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **string \$hash**. Хэш-значение, сгенерированное функцией [Users::GetPwdHash](#).
- **string \$salt1**. Значение «соли» №1, сгенерированное функцией [Users::GenPwdSalt](#).
- **string \$salt2**. Значение «соли» №2, сгенерированное функцией [Users::GenPwdSalt](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Users::SetPwdRecoverKey**

Назначение:

- Установка ключа восстановления пароля указанного пользователя в таблице [proshop\\_users](#).

Объявление:

- function SetPwdRecoverKey(\$id, \$key)

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **string \$key**. Строка ключа восстановления пароля, состоящая из английских букв и цифр длиной не более 60 символов.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Users::Update**

Назначение:

- Обновление информации об указанном пользователе в таблице [proshop\\_users](#).

Объявление:

- function Update(\$id, \$user\_info)

Параметры:

- **int \$id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **array \$user\_info**. См. аналогичный параметр функции [Users::Add](#). Единственное отличие – нельзя менять тип пользователя (значение **\$user\_info['type']** должно указывать на тип пользователя, а не изменять его).

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## **Users::UpdateCurrency**

Назначение:

- Установка предпочитаемой валюты для указанного пользователя в таблице [proshop\\_users](#).

Объявление:

- function UpdateCurrency(\$user\_id, \$curr\_id)

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).
- **int \$curr\_id**. Идентификатор предпочитаемой валюты из таблицы [proshop\\_currencies](#).

Возвращаемое значение:

- Тип **bool**. True/false - в зависимости от успешности операции.

## **Класс Visitors**

Предназначен для ведения статистики посещений магазина.

### **Visitors::GetAll**

Назначение:

- Извлечение всех уникальных посещений для заданного диапазона дат из таблицы [proshop\\_visitors](#).

Объявление:

- `function GetAll($from_date, $to_date)`

Параметры:

- **string \$from\_date**. Дата начала периода (в формате ГГГГ-ММ-ДД).
- **string \$to\_date**. Дата конца периода (в формате ГГГГ-ММ-ДД).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом с ключами, совпадающими с полями таблицы [proshop\\_visitors](#).

## **Visitors::GetCount**

Назначение:

- Определение кол-ва посещений (всего и уникальных) за весь период времени.

Объявление:

- `function GetCount()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **array**. Массив, содержащий следующие ключи:
  - **total** – кол-во посещений всего;
  - **unique** – кол-во уникальных посещений.

## **Visitors::GetIP**

Назначение:

- Определение IP-адреса указанного посетителя из таблицы [proshop\\_visitors](#).

Объявление:

- `function GetIP($id)`

Параметры:

- **int \$id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).

Возвращаемое значение:

- Тип **string**. IP-адрес указанного посетителя или пустая строка в случае ошибки.

## **Visitors::GetPartnerId**

Назначение:

- Определение идентификатора партнера, который пригласил указанного посетителя на сайт по партнерской программе.

Объявление:

- `function GetPartnerId($visitor_id)`

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).

Возвращаемое значение:

- Тип **int**. Идентификатор партнера из таблицы [proshop\\_users](#), или 0, если указанного посетителя никто не приглашал.

## **Visitors::GetUserId**

Назначение:

- Определение идентификатора пользователя (из таблицы [proshop\\_users](#)) указанного посетителя, если посетитель зарегистрирован в качестве покупателя.

Объявление:

- `function GetUserId($visitor_id)`

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).

Возвращаемое значение:

- Тип **int**. Идентификатор пользователя из таблицы [proshop\\_users](#), или 0, если указанный посетитель не зарегистрирован.

## **Visitors::GuestsOnline**

Назначение:

- Определение кол-ва незарегистрированных посетителей на сайте в данный момент времени.

Объявление:

- `function GuestsOnline()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. **False** – в случае ошибки. Иначе – кол-во гостевых посетителей на сайте в данный момент.

## **Visitors::Register**

Назначение:

- Фиксирование захода посетителя на сайт в таблице [proshop\\_visitors](#).

Объявление:

- `function Register($ip, $uagent, $referer)`

Параметры:

- **string \$ip**. IP-адрес посетителя. Макс. 15 символов.
- **string \$uagent**. Строка User Agent браузера посетителя. Макс. 254 символа.

- **string \$referer**. Адрес страницы, с которой посетитель перешел на сайт первый раз. Макс. 254 символа.

Возвращаемое значение:

- Тип **int**. Идентификатор посетителя из таблицы [proshop\\_visitors](#) или 0 - в случае ошибки.

## **Visitors::SetAsUser**

Назначение:

- Связывание указанного зарегистрированного покупателя с указанным посетителем в таблице [proshop\\_visitors](#).

Объявление:

- function SetAsUser(\$visitor\_id, \$user\_id)

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).
- **int \$user\_id**. Идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Visitors::SetPartnerId**

Назначение:

- Привязка идентификатора партнера к указанному посетителю, приглашенному этим партнером на сайт, в таблице [proshop\\_visitors](#).

Объявление:

- function SetPartnerId(\$visitor\_id, \$partner\_id)

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).
- **int \$partner\_id**. Идентификатор партнера из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Visitors::UpdateActTime**

Назначение:

- Обновление времени последнего действия указанного посетителя на сайте в таблице [proshop\\_visitors](#).

Объявление:

- function UpdateActTime(\$visitor\_id)

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Visitors::UsersOnline

Назначение:

- Определение кол-ва зарегистрированных посетителей на сайте в данный момент времени.

Объявление:

- `function UsersOnline()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – кол-во зарегистрированных посетителей на сайте в данный момент.

## Класс Votes

Предназначен для управления голосованиями в магазине.

### Votes::AddVarRoot

Назначение:

- Добавление базовой информации о варианте ответа указанного голосования в таблицу [proshop\\_votes\\_vars](#).

Объявление:

- `function AddVarRoot($vote_id)`

Параметры:

- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор созданного варианта ответа указанного голосования из таблицы [proshop\\_votes\\_vars](#).

### Votes::AddVarText

Назначение:

- Добавление текста указанного варианта ответа для указанного языка в таблицу [proshop\\_votes\\_vars\\_texts](#).

Объявление:

- `function AddVarText($var_id, $lang_id, $caption)`

Параметры:

- **int \$var\_id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **string \$caption**. Текст варианта ответа на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::AddVoteRoot**

Назначение:

- Добавление базовой информации о голосовании в таблицу [proshop\\_votes](#).

Объявление:

- `function AddVoteRoot()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. False – в случае ошибки. Иначе – идентификатор добавленного голосования из таблицы [proshop\\_votes](#).

## **Votes::AddVoteText**

Назначение:

- Добавление названия указанного голосования для указанного языка в таблицу [proshop\\_votes\\_texts](#).

Объявление:

- `function AddVoteText($vote_id, $lang_id, $caption)`

Параметры:

- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **string \$caption**. Название голосования на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::CanVote**

Назначение:

- Определение, может ли оставить свой голос указанный посетитель в указанном голосовании. Применяется для защиты от нескольких голосов одним и тем же посетителем.

Объявление:

- `function CanVote($vote_id, $visitor_id)`

Параметры:

- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).
- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).

Возвращаемое значение:

- Тип **bool**. Возвращается true, если указанный посетитель может оставить голос. Иначе – false.

## **Votes::DelVariant**

Назначение:

- Удаление указанного варианта ответа из БД.

Объявление:

- `function DelVariant($id)`

Параметры:

- **int \$id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Votes::DelVote**

Назначение:

- Удаление указанного голосования из БД.

Объявление:

- `function DelVote($id)`

Параметры:

- **int \$id**. Идентификатор голосования из таблицы [proshop\\_votes](#).

Возвращаемое значение:

- Тип **bool**. Всегда true.

## **Votes::GetActiveVote**

Назначение:

- Извлечение информации о текущем голосовании из БД.

Объявление:

- `function GetActiveVote($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название голосования.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, содержащий следующие ключи:
  - **id** – идентификатор голосования (из таблицы [proshop\\_votes](#));
  - **date** – дата и время создания голосования;
  - **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)) названия голосования;
  - **caption** – название голосования на указанном языке.

## **Votes::GetActiveVoteId**

Назначение:

- Определение идентификатора текущего активного голосования из таблицы [proshop\\_votes](#).

Объявление:

- `function GetActiveVoteId()`



Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int**. Идентификатор активного голосования из таблицы [proshop\\_votes](#), или 0, если активное голосование не найдено.

## **Votes::GetFirstVoteId**

Назначение:

- Определение идентификатора первого голосования из всего списка голосований в таблице [proshop\\_votes](#).

Объявление:

- `function GetFirstVoteId()`

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **int/bool**. `False` – в случае ошибки. Иначе – идентификатор первого голосования из таблицы [proshop\\_votes](#).

## **Votes::GetVariant**

Назначение:

- Извлечение информации об указанном варианте ответа из БД.

Объявление:

- `function GetVariant($var_id, $lang_id)`

Параметры:

- **int \$var\_id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#), на котором необходимо вернуть текст варианта ответа.

Возвращаемое значение:

- Тип **array/bool**. `False` – в случае ошибки. Иначе – массив, состоящий из следующих ключей:
  - все поля таблицы [proshop\\_votes\\_vars](#);
  - **caption** – текст варианта ответа на указанном языке.

## **Votes::GetVariants**

Назначение:

- Извлечение вариантов ответов указанного голосования из БД.

Объявление:

- `function GetVariants($vote_id, $lang_id)`

Параметры:

- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).

- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#), для которого необходимо извлечь тексты вариантов ответов.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе – массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - все поля таблицы [proshop\\_votes\\_vars](#);
  - **caption** – текст варианта ответа на указанном языке.

## **Votes::GetVote**

Назначение:

- Извлечение информации об указанном голосовании из БД.

Объявление:

- `function GetVote($vote_id, $lang_id)`

Параметры:

- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь название голосования.

Возвращаемое значение:

- Тип **array/bool**. См. возвращаемое значение функции [Votes::GetActiveVote](#).

## **Votes::GetVotes**

Назначение:

- Извлечение всех голосований из БД.

Объявление:

- `function GetVotes($lang_id)`

Параметры:

- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), для которого необходимо извлечь названия голосований.

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, состоящим из следующих ключей:
  - все поля таблицы [proshop\\_votes](#);
  - **caption** – название голосования на указанном языке.

## **Votes::IsVarTextExists**

Назначение:

- Определение, не существует ли уже указанный текст варианта ответа для заданного языка и заданного голосования в таблице [proshop\\_votes\\_vars\\_texts](#).

Объявление:

- function IsVarTextExists(\$caption, \$lang\_id, \$vote\_id, \$var\_id = 0)

Параметры:

- **string \$caption**. Текст варианта ответа.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором задан текст варианта ответа.
- **int \$vote\_id**. Идентификатор голосования (из таблицы [proshop\\_votes](#)), в пределах которого ищется заданный текст варианта ответа.
- **int \$var\_id**. Идентификатор варианта ответа (из таблицы [proshop\\_votes\\_vars](#)), который не следует принимать во внимание при поиске заданного текста варианта ответа. Если данный параметр равен 0, учитываются все варианты ответов.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## Votes::IsVoteTextExists

Назначение:

- Определение, не существует ли уже указанное название голосования для заданного языка в таблице [proshop\\_votes\\_texts](#).

Объявление:

- function IsVoteTextExists(\$caption, \$lang\_id, \$vote\_id = 0)

Параметры:

- **string \$caption**. Название голосования.
- **int \$lang\_id**. Идентификатор языка (из таблицы [proshop\\_languages](#)), на котором задано название голосования.
- **int \$vote\_id**. Идентификатор голосования (из таблицы [proshop\\_votes](#)), который не следует принимать во внимание при поиске заданного названия. Если данный параметр равен 0, учитываются все голосования.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от результата.

## Votes::Register

Назначение:

- Фиксирование факта голосования указанным посетителем в таблице [proshop\\_votes\\_antiflood](#).

Объявление:

- function Register(\$visitor\_id, \$vote\_id)

Параметры:

- **int \$visitor\_id**. Идентификатор посетителя из таблицы [proshop\\_visitors](#).
- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::SetActiveVote**

Назначение:

- Установка текущего активного голосования.

Объявление:

- function SetActiveVote(\$id)

Параметры:

- **int \$id**. Идентификатор голосования из таблицы [proshop\\_votes](#), которое необходимо сделать активным.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::SetPos**

Назначение:

- Установка позиции указанного варианта ответа по отношению к другим вариантам ответов в таблице [proshop\\_votes\\_vars](#).

Объявление:

- function SetPos(\$var\_id, \$pos)

Параметры:

- **int \$var\_id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).
- **int \$pos**. Значение позиции.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::SetVarVotes**

Назначение:

- Установка кол-ва голосов для указанного варианта ответа в таблице [proshop\\_votes\\_vars](#).

Объявление:

- function SetVarVotes(\$var\_id, \$votes)

Параметры:

- **int \$var\_id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).
- **int \$votes**. Кол-во голосов указанного варианта ответа, которое необходимо установить.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::UpdateVariant**

Назначение:

- Установка текста указанного варианта ответа на указанном языке в таблице [proshop\\_votes\\_vars\\_texts](#).

Объявление:

- `function UpdateVariant($var_id, $lang_id, $caption)`

Параметры:

- **int \$var\_id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **string \$caption**. Текст варианта ответа на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::UpdateVote**

Назначение:

- Установка названия указанного голосования на указанном языке в таблице [proshop\\_votes\\_texts](#).

Объявление:

- `function UpdateVote($id, $lang_id, $caption)`

Параметры:

- **int \$id**. Идентификатор голосования из таблицы [proshop\\_votes](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).
- **string \$caption**. Название голосования на указанном языке. Макс. 254 символа.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **Votes::VoteIdToCaption**

Назначение:

- Определение названия указанного голосования на указанном языке.

Объявление:

- `function VoteIdToCaption($vote_id, $lang_id)`

Параметры:

- **int \$vote\_id**. Идентификатор голосования из таблицы [proshop\\_votes](#).
- **int \$lang\_id**. Идентификатор языка из таблицы [proshop\\_languages](#).

Возвращаемое значение:

- Тип **string**. Название голосования.

## **Votes::VoteVar**

Назначение:

- Увеличение кол-ва голосований на единицу для указанного варианта ответа в таблице [proshop\\_votes\\_vars](#).

Объявление:

- `function VoteVar($var_id)`

Параметры:

- **int \$var\_id**. Идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## Класс **Watermark**

Предназначен для наложения водяного знака (как графического, так и текстового) на изображение.

### **Watermark::CreateGraphical**

Назначение:

- Наложение заданного графического водяного знака (в формате PNG) на заданное изображение.

Объявление:

- `function CreateGraphical($main_img_obj, $watermark_img_obj, $alpha_level = 100, $png_type = 'png24')`

Параметры:

- **resource \$main\_img\_obj**. Объект исходного изображения, на которое необходимо наложить водяной знак. Данный объект может быть возвращен функцией **imagecreatefromjpeg** и другими подобными функциями (названия которых начинаются с **imagecreatefrom**).
- **resource \$watermark\_img\_obj**. Объект изображения водяного знака, возвращенный функцией **imagecreatefrompng**.
- **int \$alpha\_level**. Уровень прозрачности водяного знака (чем больше, тем прозрачнее). Принимает значения по шкале от 0 до 100.
- **string \$png\_type**. Тип PNG-изображения водяного знака. Может принимать значения **png8** или **png24**.

Возвращаемое значение:

- Тип **resource**. Объект изображения с наложенным водяным знаком. Данный объект можно передать, например, функции **imagejpeg**.

### **Watermark::CreateTextual**

Назначение:

- Наложение заданного текста поверх заданного изображения по диагонали.

Объявление:

- `function CreateTextual($main_img_obj, $text, $font, $r = 128, $g = 128, $b = 128, $alpha_level = 100)`

Параметры:

- **resource \$main\_img\_obj**. Объект исходного изображения, на которое необходимо наложить водяной знак. Данный объект может быть возвращен функцией **imagecreatefromjpeg** и другими подобными функциями (названия которых начинаются с **imagecreatefrom**).
- **string \$text**. Строка текста, которую необходимо наложить на изображение в качестве водяного знака.
- **string \$font**. Путь к файлу TTF-шрифта текста. В магазине используется шрифт **arial.ttf**, файл которого находится в директории **includes/fonts**.
- **int \$r**. Красная составляющая цвета шрифта (от 0 до 255).
- **int \$g**. Зеленая составляющая цвета шрифта (от 0 до 255).
- **int \$b**. Синяя составляющая цвета шрифта (от 0 до 255).
- **int \$alpha\_level**. Уровень прозрачности накладываемого текста (чем больше, тем прозрачнее). Принимает значения по шкале от 0 до 100.

Возвращаемое значение:

- Тип **resource**. Возвращается модифицированный параметр **\$main\_img\_obj** – с наложенным на изображение текстом.

## Класс WishedProducts

Предназначен для управления списком желаний (отложенных товаров) зарегистрированного покупателя.

### **WishedProducts::Add**

Назначение:

- Добавление указанного продукта в список желаний указанного зарегистрированного покупателя (в таблицу [proshop\\_products\\_wished](#)).

Объявление:

- function Add(\$prod\_id, \$user\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

### **WishedProducts::DeleteByUser**

Назначение:

- Очистка списка желаний указанного пользователя.

Объявление:

- function DeleteByUser(\$user\_id)

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **WishedProducts::DeleteForUser**

Назначение:

- Удаление указанного продукта из списка желаний (из таблицы [proshop\\_products\\_wished](#)) указанного пользователя.

Объявление:

- function DeleteForUser(\$prod\_id, \$user\_id)

Параметры:

- **int \$prod\_id**. Идентификатор продукта из таблицы [proshop\\_products](#).
- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **WishedProducts::GetAll**

Назначение:

- Извлечение списка желаний заданного пользователя из таблицы [proshop\\_products\\_wished](#).

Объявление:

- function GetAll(\$user\_id)

Параметры:

- **int \$user\_id**. Идентификатор пользователя из таблицы [proshop\\_users](#).

Возвращаемое значение:

- Тип **array**. Массив, каждый элемент которого является массивом, содержащим следующие ключи:
  - **id** – идентификатор продукта из таблицы [proshop\\_products](#);
  - **user\_id** – идентификатор пользователя из таблицы [proshop\\_users](#).

## **Класс XmlParser**

Предназначен для чтения данных формата XML в виде дерева элементов и атрибутов.

### **XmlParser::GetXMLTree**

Назначение:

- Конвертирование текста в формате XML в дерево элементов и атрибутов.

Объявление:



- function GetXMLTree(\$xml)

Параметры:

- **string \$xml**. Текст в формате XML.

Возвращаемое значение:

- Тип **array/bool**. False – в случае ошибки. Иначе - ассоциативный массив, представляющий дерево элементов и атрибутов. Чтобы узнать формат возвращаемых данных, необходимо прочитать любой XML-файл и вывести на экран возвращенное значение функции с помощью **print\_r**.

## Класс Ymarket

Предназначен для формирования YML-файла для каталога товаров Яндекс.Маркет.

### **\$yml->AddCategory**

Назначение:

- Добавление категории товаров в дерево YML-документа.

Объявление:

- function AddCategory(\$id, \$parentId, \$caption)

Параметры:

- **int \$id**. Идентификатор категории из таблицы [proshop\\_categories](#).
- **int \$parentId**. Идентификатор родительской категории (из таблицы [proshop\\_categories](#)) добавляемой категории.
- **string \$caption**. Название категории.

Возвращаемое значение:

- Функция ничего не возвращает.

### **\$yml->AddCurrency**

Назначение:

- Добавление валюты в дерево YML-документа.

Объявление:

- function AddCurrency(\$id, \$rate)

Параметры:

- **string \$id**. Международное трехбуквенное обозначение валюты. Три английские буквы верхнего регистра (например, RUR, USD, EUR).
- **float \$rate**. Курс добавляемой валюты по отношению к рублю.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$yml->AddOffer**

Назначение:

- Добавление товарного предложения в дерево YML-документа.

Объявление:

- `function AddOffer($info)`

Параметры:

- **array \$info**. Массив с информацией о товарном предложении. Может содержать следующие ключи:
  - **id** – идентификатор товара из таблицы [proshop\\_products](#). Ключ обязателен;
  - **available** – флаг доступности товара на складе, является строкой (**true** – доступен сразу для продажи, **false** – недоступен или может быть доставлен со склада при заказе). Ключ обязателен;
  - **url** – URL-адрес страницы описания товарного предложения в магазине. Ключ обязателен;
  - **price** – цена в валюте, которая указана в **currencyId**. Ключ обязателен;
  - **currencyId** – валюта (добавленная с помощью [\\$yml->AddCurrency](#)), в которой указана цена единицы товара. Ключ обязателен;
  - **categoryId** – категория товара (добавленная с помощью [\\$yml->AddCategory](#)). Ключ обязателен;
  - **picture** – URL-адрес изображения товара. Ключ не обязателен;
  - **delivery** – возможность доставить товар, является строкой (**true** – товар может быть доставлен, **false** – возможен только самовывоз). Ключ не обязателен;
  - **name** – название товара. Ключ обязателен;
  - **vendor** – название производителя товара. Ключ не обязателен;
  - **description** – описание товара. Ключ не обязателен;
  - **downloadable** – возможность скачать товар, является строкой (**true** – товар цифровой, **false** – товар обычный). Ключ не обязателен.

Возвращаемое значение:

- Функция ничего не возвращает.

## **\$yml->Escape**

Назначение:

- Подготовка текстовой информации для помещения ее в качестве значения YML-элемента.

Объявление:

- `function Escape($str, $len = 0)`

Параметры:

- **string \$str**. Строка с текстом, которую необходимо отфильтровать.

- **int \$len.** Максимально допустимая длина строки. Если длина строки превышает данное значение, она обрезается и в конце ставится многоточие. Если значение данного параметра равно 0, данный параметр считается не заданным.

Возвращаемое значение:

- Тип **string**. Отфильтрованная строка **\$str**.

## **\$yaml->GetYML**

Назначение:

- Формирование XML-содержимого YML-документа, подготовленного путем вызовов методов класса.

Объявление:

- function GetYML()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Содержимое YML-документа в кодировке windows-1251.

## **\$yaml->GetYMLFileName**

Назначение:

- Извлечение имени файла YML-документа, который должен быть сформирован классом.

Объявление:

- function GetYMLFileName()

Параметры:

- Функция не принимает параметров.

Возвращаемое значение:

- Тип **string**. Имя XML-файла YML-документа.

## **\$yaml->Save**

Назначение:

- Сохранение YML-содержимого в указанный файл. Если файл существует, он будет перезаписан.

Объявление:

- function Save(\$yaml, \$file)

Параметры:

- **string \$yaml.** Текстовое содержимое, которое необходимо записать в файл.
- **string \$file.** Полный или относительный путь к сохраняемому файлу.

Возвращаемое значение:

- Тип **bool**. True/false – в зависимости от успешности операции.

## **`$yml->SetCompany`**

Назначение:

- Установка названия организации в YML-документе, которой принадлежит магазин.

Объявление:

- `function SetCompany($company)`

Параметры:

- **string \$company**. Название организации.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$yml->SetDate`**

Назначение:

- Установка даты и времени генерации YML-документа.

Объявление:

- `function SetDate($date)`

Параметры:

- **string \$date**. Дата и время в формате ГГГГ-ММ-ДД ЧЧ:ММ.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$yml->SetShopName`**

Назначение:

- Установка названия магазина.

Объявление:

- `function SetShopName($shop_name)`

Параметры:

- **string \$shop\_name**. Название магазина. Макс. 20 символов.

Возвращаемое значение:

- Функция ничего не возвращает.

## **`$yml->SetSiteUrl`**

Назначение:

- Установка URL-адреса главной страницы магазина в YML-документе.

Объявление:

- `function SetSiteUrl($site_url)`

Параметры:

- **string \$site\_url**. URL-адрес главной страницы магазина.

Возвращаемое значение:

- Функция ничего не возвращает.

## 8.7 Таблицы и поля БД

### proshop\_access\_ip

Назначение:

- Хранение IP-адресов, вход в магазин с которых запрещен.

Структура:

| Поле           | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)     |                 |          | Нет  | Нет          | auto_increment |
| <b>ip</b>      | varchar(15) | utf8_general_ci |          | Нет  |              |                |
| <b>blocked</b> | char(1)     | utf8_general_ci |          | Нет  |              |                |
| <b>created</b> | date        |                 |          | Нет  | 0000-00-00   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 1                    | A         |      |
| ip          | BTREE | Нет        | Нет      | ip      | 0                    | A         |      |
|             |       |            |          | blocked | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **ip** – IP-адрес.
- **blocked** – заблокирован ли доступ с IP-адреса **ip** (**Y** – заблокирован, **N** - нет).
- **created** – дата создания записи.

### proshop\_access\_ip\_texts

Назначение:

- Хранение текстов причин блокировки IP-адресов на разных языках.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>lang_id</b>   | int(11)      |                 |          | Нет  | 0            |                |
| <b>access_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>reason</b>    | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id        | 1                    | A         |      |
| lang_id     | BTREE | Да         | Нет      | lang_id   | 0                    | A         |      |
|             |       |            |          | access_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.

- **lang\_id** – идентификатор языка, на котором указана причина **reason** (вторичный ключ, указывающий на таблицу [proshop\\_languages](#)).
- **access\_id** – вторичный ключ, указывающий на таблицу [proshop\\_access\\_ip](#).
- **reason** – причина блокировки IP-адреса, указанного в таблице [proshop\\_access\\_ip](#).

## proshop\_admins\_allowed\_actions

Назначение:

- Хранение связей групп администраторов с действиями в модулях. Данные связи формируют набор разрешенных действий в различных модулях, которые могут выполнять администраторы, принадлежащие определенным группам.

Структура:

| Поле                  | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>             | int(11) |           |          | Нет  | Нет          | auto_increment |
| <b>admin_group_id</b> | int(11) |           |          | Нет  | Нет          |                |
| <b>action_id</b>      | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **admin\_group\_id** – идентификатор группы администраторов (вторичный ключ, указывающий на таблицу [proshop\\_admins\\_groups](#)).
- **action\_id** – идентификатор разрешенного действия какого-либо модуля (вторичный ключ, указывающий на таблицу [proshop\\_modules\\_admin\\_actions](#)) для указанной группы администраторов.

## proshop\_admins\_groups

Назначение:

- Хранение групп администраторов вместе с их названиями.

Структура:

| Поле        | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>   | int(11)     |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b> | varchar(64) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.

- **name** – название группы администраторов, не зависящее от языка.

## proshop\_admins\_restricted\_modules

Назначение:

- Хранение связей групп администраторов с модулями CMS, формируя таким образом набор модулей, доступ к которым запрещен для указанных групп администраторов.

Структура:

| Поле                  | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>             | int(11) |           |          | Нет  | Нет          | auto_increment |
| <b>admin_group_id</b> | int(11) |           |          | Нет  | Нет          |                |
| <b>module_id</b>      | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 38                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **admin\_group\_id** - идентификатор группы администраторов (вторичный ключ, указывающий на таблицу [proshop\\_admins\\_groups](#)).
- **module\_id** – идентификатор запрещенного модуля (вторичный ключ, указывающий на таблицу [proshop\\_modules](#)) для указанной группы администраторов.

## proshop\_admin\_menus

Назначение:

- Хранение служебных названий, позиций и текстовых меток названий блоков меню админки.

Структура:

| Поле         | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>    | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>pos</b>   | int(11)      |                 |          | Нет  | 0            |                |
| <b>label</b> | varchar(64)  | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 5                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное название блока меню админки.
- **pos** – позиция блока меню по отношению к другим блокам для сортировки.

- **label** – служебное имя текстовой метки из таблицы [proshop\\_labels](#), которая определяет отображаемое название блока в админке.

## proshop\_adv\_links

Назначение:

- Хранение ссылок на внешние ресурсы, отображаемых на витрине магазина. Используется модулем **adv\_links**.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>url</b>        | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>new_window</b> | char(1)      | utf8_general_ci |          | Нет  |              |                |
| <b>pos</b>        | tinyint(4)   |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **url** – URL ссылки.
- **new\_window** – флаг открытия страницы, на которую указывает ссылка, в новом окне (или вкладке) при переходе по этой ссылке (**Y** – открывать в новом окне, **N** – в том же окне).
- **pos** – позиция ссылки относительно других ссылок для сортировки.

## proshop\_adv\_links\_texts

Назначение:

- Хранение текстов ссылок, записанных в таблице [proshop\\_adv\\_links](#), на разных языках.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>link_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>title</b>   | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 4                    | A         |      |
| link_id     | BTREE | Да         | Нет      | link_id | 0                    | A         |      |
|             |       |            |          | lang_id | 4                    | A         |      |

Поля:



- **id** – уникальный идентификатор записи.
- **link\_id** – идентификатор ссылки (вторичный ключ, указывающий на таблицу [proshop\\_adv\\_links](#)), текст которой хранится в данной записи на языке **lang\_id**.
- **lang\_id** – идентификатор языка, на котором указан текст **title** (вторичный ключ, указывающий на таблицу [proshop\\_languages](#)).
- **title** – текст ссылки на языке **lang\_id**, которая указана в **link\_id**.

## proshop\_articles

Назначение:

- Хранение базовой информации о статьях (информационных страницах) магазина.

Структура:

| Поле                | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>           | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>url_name</b>     | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>show_in_list</b> | varchar(1)   | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса     | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-----------------|-------|------------|----------|----------|----------------------|-----------|------|
| <b>PRIMARY</b>  | BTREE | Да         | Нет      | id       | 2                    | A         |      |
| <b>url_name</b> | BTREE | Нет        | Нет      | url_name | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **url\_name** – имя в URL статьи (для modRewrite).
- **show\_in\_list** – флаг показа статьи в списке статей на витрине (Y – показывать статью в списке статей в модуле на витрине, N – не показывать).

## proshop\_articles\_categories

Назначение:

- Хранение информации о дереве категорий (рубрик) статей.

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>cat_id</b>   | int(11)      |                 |          | Нет  | Нет          |                |
| <b>path</b>     | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>url_name</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>pos</b>      | int(11)      |                 |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 6                    | A         |      |
| cat_id      | BTREE | Нет        | Нет      | cat_id   | 0                    | A         |      |
| url_name    | BTREE | Нет        | Нет      | url_name | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cat\_id** – идентификатор родительской категории статей для данной категории из этой же таблицы.
- **path** – путь к данной категории статей, состоящий из идентификаторов родительских категорий данной категории, объединенных дефисом (например, 10-20-30, где 30 - идентификатор данной категории).
- **url\_name** – имя в URL (для modRewrite) данной категории статей.
- **pos** – позиция данной категории по отношению к другим категориям того же уровня.

## proshop\_articles\_categories\_texts

Назначение:

- Хранение названий категорий (рубрик) статей, которые хранятся в таблице [proshop\\_articles\\_categories](#), на разных языках.

Структура:

| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| cat_id    | int(11)      |                 |          | Нет  | Нет          |                |
| caption   | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| lang_id   | int(11)      |                 |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 12                   | A         |      |
| cat_id      | BTREE | Да         | Нет      | cat_id  | 0                    | A         |      |
|             |       |            |          | lang_id | 12                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cat\_id** – идентификатор категории (рубрики) из таблицы [proshop\\_articles\\_categories](#).
- **caption** – название категории **cat\_id** на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#).

## proshop\_articles\_products

Назначение:

- Хранение связей статей и продуктов.

Структура:

| Поле       | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------|---------|-----------|----------|------|--------------|----------------|
| <u>id</u>  | int(11) |           |          | Нет  | Нет          | auto_increment |
| article_id | int(11) |           |          | Нет  | Нет          |                |
| rprod_id   | int(11) |           |          | Нет  | Нет          |                |
| pos        | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id         | 3                    | A         |      |
| article_id  | BTREE | Нет        | Нет      | article_id | 0                    | A         |      |
| rprod_id    | BTREE | Нет        | Нет      | rprod_id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **article\_id** – идентификатор статьи из таблицы [proshop\\_articles](#).
- **rprod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **pos** – позиция продукта по отношению к другим продуктам, привязанным к статье **article\_id**.

## proshop\_articles\_texts

Назначение:

- Хранение текстов статей, определенных в таблице [proshop\\_articles](#), на разных языках.

Структура:

| Поле       | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------|---------|-----------------|----------|------|--------------|----------------|
| <u>id</u>  | int(11) |                 |          | Нет  | Нет          | auto_increment |
| article_id | int(11) |                 |          | Нет  | 0            |                |
| lang_id    | int(11) |                 |          | Нет  | 0            |                |
| caption    | text    | utf8_general_ci |          | Нет  | Нет          |                |
| content    | text    | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id         | 4                    | A         |      |
| article_id  | BTREE | Да         | Нет      | article_id | 0                    | A         |      |
|             |       |            |          | lang_id    | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **article\_id** – идентификатор статьи из таблицы [proshop\\_articles](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#).
- **caption** – название статьи на языке **lang\_id**.
- **content** – содержимое статьи на языке **lang\_id**.

## proshop\_banners

Назначение:

- Хранение информации о баннерах. Используется модулем **banners**.

Структура:

| Поле                   | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>              | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>            | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>lang_id</b>         | int(11)      |                 |          | Нет  | Нет          |                |
| <b>filename</b>        | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>type</b>            | varchar(5)   | utf8_general_ci |          | Нет  | Нет          |                |
| <b>href</b>            | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>width</b>           | int(11)      |                 |          | Нет  | Нет          |                |
| <b>height</b>          | int(11)      |                 |          | Нет  | Нет          |                |
| <b>begin_timestamp</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>end_timestamp</b>   | int(11)      |                 |          | Нет  | 0            |                |
| <b>place_id</b>        | int(11)      |                 |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 0                    | A         |      |
| lang_id     | BTREE | Нет        | Нет      | lang_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – название баннера на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), для которого предназначен баннер.
- **filename** – имя файла баннера (изображения, swf-файла или html-файла), который хранится в директории, прописанной в константе [BANNERS\\_PATH](#).
- **type** – тип баннера. Может принимать значения: **img**, **flash**, **html**.
- **href** – URL страницы, на которую ведет баннер при нажатии на него.
- **width** – ширина изображения баннера в пикселях.
- **height** – высота изображения баннера в пикселях.
- **begin\_timestamp** – метка времени Unix, с наступлением которого показ баннера должен быть начат. Если равна нулю, баннер не будет показан (показ временно отключен).
- **end\_timestamp** – метка времени Unix, с наступлением которого показ баннера должен быть завершен. Если равна нулю, время завершения показа баннера считается неопределенным.
- **place\_id** – идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#), которому принадлежит баннер. Если равен нулю, баннер не принадлежит ни одному баннероместу.

## proshop\_banners\_clicks

Назначение:

- Хранение информации о нажатиях на баннеры (переходах) посетителями.

Структура:

| Поле              | Тип      | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|----------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)  |                 |          | Нет  | Нет          | auto_increment |
| <b>banner_id</b>  | int(11)  |                 |          | Нет  | Нет          |                |
| <b>page</b>       | text     | utf8_general_ci |          | Нет  | Нет          |                |
| <b>visitor_id</b> | int(11)  |                 |          | Нет  | Нет          |                |
| <b>timestamp</b>  | datetime |                 |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **banner\_id** – идентификатор баннера из таблицы [proshop\\_banners](#).
- **page** – адрес страницы магазина, на которой посетитель **visitor\_id** нажал на баннер **banner\_id**.
- **visitor\_id** – идентификатор посетителя из таблицы [proshop\\_visitors](#), который нажал на баннер.
- **timestamp** – метка времени того момента, когда произошел переход по баннеру.

## proshop\_banners\_pages

Назначение:

- Хранение адресов страниц магазина, на которых расположено определенное баннероместо, т.е. хранение всех страниц расположения всех баннеромест.

Структура:

| Поле            | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|---------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11) |                 |          | Нет  | Нет          | auto_increment |
| <b>place_id</b> | int(11) |                 |          | Нет  | Нет          |                |
| <b>page</b>     | text    | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса      | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|------------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY          | BTREE | Да         | Нет      | id       | 0                    | A         |      |
| <b>banner_id</b> | BTREE | Нет        | Нет      | place_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.

- **place\_id** – идентификатор баннероместа (из таблицы [proshop\\_banners\\_places](#)), который привязан к странице с адресом, заданным в **page**.
- **page** – адрес страницы, к которой привязано баннероместо с идентификатором **place\_id**.

## proshop\_banners\_places

Назначение:

- Хранение баннеромест, к которым привязаны баннеры, информация о которых хранится в таблице [proshop\\_banners](#).

Структура:

|                      |              |                 |  |     |     |                |
|----------------------|--------------|-----------------|--|-----|-----|----------------|
| <b>id</b>            | int(11)      |                 |  | Нет | Нет | AUTO_INCREMENT |
| <b>name</b>          | varchar(254) | utf8_general_ci |  | Нет | Нет |                |
| <b>lang_id</b>       | int(11)      |                 |  | Нет | Нет |                |
| <b>location</b>      | varchar(2)   | utf8_general_ci |  | Нет | Нет |                |
| <b>only_homepage</b> | varchar(1)   | utf8_general_ci |  | Нет | N   |                |
| <b>not_homepage</b>  | varchar(1)   | utf8_general_ci |  | Нет | N   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – название баннероместа на языке **lang\_id**.
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), для которого действует данное баннероместо.
- **location** – расположение баннероместа на странице. Возможные значения: **L** – левая колонка, **R** – правая колонка, **CT** – над центральной частью, **CB** – под центральной частью.
- **only\_homepage** – если значение данной колонки равно **Y**, баннероместо должен отображаться только на главной странице витрины.
- **not\_homepage** – если значение данной колонки равно **Y**, баннероместо не должен отображаться на главной странице витрины.

## proshop\_banners\_shows

Назначение:

- Хранение статистики показов баннеров.

Структура:

| Поле              | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>         | int(11) |           |          | Нет  | Нет          | auto_increment |
| <b>banner_id</b>  | int(11) |           |          | Нет  | Нет          |                |
| <b>visitor_id</b> | int(11) |           |          | Нет  | Нет          |                |
| <b>count</b>      | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса        | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|--------------------|-------|------------|----------|------------|----------------------|-----------|------|
| <b>PRIMARY</b>     | BTREE | Да         | Нет      | id         | 0                    | A         |      |
| <b>banner_id</b>   | BTREE | Да         | Нет      | banner_id  | 0                    | A         |      |
|                    |       |            |          | visitor_id | 0                    | A         |      |
| <b>banner_id_2</b> | BTREE | Нет        | Нет      | banner_id  | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **banner\_id** – идентификатор баннера из таблицы [proshop\\_banners](#).
- **visitor\_id** – идентификатор посетителя из таблицы [proshop\\_visitors](#), который просмотрел баннер **banner\_id**.
- **count** – кол-во просмотров баннера **banner\_id** посетителем **visitor\_id**.

## proshop\_cache\_data

Назначение:

- Хранение кэшированной информации. Данная таблица используется механизмом кэша CMS (классом [Cache](#) в частности).

Структура:

| Поле          | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>     | int(11)     |                 |          | Нет  | Нет          | auto_increment |
| <b>key</b>    | varchar(50) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>key_id</b> | varchar(32) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>data</b>   | mediumtext  | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле   | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|--------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id     | 4                    | A         |      |
| <b>key</b>     | BTREE | Да         | Нет      | key    | 0                    | A         |      |
|                |       |            |          | key_id | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **key** – ключ кэшированной информации.
- **key\_id** – вспомогательный ключ (дополнение к основному ключу **key**). Чаще всего это идентификатор языка, для которого сохраняется кэш (но не всегда).
- **data** – данные кэша в сериализованном формате (полученном функцией **serialize**).

## proshop\_cache\_times

Назначение:

- Хранение меток времени Unix для контроля актуальности кэшированной информации.

Структура:

| Поле             | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)     |                 |          | Нет  | Нет          | auto_increment |
| <b>key</b>       | varchar(50) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>timestamp</b> | int(11)     |                 |          | Да   | NULL         |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 21                   | A         |      |
| key         | BTREE | Да         | Нет      | key  | 21                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **key** – ключ кэшированной информации.
- **timestamp** – метка времени Unix актуализации кэша с указанным ключом **key**. Если при загрузке страницы магазина механизм кэша обнаружит, что метка времени **timestamp** больше, чем время последнего сохранения кэша, кэш будет считаться недействительным и будет обновлен. Таким образом, кэш всегда поддерживается в актуальном состоянии.

## proshop\_carts

Назначение:

- Хранение временной информации о корзинах покупателей для того, чтобы впоследствии очистить просроченные корзины и вернуть товар на склад, зарезервированный в результате его добавления в корзину.

Структура:

| Поле                  | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>             | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>prod_id</b>        | int(11)      |                 |          | Нет  | Нет          |                |
| <b>variant_id</b>     | int(11)      |                 |          | Нет  | Нет          |                |
| <b>options</b>        | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>qty</b>            | int(11)      |                 |          | Нет  | Нет          |                |
| <b>egood_pin_code</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>datetime</b>       | datetime     |                 |          | Нет  | Нет          |                |
| <b>cart_id</b>        | int(11)      |                 |          | Нет  | Нет          |                |
| <b>session_id</b>     | varchar(60)  | utf8_general_ci |          | Нет  | Нет          |                |
| <b>visitor_id</b>     | int(11)      |                 |          | Нет  | 0            |                |

Индексы:



| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 0                    | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **variant\_id** – идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#). Данное поле должно содержать нулевое значение, если продукт был добавлен без варианта.
- **options** – идентификаторы вариантов опций продукта (из таблицы [proshop\\_products\\_options\\_vars](#)), разделенные дефисом, или пустая строка, если продукт был добавлен в корзину без опций (продукт не имеет опций).
- **qty** – кол-во экземпляров продукта в корзине.
- **egood\_pin\_code** – секретный код цифрового продукта **prod\_id**. Данное поле не имеет значения, если продукт **prod\_id** не является цифровым, или продукт является цифровым, но не предназначен для продажи секретных кодов (пин-кодов). См. также про таблицу [proshop\\_products\\_pins](#).
- **datetime** – дата и время момента добавления продукта в корзину.
- **cart\_id** – номер корзины в пределах сессии с идентификатором **session\_id**.
- **session\_id** – идентификатор сессии, во время которой продукт был добавлен в корзину.
- **visitor\_id** – идентификатор посетителя (из таблицы [proshop\\_visitors](#)), который добавил продукт в корзину.

## proshop\_categories

Назначение:

- Хранение базовой информации о категориях магазина.

Структура:

| Поле                          | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>                     | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>cat</b>                    | int(11)      |                 |          | Нет  | 0            |                |
| <b>path</b>                   | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>pos</b>                    | int(11)      |                 |          | Нет  | 0            |                |
| <b>img</b>                    | varchar(254) | utf8_general_ci |          | Да   | <i>NULL</i>  |                |
| <b>eng_name</b>               | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>VISIBLE</b>                | char(1)      | utf8_general_ci |          | Нет  | Y            |                |
| <b>tax_id</b>                 | int(11)      |                 |          | Нет  | 0            |                |
| <b>discount_id</b>            | int(11)      |                 |          | Нет  | 0            |                |
| <b>prods_num</b>              | int(11)      |                 |          | Нет  | 0            |                |
| <b>visible_prods_num</b>      | int(11)      |                 |          | Нет  | 0            |                |
| <b>this_prods_num</b>         | int(11)      |                 |          | Нет  | 0            |                |
| <b>this_visible_prods_num</b> | int(11)      |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип      | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|----------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE    | Да         | Нет      | id   | 35                   | A         |      |
| <b>cat</b>     | BTREE    | Нет        | Нет      | cat  | 0                    | A         |      |
| <b>path</b>    | FULLTEXT | Нет        | Нет      | path | 0                    |           |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cat** – идентификатор родительской категории (из этой же таблицы **proshop\_categories**). Если значение данного поля равно 0, текущая категория является корневой.
- **path** – путь к текущей категории в виде: 120-121-122 – здесь 120 и 121 – идентификаторы родительских категорий, а 122 – идентификатор текущей категории. При этом категория с идентификатором 120 является родительской по отношению к категории 121, а категория 121 – родительской по отношению к текущей категории 122. Для корневой категории **path** должен совпадать с **id**.
- **pos** – позиция текущей категории по отношению к остальным категориям текущего уровня вложенности.
- **img** – имя файла изображения категории. Изображения категорий хранятся в директории, путь к которой прописан в константе [CATS\\_PICS](#). Если у категории изображения нет, данное поле должно быть пустым.
- **eng\_name** – имя в URL (для modRewrite) текущей категории. Если имя не задано, поле должно быть пустым.
- **VISIBLE** – флаг видимости категорий и всех вложенных в нее категорий и продуктов (Y – видима, N - невидима) на витрине.
- **tax\_id** – идентификатор налога (из таблицы [proshop\\_taxes](#)) по умолчанию для всех продуктов, принадлежащих текущей категории и всем вложенным в нее категориям.

- **discount\_id** – идентификатор скидки (из таблицы [proshop\\_discounts](#)) по умолчанию для всех продуктов, принадлежащих текущей категории и всем вложенным в нее категориям.
- **prods\_num** – кол-во всех продуктов, принадлежащих текущей категории и всем ее подкатегориям.
- **visible\_prods\_num** – кол-во всех видимых на витрине продуктов из числа **prods\_num**, принадлежащих текущей категории и всем ее подкатегориям.
- **this\_prods\_num** – кол-во всех продуктов, принадлежащих только текущей категории, не включая ее подкатегории.
- **this\_visible\_prods\_num** – кол-во всех видимых на витрине продуктов из числа **this\_prods\_num**, принадлежащих только текущей категории, не включая ее подкатегории.

## proshop\_categories\_params

Назначение:

- Хранение базовой информации обо всех параметрах категорий (характеристики продуктов).

Структура:

| Поле              | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|---------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11) |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>cat_id</b>     | int(11) |                 |          | Нет  | 0            |                |
| <b>is_numeric</b> | char(1) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле   | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|--------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id     | 14                   | A         |      |
| <b>cat_id</b>  | BTREE | Нет        | Нет      | cat_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cat\_id** – идентификатор категории (из таблицы [proshop\\_categories](#)), параметр которой описан в текущей записи.
- **is\_numeric** – флаг, определяющий, является ли параметр числовым (**Y** – параметр числовой, **N** – параметр строковый).

## proshop\_categories\_params\_texts

Назначение:

- Хранение названий параметров категорий (характеристик продуктов), базовая информация о которых задана в таблице [proshop\\_categories\\_params](#).

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>param_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса     | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-----------------|-------|------------|----------|----------|----------------------|-----------|------|
| <b>PRIMARY</b>  | BTREE | Да         | Нет      | id       | 28                   | A         |      |
| <b>param_id</b> | BTREE | Да         | Нет      | param_id | 0                    | A         |      |
|                 |       |            |          | lang_id  | 28                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **param\_id** – идентификатор параметра категории (характеристики продуктов) из таблицы [proshop\\_categories\\_params](#).
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), на котором задано название параметра в поле **caption**.
- **caption** – название параметра на языке **lang\_id**.

## proshop\_categories\_paths\_texts

Назначение:

- Хранение путей к категориям товаров на разных языках.

Структура:

| Поле           | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|---------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11) |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>cat_id</b>  | int(11) |                 |          | Нет  | <i>Нет</i>   |                |
| <b>lang_id</b> | int(11) |                 |          | Нет  | <i>Нет</i>   |                |
| <b>path</b>    | text    | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 72                   | A         |      |
| <b>cat_id</b>  | BTREE | Да         | Нет      | cat_id  | 36                   | A         |      |
|                |       |            |          | lang_id | 72                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cat\_id** – идентификатор категории из таблицы [proshop\\_categories](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#).
- **path** – путь к категории **cat\_id** на языке **lang\_id**.

## proshop\_categories\_texts

Назначение:

- Хранение разноязычных названий и описаний категорий, базовая информация о которых задана в таблице [proshop\\_categories](#).

Структура:

| Поле               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>          | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>category_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>     | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>descr</b>       | text         | utf8_general_ci |          | Да   | NULL         |                |
| <b>alt</b>         | varchar(254) | utf8_general_ci |          | Да   | NULL         |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id          | 70                   | A         |      |
| category_id | BTREE | Да         | Нет      | category_id | 0                    | A         |      |
|             |       |            |          | lang_id     | 70                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **category\_id** – идентификатор категории (из таблицы [proshop\\_categories](#)), для которой в текущей записи хранятся разноязычные текстовые составляющие (название, описание, ...).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), для которого хранятся текстовые составляющие категории - **caption**, **descr** и **alt**.
- **caption** – название категории **category\_id** на языке **lang\_id**.
- **descr** – описание категории **category\_id** на языке **lang\_id**.
- **alt** – альтернативный текст изображения категории **category\_id** на языке **lang\_id**.

## proshop\_cblocks

Назначение:

- Хранение информации о боковых блоках и центральных блоках главной страницы витрины магазина.

Структура:

| Поле                   | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>              | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>name</b>            | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>type</b>            | varchar(1)   | utf8_general_ci |          | Да   | <i>NULL</i>  |                |
| <b>elem_id</b>         | int(11)      |                 |          | Нет  | 0            |                |
| <b>label</b>           | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>column</b>          | varchar(1)   | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>column_fixed</b>    | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |
| <b>weight</b>          | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| <b>in_homepage</b>     | varchar(1)   | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>not_in_homepage</b> | varchar(1)   | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>module</b>          | varchar(32)  | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>ctheme</b>          | varchar(32)  | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 21                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя блока.
- **type** – флаг того, что блок является баннероместом. Если значение данного поля равно **B**, блок является баннероместом. Для остальных блоков значение данного поля должно быть NULL.
- **elem\_id** – идентификатор баннероместа из таблицы [proshop\\_banners\\_places](#), если значение поля **type** равно **B**. Иначе – должно быть равно 0.
- **label** – текстовая метка из таблицы [proshop\\_labels](#), в которой хранится название текущего блока, выводимое в админке в списке блоков при их просмотре и изменении их местоположений.
- **column** – обозначение колонки, в которой располагается блок на витрине. Может принимать значения: **L** – левая колонка, **C** – центральная колонка главной страницы, **R** – правая колонка.
- **column\_fixed** – флаг, определяющий, можно ли менять колонку расположения блока (**Y** – менять нельзя, **N** – менять можно).
- **weight** – «вес» блока по отношению к другим блокам той же колонки. Определяет расположение блока относительно других блоков по вертикали.
- **in\_homepage** – флаг, определяющий, должен ли отображаться блок на главной странице витрины магазина (**Y** – отображать, **N** – не отображать).
- **not\_in\_homepage** – флаг, определяющий, должен ли отображаться блок на страницах витрины магазина, отличных от главной (**Y** – отображать, **N** – не отображать). Значение данного флага игнорируется для блоков, **column** которых равен **C**.

- **module** – название модуля (из таблицы [proshop\\_modules](#)), которому принадлежит текущий блок. Если блок не принадлежит ни одному из модулей, значение данного поля должно быть пустым.
- **ctheme** – название темы оформления, для которой задано положение данного блока.

## proshop\_cities

Значение:

- Хранение списка городов стран мира.

Структура:

| Поле              | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)     |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>name</b>       | varchar(80) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>country_id</b> | int(11)     |                 |          | Нет  | Нет          |                |
| <b>code_edost</b> | varchar(10) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса       | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY           | BTREE | Да         | Нет      | id         | 1020                 | A         |      |
| <b>name</b>       | BTREE | Нет        | Нет      | name       | 0                    | A         |      |
| <b>country_id</b> | BTREE | Нет        | Нет      | country_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – название города.
- **country\_id** – идентификатор страны (из таблицы [proshop\\_countries](#)), которой принадлежит город.
- **code\_edost** – код города для сервиса расчета стоимости доставки eDost.

## proshop\_countries

Значение:

- Хранение списка стран мира.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>name</b>       | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>code</b>       | varchar(8)   | utf8_general_ci |          | Нет  |              |                |
| <b>code_edost</b> | varchar(10)  | utf8_general_ci |          | Да   | NULL         |                |
| <b>pos</b>        | int(11)      |                 |          | Нет  | Нет          |                |
| <b>avail</b>      | char(1)      | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 245                  | A         |      |
| code        | BTREE | Нет        | Нет      | code | 245                  | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – название страны.
- **code** – международное двухбуквенное обозначение страны.
- **code\_edost** – код страны для сервиса расчета стоимости доставки eDost.
- **pos** – позиция в списке относительно других стран. Используется для сортировки.
- **avail** – флаг доступности для выбора данной страны в модулях магазина (**Y** – выбор данной страны включен, **N** – выбор выключен).

## proshop\_cshippings

Назначение:

- Хранение базовой информации о способах доставки типа **custom**. Таким способом доставки является Почта России.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u>      | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| name           | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| delivery_time1 | tinyint(3)   |                 | UNSIGNED | Нет  | 0            |                |
| delivery_time2 | tinyint(3)   |                 | UNSIGNED | Нет  | 0            |                |
| is_active      | varchar(1)   | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное название способа доставки, которое используется для подключения PHP-файла службы доставки, находящегося в директории [shippings](#).
- **delivery\_time1** – начальное значение диапазона возможного срока доставки, в днях.
- **delivery\_time2** – конечное значение диапазона возможного срока доставки, в днях.
- **is\_active** – флаг активности данного метода доставки: **Y** – активен, **N** – неактивен.

## proshop\_cshippings\_texts

Назначение:

- Хранение названий служб доставки типа **custom**, определенных в таблице [proshop\\_cshippings](#).



Структура:

| Поле                | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>           | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>cshipping_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>      | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>      | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса  | Тип   | Уникальный | Упакован | Поле         | Уникальных элементов | Сравнение | Null |
|--------------|-------|------------|----------|--------------|----------------------|-----------|------|
| PRIMARY      | BTREE | Да         | Нет      | id           | 2                    | A         |      |
| cshipping_id | BTREE | Да         | Нет      | cshipping_id | 0                    | A         |      |
|              |       |            |          | lang_id      | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cshipping\_id** – идентификатор службы доставки из таблицы [proshop\\_cshippings](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано название службы доставки в **caption**.
- **caption** – название службы доставки на языке **lang\_id**.

## proshop\_currencies

Назначение:

- Хранение информации о валютах, используемых в магазине.

Структура:

| Поле                 | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>            | int(11)     |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>name</b>          | varchar(3)  | utf8_general_ci |          | Нет  |              |                |
| <b>short_name</b>    | varchar(32) | utf8_general_ci |          | Нет  |              |                |
| <b>left_side</b>     | char(1)     | utf8_general_ci |          | Нет  |              |                |
| <b>decimals</b>      | tinyint(4)  |                 |          | Нет  | <i>Нет</i>   |                |
| <b>dec_point</b>     | varchar(1)  | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>thousands_sep</b> | varchar(1)  | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>course</b>        | varchar(32) | utf8_general_ci |          | Нет  |              |                |
| <b>is_base</b>       | char(1)     | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – международное трехбуквенное обозначение валюты (например, RUR, EUR, USD).

- **short\_name** – короткое обозначение валюты, отображаемое возле значения цены (суммы) слева или справа, в зависимости от значения поля **left\_side**. Например, 100 р. или \$100.
- **left\_side** – флаг, определяющий, на какой стороне от значения цены (суммы) необходимо отображать короткое обозначение валюты: **Y** – отображать на левой стороне, **N** – на правой. Например, значение суммы в долларах чаще всего отображают со значком \$ с левой стороны от значения суммы: \$100.
- **decimals** – кол-во цифр после запятой при отображении суммы в данной валюте. Может принимать значения **0** или **2**.
- **dec\_point** – символ-разделитель дробной части (запятая или точка) при отображении суммы в данной валюте.
- **thousands\_sep** – разделитель тысячных групп разрядов (чаще всего это пустая строка, пробел или символ апострофа) при отображении суммы в данной валюте. Например, если это пробел, то миллион рублей будет отображаться следующим образом: 1 000 000 р.
- **course** – курс валюты по отношению к базовой валюте (валюте, значение поля **is\_base** которой равно **Y**). Может содержать до 7-ми знаков после точки.
- **is\_base** – флаг, определяющий, является ли текущая валюта базовой валютой магазина (**Y** – является, **N** – не является). В таблице может быть только одна базовая валюта, в которой прописаны все цены и суммы в других таблицах CMS.

## proshop\_cthemes

Назначение:

- Хранение названий тем оформления витрины.

Структура:

| Поле           | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)     |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>theme</b>   | varchar(32) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>caption</b> | varchar(32) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **theme** – служебное название темы оформления витрины.
- **caption** – отображаемое название темы оформления витрины.

## proshop\_cthemes\_colors

Назначение:

- Хранение информации о цветовых схемах различных тем оформления витрины магазина, определенных в таблице [proshop\\_cthemes](#).

Структура:

| Поле         | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>    | int(11)     |                 |          | Нет  | Нет          | auto_increment |
| <b>theme</b> | varchar(32) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>color</b> | varchar(32) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле  | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|-------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id    | 12                   | A         |      |
| <b>theme</b>   | BTREE | Нет        | Нет      | theme | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **theme** – название темы оформления.
- **color** – название цветовой схемы темы оформления **theme**.

## proshop\_discounts

Назначение:

- Хранение информации о скидках разных типов (для пользователей, для продуктов, глобальных скидок).

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>type</b>      | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>elem_type</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>value</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>active</b>    | char(1)      | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **type** – тип скидки (абсолютная или в процентах). Может принимать значения: **per** – размер скидки **value** указан в процентах, **abs** – размер скидки **value** указан в базовой валюте магазина.
- **elem\_type** – идентификатор типа скидки из таблицы [proshop\\_discounts\\_types](#).
- **value** – величина скидки.

- **active** – флаг активности данной скидки: **Y** – скидка доступна и активна, **N** – скидка временно отключена.

## proshop\_discounts\_cart

Назначение:

- Хранение диапазонов сумм заказов и скидок на заказ (разовых и накопительных) в абсолютных и относительных величинах.

Структура:

| Поле            | Тип           | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|---------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)       |                 |          | Нет  | Нет          | auto_increment |
| <b>low_band</b> | decimal(12,2) |                 |          | Нет  | Нет          |                |
| <b>discount</b> | decimal(12,2) |                 |          | Нет  | Нет          |                |
| <b>per</b>      | decimal(12,1) |                 |          | Нет  | Нет          |                |
| <b>accum</b>    | varchar(1)    | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 4                    | A         |      |

Поле:

- **id** – уникальный идентификатор записи.
- **low\_band** – нижняя граница диапазона суммы заказа в базовой валюте, с которой начинает действовать скидка.
- **discount** – абсолютное значение скидки в базовой валюте.
- **per** – относительное значение скидки – в процентах от суммы заказа.
- **accum** – если значение данного поля равно **Y**, скидка является накопительной.

## proshop\_discounts\_conf

Назначение:

- Хранение поведения магазина при наложении нескольких скидок: складывать ли все скидки или выбирать из всех максимальную.

Структура:

| Поле          | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>     | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>   | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>active</b> | char(1)      | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – код поведения магазина при наложении скидок. Может принимать следующие значения: **sum\_of\_all** – суммировать все скидки при их наложении, **max\_of\_all** – выбирать из всех пересекающихся скидок максимальную.
- **active** – флаг активности поведения: **Y** – поведение активно, **N** – не активно. В таблице может храниться только одно активное поведение.

## proshop\_discounts\_texts

Назначение:

- Хранение названий скидок, определенных в таблице [proshop\\_discounts](#), на разных языках.

Структура:

| Поле               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>          | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>discount_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>     | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id          | 6                    | A         |      |
| discount_id | BTREE | Да         | Нет      | discount_id | 0                    | A         |      |
|             |       |            |          | lang_id     | 6                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **discount\_id** – идентификатор скидки (из таблицы [proshop\\_discounts](#)), название которой хранится в поле **caption** на языке **lang\_id**.
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), на котором хранится название скидки в поле **caption**.
- **caption** – название скидки **discount\_id** на языке **lang\_id**.

## proshop\_discounts\_types

Назначение:

- Хранение типов (глобальные, для продуктов, для пользователей) скидок, которые определены в таблице [proshop\\_discounts](#).

Структура:

| Поле        | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>   | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя типа скидки. Может принимать следующие значения: **global**, **products**, **users**.

## proshop\_emspost\_options

Назначение:

- Хранение настроек метода доставки EMS.

Структура:

| Поле         | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u>    | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>name</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>value</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – название настройки.
- **value** – значение настройки.

## proshop\_errors

Назначение:

- Хранение лога ошибок, возникающих в процессе работы CMS. Лог ошибок ведется только при отключенном режиме отладки (см. описание константы [\\_DEBUG\\_](#)).

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u>       | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>add_date</b> | datetime     |                 |          | Нет  | Нет          |                |
| <b>errno</b>    | int(11)      |                 |          | Нет  | Нет          |                |
| <b>errmsg</b>   | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>file</b>     | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>line</b>     | int(11)      |                 |          | Нет  | Нет          |                |
| <b>uri</b>      | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 109                  | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **add\_date** – дата и время возникновения ошибки по часам сервера.
- **errno** – номер ошибки.
- **errmsg** – текст сообщения об ошибке.
- **file** – имя (и путь) файла, в котором произошла ошибка.
- **line** – номер строки файла **file**, в которой произошла ошибка.
- **uri** – адрес страницы магазина, на которой находился пользователь, когда произошла ошибка.

## proshop\_fastmenu

Назначение:

- Хранение базовой информации об элементах быстрого меню, доступного на витрине.

Структура:

| Поле       | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------|---------|-----------------|----------|------|--------------|----------------|
| <u>id</u>  | int(11) |                 |          | Нет  | Нет          | auto_increment |
| is_visible | char(1) | utf8_general_ci |          | Нет  |              |                |
| pos        | int(11) |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **is\_visible** – флаг видимости элемента на витрине (**Y** – видим, **N** – временно невидим).
- **pos** – позиция текущего элемента относительно других элементов быстрого меню.

## proshop\_fastmenu\_texts

Назначение:

- Хранение текстов ссылок, определенных в таблице [proshop\\_fastmenu](#), на разных языках.

Структура:

| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| menu_id   | int(11)      |                 |          | Нет  | 0            |                |
| lang_id   | int(11)      |                 |          | Нет  | 0            |                |
| title     | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| url       | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| menu_id     | BTREE | Да         | Нет      | menu_id | 0                    | A         |      |
|             |       |            |          | lang_id | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **menu\_id** – идентификатор элемента быстрого меню из таблицы [proshop\\_fastmenu](#).
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), для которого заданы тексты **title** и **url**.
- **title** – текст названия ссылки элемента на языке lang\_id.
- **url** – URL ссылки, действующий на языке lang\_id.

## proshop\_homepage\_texts

Назначение:

- Хранение текстов главной страницы витрины магазина на разных языках.

Структура:

| Поле           | Тип      | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|----------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)  |                 |          | Нет  | Нет          | auto_increment |
| <b>lang_id</b> | int(11)  |                 |          | Нет  | 0            |                |
| <b>content</b> | longtext | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| lang_id     | BTREE | Да         | Нет      | lang_id | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), на котором задан текст **content**.
- **content** – текст главной страницы витрины магазина для языка lang\_id.

## proshop\_labels

Назначение:

- Хранение базовой информации о текстовых метках, предназначенных для поддержки многоязычности пользовательского интерфейса.

Структура:



| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| eng_name  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| zone      | varchar(1)   | utf8_general_ci |          | Нет  | A            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 1266                 | A         |      |
| eng_name    | BTREE | Нет        | Нет      | eng_name | 1266                 | A         |      |
| zone        | BTREE | Нет        | Нет      | zone     | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **eng\_name** – служебное имя текстовой метки.
- **zone** – зона, в которой может применяться метка: **A** – админка, **C** – витрина.

## proshop\_labels\_texts

Назначение:

- Хранение текстов меток, определенных в таблице [proshop\\_labels](#), на разных языках.

Структура:

| Поле      | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|---------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11) |                 |          | Нет  | Нет          | auto_increment |
| label_id  | int(11) |                 |          | Нет  | 0            |                |
| lang_id   | int(11) |                 |          | Нет  | 0            |                |
| caption   | text    | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 3499                 | A         |      |
| label_id    | BTREE | Да         | Нет      | label_id | 0                    | A         |      |
|             |       |            |          | lang_id  | 3499                 | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **label\_id** – идентификатор текстовой метки из таблицы [proshop\\_labels](#), текст которой задан в **caption** на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задан текст метки **caption**.
- **caption** – текст метки **label\_id** на языке **lang\_id**.

## proshop\_languages

Назначение:

- Хранение информации о языках, применяемых в магазине.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>title</b>      | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>short</b>      | varchar(2)   | utf8_general_ci |          | Нет  |              |                |
| <b>is_default</b> | char(1)      | utf8_general_ci |          | Нет  | N            |                |
| <b>curr_id</b>    | int(11)      |                 |          | Нет  | Нет          |                |
| <b>flag</b>       | varchar(20)  | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле  | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|-------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id    | 2                    | A         |      |
| <b>short</b>   | BTREE | Нет        | Нет      | short | 0                    | A         |      |
| <b>title</b>   | BTREE | Нет        | Нет      | title | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **title** – название языка.
- **short** – международное двубуквенное обозначение языка (две английские буквы нижнего регистра).
- **is\_default** – флаг, определяющий, является ли текущий язык языком магазина по умолчанию (Y – является, N – не является). В таблице может быть только одна запись со значением данного поля, равным Y.
- **curr\_id** – идентификатор валюты (из таблицы [proshop\\_currencies](#)), которая должна быть выбрана при выборе данного языка на витрине.
- **flag** – имя файла изображения флага языка, который располагается в директории, определенной в константе [FLAGS\\_PICS](#).

## proshop\_languages\_currencies

Назначение:

- Хранение связей валют с языками (наборов валют, разрешенных к выбору на витрине при выборе определенного языка пользователем).

Структура:

| Поле           | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>      | int(11) |           |          | Нет  | Нет          | auto_increment |
| <b>lang_id</b> | int(11) |           |          | Нет  | Нет          |                |
| <b>curr_id</b> | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| lang_id     | BTREE | Нет        | Нет      | lang_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#).
- **curr\_id** – идентификатор валюты из таблицы [proshop\\_currencies](#).

## proshop\_login\_history

Назначение:

- Хранение истории попыток авторизаций (входа в аккаунт) пользователей (в т.ч. администраторов) – удачных и неудачных.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию        | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|---------------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет                 | auto_increment |
| <b>email</b>      | varchar(254) | utf8_general_ci |          | Нет  | Нет                 |                |
| <b>user_id</b>    | int(11)      |                 |          | Нет  | 0                   |                |
| <b>ip</b>         | varchar(15)  | utf8_general_ci |          | Нет  |                     |                |
| <b>date</b>       | datetime     |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>is_success</b> | char(1)      | utf8_general_ci |          | Нет  | Y                   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 17                   | A         |      |
| user_id     | BTREE | Нет        | Нет      | user_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **email** – e-mail пользователя, который пытался авторизоваться.
- **user\_id** – идентификатор пользователя (из таблицы [proshop\\_users](#)), который пытался авторизоваться, или 0, если не удалось определить идентификатор.
- **ip** – IP-адрес пользователя.
- **date** – дата и время попытки авторизации по часам сервера.
- **is\_success** – флаг успешности авторизации (**Y** – успешна, **N** - неуспешна).

## proshop\_manufacturers

Назначение:

- Хранение базовой информации о производителях.

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>eng_name</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>img</b>      | varchar(254) | utf8_general_ci |          | Да   | <i>NULL</i>  |                |
| <b>VISIBLE</b>  | char(1)      | utf8_general_ci |          | Нет  |              |                |
| <b>pos</b>      | int(11)      |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **eng\_name** – имя в URL (для modRewrite) производителя. Может быть пустым.
- **img** – имя файла изображения производителя. Файлы изображений производителей хранятся в директории, путь к которой прописан в константе [MANUF\\_PICS](#).
- **VISIBLE** – флаг видимости производителя на витрине (**Y** – видим, **N** - невидим).
- **pos** – позиция производителя относительно других производителей в таблице – для сортировки списка.

## proshop\_manufacturers\_texts

Назначение:

- Хранение текстов, относящихся к производителям, определенным в таблице [proshop\\_manufacturers](#), на разных языках.

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>manuf_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>alt_text</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>caption</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>descr</b>    | text         | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса     | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-----------------|-------|------------|----------|----------|----------------------|-----------|------|
| <b>PRIMARY</b>  | BTREE | Да         | Нет      | id       | 4                    | A         |      |
| <b>manuf_id</b> | BTREE | Да         | Нет      | manuf_id | 0                    | A         |      |
|                 |       |            |          | lang_id  | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.

- **manuf\_id** – идентификатор производителя из таблицы [proshop\\_manufacturers](#), для которого заданы тексты **alt\_text**, **caption** и **descr**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором заданы тексты **alt\_text**, **caption** и **descr**.
- **alt\_text** – альтернативный текст изображения (атрибут **alt** тэга **img**) производителя на языке **lang\_id**.
- **caption** – название производителя на языке **lang\_id**.
- **descr** – описание производителя на языке **lang\_id**.

## proshop\_meta\_tags

Назначение:

- Хранение базовой информации о мета-тэгах различных страниц витрины магазина (связи мета-тэгов с различными страницами).

Структура:

| Поле           | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>      | int(11) |           |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>type_id</b> | int(11) |           |          | Нет  | 0            |                |
| <b>elem_id</b> | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 140                  | A         |      |
| <b>type_id</b> | BTREE | Нет        | Нет      | type_id | 7                    | A         |      |
| <b>elem_id</b> | BTREE | Нет        | Нет      | elem_id | 140                  | A         |      |

Поля:

- **id** – уникальный идентификатор записи (мета-тэга).
- **type\_id** – идентификатор типа мета-тэга (из таблицы [proshop\\_meta\\_types](#)), который определяет, к какой таблице относится идентификатор, хранящийся в поле **elem\_id**.
- **elem\_id** – идентификатор элемента из таблицы, которая определяется типом **type\_id** данного мета-тэга (см. описание таблицы [proshop\\_meta\\_types](#)).

## proshop\_meta\_texts

Назначение:

- Хранение текстов мета-тэгов, определенных в таблице [proshop\\_meta\\_tags](#), на разных языках.

Структура:

| Поле              | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|---------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11) |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>tag_id</b>     | int(11) |                 |          | Нет  | 0            |                |
| <b>lang_id</b>    | int(11) |                 |          | Нет  | 0            |                |
| <b>meta_title</b> | text    | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>meta_keys</b>  | text    | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>meta_desc</b>  | text    | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 191                  | A         |      |
| <b>tag_id</b>  | BTREE | Да         | Нет      | tag_id  | 0                    | A         |      |
|                |       |            |          | lang_id | 191                  | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **tag\_id** – идентификатор мета-тэга из таблицы [proshop\\_meta\\_tags](#), тексты которого заданы в данной записи.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором заданы тексты в данной записи.
- **meta\_title** – текст тэга `<title>` на языке **lang\_id**.
- **meta\_keys** – текст тэга `<meta http-equiv="keywords" />` на языке **lang\_id**.
- **meta\_desc** – текст тэга `<meta http-equiv="description" />` на языке **lang\_id**.

## proshop\_meta\_types

Назначение:

- Хранение типов мета-тэгов, применяемых в таблице [proshop\\_meta\\_tags](#) для определения, к какой таблице относится тот или иной мета-тэг.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>type_name</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 8                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **type\_name** – служебное имя типа мета-тэга. По данному имени можно определить, к какой таблице относится мета-тэг данного типа. См. также методы класса [MetaTags](#).

## proshop\_modules

Назначение:

- Хранение информации о модулях магазина.

Структура:

| Поле                    | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>               | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>admin_menu_id</b>    | int(11)      |                 |          | Нет  | Нет          |                |
| <b>pos</b>              | int(11)      |                 |          | Нет  | Нет          |                |
| <b>name</b>             | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>label</b>            | varchar(64)  | utf8_general_ci |          | Нет  |              |                |
| <b>avail</b>            | char(1)      | utf8_general_ci |          | Нет  |              |                |
| <b>changeable</b>       | char(1)      | utf8_general_ci |          | Нет  | N            |                |
| <b>admin_autorun</b>    | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |
| <b>customer_autorun</b> | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |
| <b>icon</b>             | varchar(254) | utf8_general_ci |          | Да   | NULL         |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 52                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **admin\_menu\_id** – идентификатор блока меню админки (из таблицы [proshop\\_admin\\_menus](#)), в котором необходимо отобразить ссылку на данный модуль. Если данное поле равно 0, ссылка на данный модуль не отображается ни в одном из блоков меню админки.
- **pos** – позиция ссылки данного модуля по отношению к ссылкам других модулей, расположенных в том же блоке меню админки. Значение данного поля игнорируется, если **admin\_menu\_id** равно 0.
- **name** – название модуля.
- **label** – текстовая метка (из таблицы [proshop\\_labels](#)) зоны А (для админки), в которой прописано название модуля на разных языках. Используется для отображения названия модуля в списке модулей в админке, а также в блоках меню админки.
- **avail** – флаг включенности данного модуля (**Y** – модуль включен, **N** – модуль выключен).
- **changeable** – флаг возможности включения/отключения данного модуля через админку (**Y** – модуль можно включить или выключить, **N** – модуль нельзя включить или выключить).
- **admin\_autorun** – флаг автозапуска PHP-файла модуля в админке при каждой загрузке любой страницы админки (при условии, что флаг **avail** имеет значение **Y**).
- **customer\_autorun** – флаг автозапуска PHP-файла модуля на витрине при каждой загрузке любой страницы витрины (при условии, что флаг **avail** имеет значение **Y**).

- **icon** – имя файла иконки модуля, отображаемой в меню админки. Файлы иконок модулей хранятся в директории **templates/admin/images/modules\_icons** и их серые варианты – в директории **templates/admin/images/modules\_gray\_icons**. Значение данного поля должно быть равно NULL или быть пустым, если значение поля **admin\_menu\_id** равно 0.

## proshop\_modules\_admin\_actions

Назначение:

- Хранение наборов действий (над модулями), которые могут быть заблокированы для определенных групп администраторов, хранящихся в таблице [proshop\\_admins\\_groups](#). Идентификаторы разрешенных наборов действий для определенных групп администраторов хранятся в таблице [proshop\\_admins\\_allowed\\_actions](#).

Структура:

| Поле          | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>     | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>module</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>acts</b>   | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>label</b>  | varchar(64)  | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле   | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|--------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id     | 54                   | A         |      |
| <b>module</b>  | BTREE | Нет        | Нет      | module | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **module** – название модуля, для которого задан набор действий в поле **acts**. Модуль с таким названием должен храниться в таблице [proshop\\_modules](#).
- **acts** – набор действий (действия указаны через точку с запятой) указанного модуля **module**.
- **label** – текстовая метка (из таблицы [proshop\\_labels](#)), определяющая название данного набора действий на разных языках.

## proshop\_msgs

Назначение:

- Хранение информации о личных сообщениях (переписка между администрацией и зарегистрированным покупателем).

Структура:



| Поле                 | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>            | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>topic_id</b>      | int(11)      |                 |          | Нет  | Нет          |                |
| <b>user_id</b>       | int(11)      |                 |          | Нет  | Нет          |                |
| <b>user_name</b>     | varchar(255) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>user_email</b>    | varchar(255) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>user_is_admin</b> | varchar(1)   | utf8_general_ci |          | Нет  | Нет          |                |
| <b>datetime</b>      | datetime     |                 |          | Нет  | Нет          |                |
| <b>text</b>          | text         | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **topic\_id** – идентификатор темы сообщения из таблицы [proshop\\_msgs\\_topics](#).
- **user\_id** – идентификатор пользователя, оставившего сообщение (из таблицы [proshop\\_users](#)). Если сообщение оставлено незарегистрированным пользователем, данное поле равно 0.
- **user\_name** – имя пользователя, оставившего сообщение.
- **user\_email** – e-mail пользователя, оставившего сообщение.
- **user\_is\_admin** – если данное поле равно Y, сообщение оставил администратор. Иначе – обычный пользователь.
- **datetime** – метка даты и времени написания сообщения.
- **text** – текст сообщения.

## proshop\_msgs\_topics

Назначение:

- Хранение информации о темах личных сообщений (переписка администрации и покупателей).

Структура:

| Поле                  | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>             | int(11)      |                 |          | Нет  | <i>Нет</i>   | AUTO_INCREMENT |
| <b>create_time</b>    | datetime     |                 |          | Нет  | <i>Нет</i>   |                |
| <b>user_upd_time</b>  | datetime     |                 |          | Нет  | <i>Нет</i>   |                |
| <b>admin_upd_time</b> | datetime     |                 |          | Да   | <i>NULL</i>  |                |
| <b>update_time</b>    | datetime     |                 |          | Нет  | <i>Нет</i>   |                |
| <b>status</b>         | varchar(1)   | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>user_read</b>      | varchar(1)   | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>user_id</b>        | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| <b>user_name</b>      | varchar(255) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>user_email</b>     | varchar(255) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>admin_id</b>       | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| <b>title</b>          | varchar(255) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **create\_time** – метка даты и времени создания темы.
- **user\_upd\_time** – метка даты и времени последнего обновления темы покупателем.
- **admin\_upd\_time** – метка даты и времени последнего обновления темы администратором.
- **update\_time** – метка даты и времени последнего обновления темы (кем угодно).
- **status** – статус темы: **N** – новая (не прочитанная), **O** – открытая для обсуждения, **C** - закрытая.
- **user\_read** – флаг того, что пользователь, инициировавший тему, прочитал последнее сообщение в ней: **Y** – пользователь прочел последнее сообщение, **N** – не прочел.
- **user\_id** – идентификатор пользователя (из таблицы [proshop\\_users](#)), инициировавшего тему. Может быть равен 0, если тема создана незарегистрированным пользователем.
- **user\_name** – имя пользователя, инициировавшего тему.
- **user\_email** – e-mail пользователя, инициировавшего тему.
- **admin\_id** – идентификатор последнего администратора (из таблицы [proshop\\_users](#)), написавшего сообщение в тему. Если равен 0, то администратор либо удален из базы, либо ни один администратор еще не написал в эту тему.
- **title** – название темы.

## proshop\_news

Назначение:

- Хранение базовой информации о новостях магазина.

Структура:

| Поле                    | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>               | int(11)     |                 |          | Нет  | <i>Нет</i>   | AUTO_INCREMENT |
| <b>date</b>             | date        |                 |          | Нет  | 0000-00-00   |                |
| <b>update_date</b>      | datetime    |                 |          | Нет  | <i>Нет</i>   |                |
| <b>img</b>              | varchar(60) | utf8_general_ci |          | Да   | NULL         |                |
| <b>sent</b>             | char(1)     | utf8_general_ci |          | Нет  | N            |                |
| <b>show_in_homepage</b> | varchar(1)  | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса        | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|--------------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY            | BTREE | Да         | Нет      | id          | 2                    | A         |      |
| <b>date</b>        | BTREE | Нет        | Нет      | date        | 0                    | A         |      |
| <b>update_date</b> | BTREE | Нет        | Нет      | update_date | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **date** – дата публикации новости (раньше этой даты новость не должна появляться на витрине).
- **update\_date** – дата и время последнего редактирования новости.
- **img** – имя файла превью-изображения новости, который располагается в директории, путь к которой прописан в константе [NEWS\\_PICS](#).
- **sent** – флаг признака того, что новость была разослана всем подписчикам магазина (Y – новость была разослана, N – новость еще не разослана).
- **show\_in\_homepage** – флаг отображения анонса новости на главной странице витрины магазина: Y - отображать, N – не отображать.

## proshop\_news\_edit\_history

Назначение:

- Хранение статистики, кто и когда редактировал новость.

Структура:

| Поле               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>          | int(11)      |                 |          | Нет  | <i>Нет</i>   | AUTO_INCREMENT |
| <b>fact_id</b>     | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| <b>update_date</b> | datetime     |                 |          | Нет  | <i>Нет</i>   |                |
| <b>user_id</b>     | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| <b>user_name</b>   | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>user_email</b>  | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id          | 10                   | A         |      |
| <b>fact_id</b> | BTREE | Нет        | Нет      | fact_id     | 0                    | A         |      |
|                |       |            |          | update_date | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **fact\_id** – идентификатор новости из таблицы [proshop\\_news](#).
- **update\_date** – дата и время редактирования новости.
- **user\_id** – идентификатор пользователя (из таблицы [proshop\\_users](#)), отредактировавшего новость.
- **user\_name** – имя пользователя, отредактировавшего новость.
- **user\_email** – e-mail пользователя, отредактировавшего новость.

## proshop\_news\_subscribers

Назначение:

- Хранение информации о подписчиках на новости магазина.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>email</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>confirmed</b> | char(1)      | utf8_general_ci |          | Нет  | N            |                |
| <b>key</b>       | varchar(60)  | utf8_general_ci |          | Да   | <i>NULL</i>  |                |
| <b>lang_id</b>   | int(11)      |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 0                    | A         |      |
| <b>email</b>   | BTREE | Да         | Нет      | email   | 0                    | A         |      |
| <b>lang_id</b> | BTREE | Нет        | Нет      | lang_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **email** – e-mail подписчика.
- **confirmed** – флаг того, что подписчик подтвердил получение новостей по почте (**Y** – подтвердил, **N** – еще не подтвердил). Не подтвержденным подписчикам новости не рассылаются.
- **key** – ключ подтверждения подписки (или отписки) на новости, отсылаемый в письме со ссылкой для подтверждения подписки (отписки).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором следует присылать новости подписчикам.

## proshop\_news\_texts

Назначение:

- Хранение текстов новостей, определенных в таблице [proshop\\_news](#), на разных языках.

Структура:

| Поле               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>          | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>fact_id</b>     | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>     | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>     | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>content</b>     | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>short_descr</b> | varchar(255) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 4                    | A         |      |
| fact_id     | BTREE | Да         | Нет      | fact_id | 0                    | A         |      |
|             |       |            |          | lang_id | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **fact\_id** – идентификатор новости из таблицы [proshop\\_news](#), тексты которой записаны в полях **caption**, **content** и **short\_descr** на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором записаны тексты **caption**, **content** и **short\_descr**.
- **caption** – заголовок новости на языке **lang\_id**.
- **content** – текст новости на языке **lang\_id**.
- **short\_descr** – текст анонса новости на языке **lang\_id**.

## proshop\_options

Назначение:

- Хранение опций различных модулей.

Структура:

| Поле         | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>    | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>value</b> | text         | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 18                   | A         |      |
| name        | BTREE | Да         | Нет      | name | 18                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – имя опции любого модуля.
- **value** – значение опции любого модуля.

## **proshop\_orders**

Назначение:

- Хранение заказов магазина.

Структура:

# Руководство разработчика Fast-Sales Pro

| Поле                    | Тип           | Сравнение       | Атрибуты | Null | По умолчанию        | Дополнительно  |
|-------------------------|---------------|-----------------|----------|------|---------------------|----------------|
| id                      | int(11)       |                 |          | Нет  | Нет                 | AUTO_INCREMENT |
| customer_id             | int(11)       |                 |          | Нет  | 0                   |                |
| customer_title          | varchar(254)  | utf8_general_ci |          | Нет  | Нет                 |                |
| customer_first_name     | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_last_name      | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_second_name    | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_gender         | char(1)       | utf8_general_ci |          | Нет  |                     |                |
| customer_company        | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_ogrn           | varchar(60)   | utf8_general_ci |          | Да   | NULL                |                |
| customer_inn            | varchar(60)   | utf8_general_ci |          | Да   | NULL                |                |
| customer_kpp            | varchar(60)   | utf8_general_ci |          | Да   | NULL                |                |
| customer_contact_person | varchar(100)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_legal_address  | varchar(200)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_bank_name      | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_bank_bik       | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_bank_rs        | varchar(100)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_bank_ks        | varchar(100)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_phone          | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_mobile_phone   | varchar(254)  | utf8_general_ci |          | Нет  | Нет                 |                |
| customer_fax            | varchar(254)  | utf8_general_ci |          | Нет  | Нет                 |                |
| customer_address        | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_ship_address   | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_zip_code       | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_ship_zip_code  | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_city           | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_ship_city      | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_country        | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_ship_country   | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_state          | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_ship_state     | varchar(254)  | utf8_general_ci |          | Да   | NULL                |                |
| customer_email          | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| customer_website        | varchar(254)  | utf8_general_ci |          | Нет  | Нет                 |                |
| customer_person_type    | varchar(1)    | utf8_general_ci |          | Нет  | N                   |                |
| date                    | datetime      |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| edit_date               | datetime      |                 |          | Да   | NULL                |                |
| sum                     | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| tax                     | varchar(32)   | utf8_general_ci |          | Нет  | 0                   |                |
| discount_per            | decimal(12,2) |                 |          | Нет  | Нет                 |                |
| weight                  | varchar(32)   | utf8_general_ci |          | Нет  |                     |                |
| shipping_cost           | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| status_id               | int(11)       |                 |          | Нет  | 0                   |                |
| emailed                 | char(1)       | utf8_general_ci |          | Нет  |                     |                |
| change_emailed          | varchar(1)    | utf8_general_ci |          | Нет  | N                   |                |
| comment                 | text          | utf8_general_ci |          | Да   | NULL                |                |
| payment_id              | int(11)       |                 |          | Нет  | 0                   |                |
| payment_method          | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| payment_details         | text          | utf8_general_ci |          | Нет  | Нет                 |                |
| shipping_method         | varchar(254)  | utf8_general_ci |          | Нет  |                     |                |
| lang_id                 | int(11)       |                 |          | Да   | NULL                |                |
| curr_id                 | int(11)       |                 |          | Да   | NULL                |                |
| visitor_id              | int(11)       |                 |          | Нет  | Нет                 |                |
| allow_download          | varchar(1)    | utf8_general_ci |          | Нет  | Y                   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id          | 50                   | A         |      |
| status_id   | BTREE | Нет        | Нет      | status_id   | 0                    | A         |      |
| customer_id | BTREE | Нет        | Нет      | customer_id | 0                    | A         |      |
|             |       |            |          | status_id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (номер заказа).
- **customer\_id** – идентификатор зарегистрированного покупателя (из таблицы [proshop\\_users](#)), который совершил заказ. Может принимать значение 0 в том случае, если заказ осуществлен без регистрации.
- **customer\_title** – обращение к покупателю. См. поле **title** таблицы [proshop\\_users](#).
- **customer\_first\_name** – имя покупателя (без фамилии). См. поле **first\_name** таблицы [proshop\\_users](#).
- **customer\_last\_name** – фамилия покупателя. См. поле **last\_name** таблицы [proshop\\_users](#).
- **customer\_second\_name** – отчество покупателя. См. поле **second\_name** таблицы [proshop\\_users](#).
- **customer\_gender** – пол покупателя. См. поле **gender** таблицы [proshop\\_users](#).
- **customer\_company** – название организации покупателя. См. поле **company\_name** таблицы [proshop\\_users](#).
- **customer\_ogrn** – ОГРН организации покупателя. См. поле **ogrn** таблицы [proshop\\_users](#).
- **customer\_inn** – ИНН организации покупателя. См. поле **inn** таблицы [proshop\\_users](#).
- **customer\_kpp** – КПП организации покупателя. См. поле **kpp** таблицы [proshop\\_users](#).
- **customer\_contact\_person** – ФИО контактного лица организации. См. поле **contact\_person** таблицы [proshop\\_users](#).
- **customer\_legal\_address** – юридический адрес организации. См. поле **legal\_address** таблицы [proshop\\_users](#).
- **customer\_bank\_name** – наименование банка организации покупателя. См. поле **bank\_name** таблицы [proshop\\_users](#).
- **customer\_bank\_bik** – БИК банка организации покупателя. См. поле **bank\_bik** таблицы [proshop\\_users](#).
- **customer\_bank\_rs** – р/с организации покупателя. См. поле **bank\_rs** таблицы [proshop\\_users](#).
- **customer\_bank\_ks** – к/с банка организации покупателя. См. поле **bank\_ks** таблицы [proshop\\_users](#).
- **customer\_phone** – номера телефонов покупателя. См. поле **phone** таблицы [proshop\\_users](#).
- **customer\_mobile\_phone** – номер мобильного телефона покупателя. См. поле **mobile\_phone** таблицы [proshop\\_users](#).
- **customer\_fax** – номер факса покупателя. См. поле **fax** таблицы [proshop\\_users](#).



- **customer\_address** – адрес (улица, номер дома, квартиры) покупателя. См. поле **address** таблицы [proshop\\_users](#).
- **customer\_ship\_address** – отдельный адрес для доставки заказа покупателю. См. поле **ship\_address** таблицы [proshop\\_users](#).
- **customer\_zip\_code** – почтовый индекс адреса покупателя. См. поле **zip\_code** таблицы [proshop\\_users](#).
- **customer\_ship\_zip\_code** – почтовый индекс отдельного адреса для доставки заказа покупателю. См. поле **ship\_zip\_code** таблицы [proshop\\_users](#).
- **customer\_city** – город покупателя. См. поле **city** таблицы [proshop\\_users](#).
- **customer\_ship\_city** – город отдельного адреса доставки заказа покупателю. См. поле **ship\_city** таблицы [proshop\\_users](#).
- **customer\_country** – название страны покупателя.
- **customer\_ship\_country** – название страны отдельного адреса доставки заказа покупателю.
- **customer\_state** – название области страны покупателя.
- **customer\_ship\_state** – название области страны отдельного адреса доставки заказа покупателю.
- **customer\_email** – e-mail покупателя. См. поле **email** таблицы [proshop\\_users](#).
- **customer\_website** – адрес сайта покупателя. См. поле **website** таблицы [proshop\\_users](#).
- **customer\_person\_type** – тип покупателя: **A** – юридическое лицо, **N** – физическое лицо.
- **date** – дата и время совершения заказа.
- **edit\_date** – дата и время редактирования состава заказа. Если состав заказа ни разу не редактировался, данное поле содержит NULL.
- **sum** – сумма заказа с учетом всех скидок и налогов в базовой валюте магазина. Не включает стоимость доставки.
- **tax** – сумма налога в базовой валюте, который входит в **sum**.
- **discount\_per** – величина процента скидки на заказ, которая учтена в **sum**.
- **weight** – суммарный вес заказа в кг.
- **shipping\_cost** – стоимость доставки в базовой валюте магазина.
- **status\_id** – идентификатор статуса заказа из таблицы [proshop\\_orders\\_statuses](#).
- **emailed** – флаг того, что оповещение об успешно принятом заказе было отослано покупателю (**Y** – покупатель был оповещен, **N** – покупатель не был оповещен).
- **change\_emailed** – флаг того, что оповещение об изменении состава заказа было отослано покупателю (**Y** – покупатель был оповещен, **N** – покупатель не был оповещен).
- **comment** – текст комментария покупателя к заказу.
- **payment\_id** – идентификатор способа оплаты заказа из таблицы [proshop\\_payments](#).
- **payment\_method** – название способа оплаты заказа.
- **payment\_details** – реквизиты для совершения оплаты в ручном режиме методом, который указан в **payment\_method**.
- **shipping\_method** – название способа доставки заказа.

- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), который был выбран в магазине в момент оформления заказа.
- **curr\_id** – идентификатор валюты (из таблицы [proshop\\_currencies](#)), которая была выбрана в магазине в момент оформления заказа.
- **visitor\_id** – идентификатор посетителя (из таблицы [proshop\\_visitors](#)), который совершил заказ.
- **allow\_download** – флаг, определяющий, будет ли покупатель скачивать заказанные им цифровые товары после оплаты заказа, или ему нужно их доставить наземным способом: **Y** – будет скачивать, **N** – доставить наземным способом.

## proshop\_orders\_products

Назначение:

- Хранение информации о заказанных продуктах (по каждому заказу в отдельности).

Структура:

| Поле                   | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>              | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| order_id               | int(11)      |                 |          | Нет  | 0            |                |
| product_id             | int(11)      |                 |          | Нет  | 0            |                |
| variant_id             | int(11)      |                 |          | Нет  | 0            |                |
| product_cat            | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| cat_id                 | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| product_caption        | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| product_sku            | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| product_options        | text         | utf8_general_ci |          | Да   | <i>NULL</i>  |                |
| product_qty            | int(11)      |                 |          | Нет  | 0            |                |
| product_price          | varchar(32)  | utf8_general_ci |          | Нет  |              |                |
| product_weight         | varchar(32)  | utf8_general_ci |          | Нет  |              |                |
| product_egood          | char(1)      | utf8_general_ci |          | Нет  |              |                |
| product_egood_pin_code | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| cancelled              | char(1)      | utf8_general_ci |          | Нет  | N            |                |
| pending                | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |
| supplier_notified      | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |
| if_null_qty            | varchar(50)  | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| tax_percent            | int(11)      |                 |          | Нет  | 0            |                |
| tax_value              | varchar(32)  | utf8_general_ci |          | Нет  | 0            |                |
| no_vat                 | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null | Комментарий |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|-------------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 89                   | A         |      |             |
| order_id    | BTREE | Нет        | Нет      | order_id | 0                    | A         |      |             |

Поля:

- **id** – уникальный идентификатор записи.
- **order\_id** – идентификатор (номер) заказа из таблицы [proshop\\_orders](#), которому принадлежит данный заказанный продукт.
- **product\_id** – идентификатор продукта из таблицы [proshop\\_products](#). Данное поле может иметь значение 0, если заказанного продукта уже нет в таблице [proshop\\_products](#).
- **variant\_id** – идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#). Данное поле может иметь значение 0, если данного варианта заказанного продукта уже нет в таблице [proshop\\_products\\_variants](#).
- **product\_cat** – название категории заказанного продукта.
- **cat\_id** – идентификатор категории заказанного продукта из таблицы [proshop\\_categories](#). Данное поле может иметь значение 0, если данной категории уже нет в таблице [proshop\\_categories](#).
- **product\_caption** – название заказанного продукта.
- **product\_sku** – артикул заказанного продукта.
- **product\_options** – опции заказанного продукта, если продукт имеет опции.
- **product\_qty** – кол-во заказанных экземпляров продукта.
- **product\_price** – цена одного экземпляра продукта в базовой валюте, включающая все налоги и скидки.
- **product\_weight** – вес одного экземпляра продукта (или цифрового носителя, если продукт является цифровым) в кг.
- **product\_egood** – флаг того, что продукт является цифровым (**Y** – продукт цифровой, **N** – продукт обычный).
- **product\_egood\_pin\_code** – секретный код цифрового продукта, если продукт является цифровым (значение поля **product\_egood** равно **Y**) и если он предназначен для продажи пин-кодов (секретных кодов).
- **cancelled** – флаг того, что заказ данного продукта отменен (**Y** – заказ отменен, **N** – заказ не отменен).
- **pending** – флаг того, что заказ данного продукта отложен до поступления продукта на склад (**Y** – заказ отложен, **N** – заказ не отложен). Значение **Y** данного параметра означает, что продукт на момент его заказа отсутствовал на складе.
- **supplier\_notified** – флаг того, что данный продукт уже экспортировался во внешний файл списка заказанных с отложенной отгрузкой товаров для последующей передачи данного списка поставщику: **Y** – товар экспортировался в список, **N** – не экспортировался.
- **if\_null\_qty** – строка, содержащая условия доставки товара в случае его отсутствия на складе.
- **tax\_percent** – величина процента налога (от 0 до 100), который входит в **product\_price**.
- **tax\_value** – величина налога в базовой валюте, который входит в **product\_price**.

- **no\_vat** – флаг того, что налог, который входит в **product\_price**, является нулевым и является особой формой нулевого налога - «Без НДС» (**Y** – налог является нулевым налогом особой формы «Без НДС», **N** – налог является обычным нулевым или не нулевым налогом).

## proshop\_orders\_statuses

Назначение:

- Хранение возможных статусов заказов.

Структура:

| Поле                 | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>            | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>pos</b>           | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>          | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>may_cancel</b>    | char(1)      | utf8_general_ci |          | Нет  |              |                |
| <b>may_admin_set</b> | char(1)      | utf8_general_ci |          | Нет  |              |                |
| <b>may_changed</b>   | varchar(1)   | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 8                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (статуса).
- **pos** – позиция данного статуса относительно других статусов в этой таблице. Применяется для сортировки отображения статусов.
- **name** – служебное имя статуса.
- **may\_cancel** – флаг, определяющий возможность отмены заказа (покупателем), имеющего данный статус: **Y** – покупатель сможет отменить заказ с данным статусом, **N** – не сможет.
- **may\_admin\_set** – флаг, определяющий возможность установки данного статуса у заказа в админке администратором: **Y** – администратор может установить данный статус какому-нибудь заказу, **N** – не может.
- **may\_changed** – флаг, определяющий, можно ли менять состав заказа (в админке), имеющего данный статус: **Y** – администратор может менять состав заказа с данным статусом, **N** – не может.

## proshop\_orders\_statuses\_texts

Назначение:

- Хранение названий статусов заказов для разных языков.

Структура:

| Поле      | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|-------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)     |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| status_id | int(11)     |                 |          | Нет  | Нет          |                |
| lang_id   | int(11)     |                 |          | Нет  | Нет          |                |
| caption   | varchar(60) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id        | 16                   | A         |      |
| status_id   | BTREE | Да         | Нет      | status_id | 8                    | A         |      |
|             |       |            |          | lang_id   | 16                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **status\_id** – идентификатор статуса заказа из таблицы [proshop\\_orders\\_statuses](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#).
- **caption** – название статуса **status\_id** на языке **lang\_id**.

## proshop\_orders\_statuses\_were

Назначение:

- Хранение идентификаторов статусов заказов, которые уже выставлялись ранее заказам. Данная таблица необходима для того, чтобы можно было определить, выставлялся ли определенный статус определенному заказу ранее.

Структура:

| Поле      | Тип      | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|----------|-----------|----------|------|--------------|----------------|
| <u>id</u> | int(11)  |           |          | Нет  | Нет          | AUTO_INCREMENT |
| order_id  | int(11)  |           |          | Нет  | 0            |                |
| status_id | int(11)  |           |          | Нет  | 0            |                |
| timestamp | datetime |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id        | 65                   | A         |      |
| order_id    | BTREE | Нет        | Нет      | order_id  | 0                    | A         |      |
|             |       |            |          | status_id | 0                    | A         |      |
| timestamp   | BTREE | Нет        | Нет      | timestamp | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **order\_id** – идентификатор (номер) заказа из таблицы [proshop\\_orders](#).
- **status\_id** – идентификатор статуса заказа из таблицы [proshop\\_orders\\_statuses](#).
- **timestamp** – дата и время установки статуса заказа.

## proshop\_payments

Назначение:

- Хранение базовой информации о способах оплаты заказов в магазине.

Структура:

| Поле                    | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>               | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>name</b>             | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>is_active</b>        | char(1)      | utf8_general_ci |          | Нет  |              |                |
| <b>is_auto</b>          | varchar(1)   | utf8_general_ci |          | Нет  | Y            |                |
| <b>is_auto_enabled</b>  | varchar(1)   | utf8_general_ci |          | Нет  | Y            |                |
| <b>lstep_email_only</b> | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |
| <b>pos</b>              | int(11)      |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 13                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя метода оплаты. Для оплаты, принимаемой вручную (не поддерживаемой автоматический режим), данное поле должно иметь значение **manual**.
- **is\_active** – флаг, определяющий активность данного метода оплаты: **Y** – активен, **N** - временно выключен.
- **is\_auto** – флаг, определяющий, включен ли автоматический режим приема платежа для данного метода оплаты: **Y** – автоматический режим включен, **N** – выключен (применяется ручной режим). Данный флаг должен быть установлен в значение **N**, если значение флага **is\_auto\_enabled** установлено в **N**.
- **is\_auto\_enabled** – флаг, определяющий, является ли данный метод оплаты автоматической платежной системой (такой, как **WebMoney**): **Y** – является платежной системой, **N** – не является платежной системой. Значение данного флага устанавливается в **Y**, если для данного метода платежа создан PHP-файл в директории [payments](#). Настройки автоматических платежных систем хранятся в таблице [proshop\\_payments\\_settings](#).
- **lstep\_email\_only** – флаг, определяющий, посылать ли текст с реквизитами для оплаты покупателю только по e-mail вместо вывода реквизитов на последней странице заказа (**Y** – посылать по e-mail и не показывать на последней странице заказа, **N** – посылать по e-mail и показывать на последней странице заказа). Тексты реквизитов хранятся в таблице [proshop\\_payments\\_texts](#).
- **pos** – позиция сортировки.

## proshop\_payments\_settings

Назначение:

- Хранение настроек автоматических платежных систем, определенных в таблице [proshop\\_payments](#), для их правильного функционирования и приема оплаты за заказы покупателей.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>payment_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>       | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>name_label</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>desc_label</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>value</b>      | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>maxlength</b>  | smallint(3)  |                 | UNSIGNED | Нет  | 254          |                |

Индексы:

| Имя индекса       | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------------|-------|------------|----------|------------|----------------------|-----------|------|
| <b>PRIMARY</b>    | BTREE | Да         | Нет      | id         | 40                   | A         |      |
| <b>payment_id</b> | BTREE | Нет        | Нет      | payment_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **payment\_id** – идентификатор платежной системы из таблицы [proshop\\_payments](#).
- **name** – служебное имя настройки.
- **name\_label** – текстовая метка (из таблицы [proshop\\_labels](#)) для отображения названия настройки на разных языках в админке.
- **desc\_label** – текстовая метка (из таблицы [proshop\\_labels](#)) для отображения описания настройки на разных языках в админке.
- **value** – значение настройки.
- **maxlength** – максимально возможная длина значения **value** настройки.

## proshop\_payments\_texts

Назначение:

- Хранение текстов (таких, как название, описание, реквизиты) методов оплаты, определенных в таблице [proshop\\_payments](#), на разных языках.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>payment_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>    | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>    | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>descr</b>      | text         | utf8_general_ci |          | Да   | <i>NULL</i>  |                |
| <b>laststep</b>   | text         | utf8_general_ci |          | Да   | <i>NULL</i>  |                |

Индексы:

| Имя индекса       | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------------|-------|------------|----------|------------|----------------------|-----------|------|
| <b>PRIMARY</b>    | BTREE | Да         | Нет      | id         | 21                   | A         |      |
| <b>payment_id</b> | BTREE | Да         | Нет      | payment_id | 0                    | A         |      |
|                   |       |            |          | lang_id    | 21                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **payment\_id** – идентификатор способа оплаты из таблицы [proshop\\_payments](#), для которого в данной записи хранятся тексты на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором хранятся тексты в данной записи.
- **caption** – название метода оплаты **payment\_id** на языке **lang\_id**.
- **descr** – описание метода оплаты **payment\_id** на языке **lang\_id**.
- **laststep** – текст (реквизиты) на последней странице заказа для метода оплаты **payment\_id** на языке **lang\_id**.

## proshop\_payments\_transactions

Назначение:

- Хранение информации о платежных транзакциях для автоматических платежных систем. Может применяться для отлова ошибок непоступления оплаты за заказ, когда на самом деле покупатель оплату совершил.

Структура:

| Поле                | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>           | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>payment_name</b> | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>date</b>         | datetime     |                 |          | Нет  | <i>Нет</i>   |                |
| <b>data</b>         | text         | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:



- **id** – уникальный идентификатор записи.
- **payment\_name** – служебное название платежной системы из таблицы [proshop\\_payments](#).
- **date** – дата и время совершения платежной транзакции по часам сервера.
- **date** – текст с информацией о транзакции (обычно здесь следует хранить содержимое массива **\$\_REQUEST**, полученное при прохождении транзакции).

## proshop\_products

Назначение:

- Хранение базовой информации о продуктах магазина.

Структура:

| Поле                 | Тип           | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------------|---------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>            | int(11)       |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>cat</b>           | int(11)       |                 |          | Нет  | 0            |                |
| <b>sku</b>           | varchar(100)  | utf8_general_ci |          | Нет  |              |                |
| <b>eng_name</b>      | varchar(254)  | utf8_general_ci |          | Нет  |              |                |
| <b>img</b>           | varchar(100)  | utf8_general_ci |          | Нет  |              |                |
| <b>thumb</b>         | varchar(100)  | utf8_general_ci |          | Нет  |              |                |
| <b>price</b>         | decimal(12,2) |                 |          | Нет  | 0.00         |                |
| <b>prev_price</b>    | decimal(12,2) |                 |          | Нет  | 0.00         |                |
| <b>qty</b>           | varchar(32)   | utf8_general_ci |          | Нет  |              |                |
| <b>min_order_qty</b> | smallint(5)   |                 | UNSIGNED | Нет  | 1            |                |
| <b>weight</b>        | varchar(32)   | utf8_general_ci |          | Нет  |              |                |
| <b>is_visible</b>    | char(1)       | utf8_general_ci |          | Нет  | Y            |                |
| <b>tax_id</b>        | int(11)       |                 |          | Нет  | 0            |                |
| <b>discount_id</b>   | int(11)       |                 |          | Нет  | 0            |                |
| <b>manuf_id</b>      | int(11)       |                 |          | Нет  | 0            |                |
| <b>sold_qty</b>      | int(11)       |                 |          | Нет  | 0            |                |
| <b>rating</b>        | float         |                 |          | Нет  | 0            |                |
| <b>egood</b>         | char(1)       | utf8_general_ci |          | Нет  | N            |                |
| <b>egood_pins</b>    | varchar(1)    | utf8_general_ci |          | Нет  | N            |                |
| <b>is_free</b>       | varchar(1)    | utf8_general_ci |          | Нет  | N            |                |
| <b>add_date</b>      | date          |                 |          | Нет  | 0000-00-00   |                |
| <b>edit_date</b>     | date          |                 |          | Нет  | Нет          |                |
| <b>caption_1</b>     | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |
| <b>short_descr_1</b> | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |
| <b>alt_1</b>         | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |
| <b>if_null_qty_1</b> | varchar(70)   | utf8_general_ci |          | Да   | NULL         |                |

Индексы:

| Имя индекса | Тип      | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|----------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE    | Да         | Нет      | id        | 111                  | A         |      |
| eng_name    | BTREE    | Нет        | Нет      | eng_name  | 111                  | A         |      |
| sold_qty    | BTREE    | Нет        | Нет      | sold_qty  | 5                    | A         |      |
| cat         | BTREE    | Нет        | Нет      | cat       | 27                   | A         |      |
| rating      | BTREE    | Нет        | Нет      | rating    | 5                    | A         |      |
| price       | BTREE    | Нет        | Нет      | price     | 111                  | A         |      |
| manuf_id    | BTREE    | Нет        | Нет      | manuf_id  | 3                    | A         |      |
| weight      | BTREE    | Нет        | Нет      | weight    | 1                    | A         |      |
| caption_1   | BTREE    | Нет        | Нет      | caption_1 | 111                  | A         | YES  |
| caption_4   | BTREE    | Нет        | Нет      | caption_4 | 111                  | A         | YES  |
| sku         | FULLTEXT | Нет        | Нет      | sku       | 1                    |           |      |
|             |          |            |          | eng_name  | 1                    |           |      |

Поля:

- **id** – уникальный идентификатор записи (продукта).
- **cat** – идентификатор главной категории продукта (из таблицы [proshop\\_categories](#)).
- **sku** – артикул продукта. Может быть пустым.
- **eng\_name** – имя в URL (для modRewrite) продукта. Может быть пустым.
- **img** – имя файла основного изображения продукта. Файлы изображений продуктов хранятся в директории, путь к которой прописан в константе [NOWM\\_PRODS\\_PICS](#) и [WM\\_PRODS\\_PICS](#) (если включено наложение водяного знака). Данное поле может быть пустым.
- **thumb** – имя файла уменьшенного варианта основного изображения продукта. Файлы изображений продуктов хранятся в директории, путь к которой прописан в константе [NOWM\\_PRODS\\_PICS](#) и [WM\\_PRODS\\_PICS](#) (если включено наложение водяного знака). Данное поле может быть пустым.
- **price** – цена одного экземпляра продукта в базовой валюте магазина.
- **prev\_price** – старая (зачеркнутая) цена одного экземпляра продукта в базовой валюте. Может быть равна 0.
- **qty** – кол-во доступных экземпляров продукта на складе. Если данный параметр равен пустой строке, считается, что кол-во бесконечно. Если данный параметр равен 0, считается, что продукта нет на складе.
- **min\_order\_qty** – минимальное кол-во данного продукта, которое может заказать покупатель.
- **weight** – вес одного экземпляра продукта в кг.
- **is\_visible** – флаг, определяющий, видим ли продукт на витрине: **Y** – продукт видим, **N** – продукт временно невидим.
- **tax\_id** – идентификатор налога (из таблицы [proshop\\_taxes](#)), примененного к цене данного продукта. Может быть равен 0, если налог к продукту не применен.
- **discount\_id** – идентификатор продуктовой скидки (из таблицы [proshop\\_discounts](#)), примененной к цене данного продукта. Может быть равен 0, если продуктовая скидка к продукту не применена.

- **manuf\_id** – идентификатор производителя данного продукта из таблицы [proshop\\_manufacturers](#). Может быть равен 0.
- **sold\_qty** – кол-во проданных экземпляров данного продукта.
- **rating** – рейтинг данного продукта, полученный путем создания отзывов покупателей о данном продукте (подсчет средней оценки, складывающейся из всех оценок в таблице [proshop\\_products\\_discuss](#)).
- **egood** – флаг того, что продукт является цифровым (электронным): **Y** – продукт является цифровым, **N** – продукт является обычным.
- **egood\_pins** – флаг того, что цифровой продукт содержит набор пин-кодов (предназначен для продажи секретных кодов вместо файлов): **Y** – цифровой продукт содержит набор секретных кодов, **N** – цифровой продукт не содержит набора секретных кодов и предназначен для продажи файлов. Значение данного поля не имеет смысла, если поле **egood** имеет значение **N**.
- **is\_free** – флаг того, что цифровой продукт может скачиваться посетителями бесплатно: **Y** – цифровой продукт бесплатный, **N** – цифровой продукт не бесплатный. Значение данного поля игнорируется, если значение поля **egood** равно **N**.
- **add\_date** – дата добавления записи о продукте в таблицу.
- **edit\_date** – дата последнего изменения записи о продукте в таблице.
- **caption\_N** – название продукта на языке, идентификатор которого указан вместо N.
- **short\_descr\_N** – краткое описание продукта на языке, идентификатор которого указан вместо N.
- **alt\_N** – альтернативный текст главного изображения продукта на языке, идентификатор которого указан вместо N.
- **if\_null\_qty\_N** – текст, выводимый на витрине в том случае, если товара нет на складе. N – идентификатор языка (из таблицы [proshop\\_languages](#)), на котором задан текст.

## proshop\_products\_acquired

Назначение:

- Хранение списков купленных продуктов для зарегистрированных покупателей (для каждого покупателя хранится свой список продуктов).

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>user_id</b> | int(11)      |                 |          | Нет  | Нет          |                |
| <b>prod_id</b> | int(11)      |                 |          | Нет  | Нет          |                |
| <b>options</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 3                    | A         |      |
| user_id     | BTREE | Да         | Нет      | user_id | 1                    | A         |      |
|             |       |            |          | prod_id | 3                    | A         |      |
| options     | BTREE | Нет        | Нет      | options | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **user\_id** – идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#).
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **options** - комбинация вариантов опций продукта **prod\_id**, являющаяся строкой, состоящей из идентификаторов вариантов опций (из таблицы [proshop\\_products\\_options\\_vars](#)), «склеянных» дефисом. Например: 12-24-25.

## proshop\_products\_categories

Назначение:

- Хранение связей продуктов с их дополнительными категориями.

Структура:

| Поле      | Тип        | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)    |                 |          | Нет  | Нет          | auto_increment |
| prod_id   | int(11)    |                 |          | Нет  | 0            |                |
| cat_id    | int(11)    |                 |          | Нет  | 0            |                |
| is_main   | varchar(1) | utf8_general_ci |          | Нет  | N            |                |
| pos       | int(11)    |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 115                  | A         |      |
| cat_id      | BTREE | Нет        | Нет      | cat_id  | 28                   | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id | 115                  | A         |      |
|             |       |            |          | cat_id  | 115                  | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **cat\_id** – идентификатор дополнительной категории (из таблицы [proshop\\_categories](#)) продукта **prod\_id**.
- **is\_main** – флаг, определяющий, является ли категория **cat\_id** главной для продукта **prod\_id**: Y – является, N – не является.
- **pos** – позиция продукта **prod\_id** относительно других продуктов в категории **cat\_id**.

## proshop\_products\_details

Назначение:

- Хранение связей продуктов с параметрами категорий (из таблицы [proshop\\_categories\\_params](#)) для формирования параметров (деталей) продуктов.

Структура:

| Поле            | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>       | int(11) |           |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>prod_id</b>  | int(11) |           |          | Нет  | 0            |                |
| <b>param_id</b> | int(11) |           |          | Нет  | 0            |                |
| <b>pos</b>      | int(11) |           |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса     | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-----------------|-------|------------|----------|----------|----------------------|-----------|------|
| <b>PRIMARY</b>  | BTREE | Да         | Нет      | id       | 10                   | A         |      |
| <b>prod_id</b>  | BTREE | Нет        | Нет      | prod_id  | 10                   | A         |      |
| <b>param_id</b> | BTREE | Нет        | Нет      | param_id | 5                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **param\_id** – идентификатор параметра категории продукта (в т.ч. родительской категории категории продукта) из таблицы [proshop\\_categories\\_params](#).
- **pos** – позиция данного параметра продукта относительно других параметров этого же продукта для сортировки деталей продукта.

## proshop\_products\_details\_texts

Назначение:

- Хранение значений деталей продуктов, которые определены в таблице [proshop\\_products\\_details](#). Хранятся как текстовые детали (значения на разных языках), так и числовые.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>detail_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>   | int(11)      |                 |          | Да   | <i>NULL</i>  |                |
| <b>value</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id        | 18                   | A         |      |
| detail_id   | BTREE | Да         | Нет      | detail_id | 9                    | A         |      |
|             |       |            |          | lang_id   | 18                   | A         | YES  |

Поля:

- **id** – уникальный идентификатор записи.
- **detail\_id** – идентификатор параметра (детали) продукта из таблицы [proshop\\_products\\_details](#).
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), на котором задано значение **value** текстового параметра (детали) **detail\_id** продукта. Если параметр продукта является числовым, поле **lang\_id** должно быть равно NULL, т.к. числа не зависят от языка.
- **value** – значение параметра **detail\_id** продукта.

## proshop\_products\_discuss

Назначение:

- Хранение оценок продуктов и комментариев к ним, оставленных посетителями/покупателями.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию        | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|---------------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет                 | auto_increment |
| <b>prod_id</b>    | int(11)      |                 |          | Нет  | 0                   |                |
| <b>visitor_id</b> | int(11)      |                 |          | Нет  | 0                   |                |
| <b>user_id</b>    | int(11)      |                 |          | Нет  | 0                   |                |
| <b>lang_id</b>    | int(11)      |                 |          | Нет  | Нет                 |                |
| <b>author</b>     | varchar(254) | utf8_general_ci |          | Нет  |                     |                |
| <b>email</b>      | varchar(254) | utf8_general_ci |          | Нет  | Нет                 |                |
| <b>rate</b>       | int(11)      |                 |          | Нет  | Нет                 |                |
| <b>comment</b>    | text         | utf8_general_ci |          | Нет  | Нет                 |                |
| <b>approved</b>   | char(1)      | utf8_general_ci |          | Нет  | N                   |                |
| <b>date</b>       | datetime     |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>ip</b>         | varchar(32)  | utf8_general_ci |          | Нет  |                     |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id         | 6                    | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id    | 6                    | A         |      |
| visitor_id  | BTREE | Нет        | Нет      | visitor_id | 3                    | A         |      |
| user_id     | BTREE | Нет        | Нет      | user_id    | 1                    | A         |      |
| lang_id     | BTREE | Нет        | Нет      | lang_id    | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#), которому выставлена оценка **rate** посетителем **visitor\_id**.
- **visitor\_id** – идентификатор посетителя из таблицы [proshop\\_visitors](#), который поставил оценку продукту **prod\_id**.
- **user\_id** – идентификатор пользователя из таблицы [proshop\\_users](#), который поставил оценку продукту **prod\_id**, если пользователь является зарегистрированным. Иначе – поле должно содержать 0.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), который был выбран в момент проставления оценки продукту.
- **author** – имя автора оценки **rate**.
- **email** – e-mail автора оценки **rate**.
- **rate** – оценка продукта по пятибалльной шкале (от 1 до 5).
- **comment** – комментарий к оценке продукта, написанный на языке **lang\_id**. Может быть пустым.
- **approved** – флаг того, что данная запись с комментарием одобрена модератором (администратором) сайта: **Y** – одобрена, **N** – еще не одобрена.
- **date** – дата и время момента проставления оценки продукту.
- **ip** – IP-адрес посетителя в момент проставления оценки продукту.

## proshop\_products\_featured

Назначение:

- Хранение идентификаторов продуктов, являющихся специальными предложениями.

Структура:

| Поле           | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>      | int(11) |           |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>prod_id</b> | int(11) |           |          | Нет  | 0            |                |
| <b>pos</b>     | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| <b>prod_id</b> | BTREE | Да         | Нет      | prod_id | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **pos** – позиция данного специального предложения относительно других продуктов-спецпредложений.

## proshop\_products\_files

Назначение:

- Хранение имен/путей файлов цифровых продуктов.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>prod_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>    | varchar(254) | utf8_general_ci |          | Да   | NULL         |                |
| <b>path</b>    | text         | utf8_general_ci |          | Да   | NULL         |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 1                    | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор цифрового продукта из таблицы [proshop\\_products](#).
- **name** – имя файла цифрового продукта **prod\_id**, хранящегося в директории, путь к которой прописан в константе [EGOODS\\_PATH](#). Данное поле должно быть равно NULL, если задано поле **path**.
- **path** – полный путь к файлу цифрового продукта **prod\_id** на сервере. Данное поле должно быть равно NULL, если задано поле **name**.

## proshop\_products\_files\_keys

Назначение:

- Хранение информации о секретных ключах для скачивания цифровых продуктов.

Структура:

| Поле                 | Тип          | Сравнение       | Атрибуты | Null | По умолчанию        | Дополнительно  |
|----------------------|--------------|-----------------|----------|------|---------------------|----------------|
| <b>id</b>            | int(11)      |                 |          | Нет  | Нет                 | auto_increment |
| <b>file_id</b>       | int(11)      |                 |          | Нет  | 0                   |                |
| <b>order_id</b>      | int(11)      |                 |          | Нет  | Нет                 |                |
| <b>order_prod_id</b> | int(11)      |                 |          | Нет  | 0                   |                |
| <b>visitor_id</b>    | int(11)      |                 |          | Нет  | Нет                 |                |
| <b>email</b>         | varchar(254) | utf8_general_ci |          | Нет  | Нет                 |                |
| <b>key</b>           | varchar(40)  | utf8_general_ci |          | Нет  |                     |                |
| <b>date</b>          | datetime     |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>last_access</b>   | datetime     |                 |          | Да   | NULL                |                |
| <b>allowed</b>       | varchar(1)   | utf8_general_ci |          | Нет  | Y                   |                |

Индексы:



| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 1                    | A         |      |
| file_id     | BTREE | Нет        | Нет      | file_id  | 1                    | A         |      |
| order_id    | BTREE | Нет        | Нет      | order_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **file\_id** – идентификатор файла цифрового продукта из таблицы [proshop\\_products\\_files](#).
- **order\_id** – идентификатор заказа (из таблицы [proshop\\_orders](#)), для которого был сгенерирован ключ **key**. Данное поле должно быть равно 0, если ключ **key** был сгенерирован не для заказа, а для скачивания бесплатного цифрового продукта.
- **order\_prod\_id** – идентификатор записи о заказанном товаре в таблице [proshop\\_orders\\_products](#). Данное поле должно быть равно 0, если ключ **key** был сгенерирован не для заказа, а для скачивания бесплатного цифрового продукта.
- **visitor\_id** – идентификатор посетителя (из таблицы [proshop\\_visitors](#)), для которого был создан ключ **key**.
- **email** – e-mail посетителя/покупателя, для которого был создан ключ **key**.
- **key** – секретный ключ для скачивания файла **file\_id**.
- **date** – дата и время создания ключа **key**.
- **last\_access** – дата и время последнего использования ключа **key** (другими словами, дата и время последней попытки скачивания файла **file\_id**).
- **allowed** – флаг доступности ключа: **Y** – ключ действителен и доступен, **N** – ключ временно заблокирован.

## proshop\_products\_files\_sessions

Назначение:

- Хранение временных данных о скачивающем бесплатный файл цифрового продукта пользователе. Предназначена для того, чтобы не дать скачать файл пользователю в несколько потоков.

Структура:

| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| ip        | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| uagent    | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **ip** – IP-адрес посетителя, который скачивает файл бесплатного цифрового продукта.

- **uagent** – строка User Agent браузера (или скачивающей программы) посетителя, который скачивает файл бесплатного цифрового продукта.

## proshop\_products\_options

Назначение:

- Хранение базовой информации об опциях продуктов.

Структура:

| Поле             | Тип        | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)    |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>type_id</b>   | int(11)    |                 |          | Нет  | <i>Нет</i>   |                |
| <b>prod_id</b>   | int(11)    |                 |          | Нет  | 0            |                |
| <b>pos</b>       | int(11)    |                 |          | Нет  | 0            |                |
| <b>is_global</b> | varchar(1) | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 5                    | A         |      |
| <b>prod_id</b> | BTREE | Нет        | Нет      | prod_id | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (опции продукта).
- **type\_id** – идентификатор типа опции продукта (из таблицы [proshop\\_products\\_options\\_types](#)).
- **prod\_id** – идентификатор продукта (из таблицы [proshop\\_products](#)), для которого определена текущая опция. Значение данного поля должно быть равно 0, если значение поля **is\_global** равно **Y**.
- **pos** – значение позиции опции по отношению к остальным опциям этого же продукта **prod\_id**. Данное поле должно быть равно 0, если поле **is\_global** имеет значение **Y**.
- **is\_global** – флаг, определяющий, является ли данная опция продукта глобальной: **Y** – опция является глобальной и может клонироваться при создании опций продуктов на ее основе, **N** – опция не является глобальной и принадлежит продукту **prod\_id**. Значение данного поля должно быть равно **N**, если значение поля **prod\_id** не равно 0.

## proshop\_products\_options\_def\_combs

Назначение:

- Хранение комбинаций вариантов опций продуктов по умолчанию, отображаемых при заходе на страницу (карточку) продукта на витрине.

Структура:

| Поле      | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|---------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11) |                 |          | Нет  | Нет          | auto_increment |
| prod_id   | int(11) |                 |          | Нет  | Нет          |                |
| vars      | text    | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 1                    | A         |      |
| prod_id     | BTREE | Да         | Нет      | prod_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта (из таблицы [proshop\\_products](#)), для которого в данной записи задана комбинация **vars** вариантов опций по умолчанию.
- **vars** – комбинация вариантов опций, являющаяся строкой, состоящей из идентификаторов вариантов опций (из таблицы [proshop\\_products\\_options\\_vars](#)), «склеяных» дефисом. Например: 12-24-25.

## proshop\_products\_options\_exceptions

Назначение:

- Хранение запрещенных комбинаций вариантов опций для заданных продуктов.

Структура:

| Поле      | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|---------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11) |                 |          | Нет  | Нет          | auto_increment |
| prod_id   | int(11) |                 |          | Нет  | Нет          |                |
| vars      | text    | utf8_general_ci |          | Да   | NULL         |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 0                    | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта (из таблицы [proshop\\_products](#)), для которого в данной записи задана запрещенная комбинация **vars** вариантов опций.
- **vars** – запрещенная комбинация вариантов опций, являющаяся строкой, состоящей из идентификаторов вариантов опций (из таблицы [proshop\\_products\\_options\\_vars](#)), «склеяных» дефисом. Например: 12-24-25.

## proshop\_products\_options\_texts

Назначение:

- Хранение названий опций продуктов, определенных в таблице [proshop\\_products\\_options](#), на разных языках.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>option_id</b> | int(11)      |                 |          | Нет  | Нет          |                |
| <b>lang_id</b>   | int(11)      |                 |          | Нет  | Нет          |                |
| <b>caption</b>   | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса      | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|------------------|-------|------------|----------|-----------|----------------------|-----------|------|
| <b>PRIMARY</b>   | BTREE | Да         | Нет      | id        | 10                   | A         |      |
| <b>option_id</b> | BTREE | Да         | Нет      | option_id | 5                    | A         |      |
|                  |       |            |          | lang_id   | 10                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **option\_id** – идентификатор опции продукта из таблицы [proshop\\_products\\_options](#), для которой задано название **caption** на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано название **caption** опции.
- **caption** – название опции продукта **option\_id** на языке **lang\_id**.

## proshop\_products\_options\_types

Назначение:

- Хранение типов опций продуктов, определенных в таблице [proshop\\_products\\_options](#).

Структура:

| Поле        | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>   | int(11)     |                 |          | Нет  | Нет          | auto_increment |
| <b>type</b> | varchar(16) | utf8_general_ci |          | Нет  | select       |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **type** – тип опции продукта, который определяет, каким образом будут выбираться варианты опций продукта на витрине в карточке продукта. Может принимать значения: **select** - варианты выбираются select-боксом.

## proshop\_products\_options\_vars

Назначение:

- Хранение вариантов опций продукта, определенных в таблице [proshop\\_products\\_options](#).

Структура:

| Поле                  | Тип           | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------------|---------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>             | int(11)       |                 |          | Нет  | Нет          | auto_increment |
| <b>option_id</b>      | int(11)       |                 |          | Нет  | Нет          |                |
| <b>pos</b>            | int(11)       |                 |          | Нет  | Нет          |                |
| <b>price_mod</b>      | decimal(12,2) |                 |          | Нет  | Нет          |                |
| <b>price_mod_type</b> | varchar(3)    | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id        | 13                   | A         |      |
| option_id   | BTREE | Нет        | Нет      | option_id | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (варианта).
- **option\_id** – идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **pos** – позиция данного варианта относительно других вариантов указанной опции **option\_id** для сортировки.
- **price\_mod** – модификатор цены продукта для данного варианта опции **option\_id**. Может быть положительным, либо отрицательным. Задается в процентах или в единицах базовой валюты магазина – в зависимости от значения поля **price\_mod\_type**.
- **price\_mod\_type** – тип модификатора цены **price\_mod** – абсолютный (**abs**) или в процентах (**per**).

## proshop\_products\_options\_vars\_combs

Назначение:

- Хранение комбинаций вариантов опций продуктов, из которых состоят варианты продуктов, определенные в таблице [proshop\\_products\\_variants](#).

Структура:

| Поле              | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>         | int(11) |           |          | Нет  | Нет          | auto_increment |
| <b>variant_id</b> | int(11) |           |          | Нет  | Нет          |                |
| <b>option_id</b>  | int(11) |           |          | Нет  | Нет          |                |
| <b>var_id</b>     | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id         | 4                    | A         |      |
| variant_id  | BTREE | Нет        | Нет      | variant_id | 2                    | A         |      |
| var_id      | BTREE | Нет        | Нет      | var_id     | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **variant\_id** – идентификатор варианта продукта из таблицы [proshop\\_products\\_variants](#), с которым связаны **option\_id** и **var\_id**.
- **option\_id** – идентификатор опции продукта из таблицы [proshop\\_products\\_options](#).
- **var\_id** – идентификатор варианта опции **option\_id** из таблицы [proshop\\_products\\_options\\_vars](#).

## proshop\_products\_options\_vars\_texts

Назначение:

- Хранение названий вариантов опций продуктов, определенных в таблице [proshop\\_products\\_options\\_vars](#), на разных языках.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>var_id</b>  | int(11)      |                 |          | Нет  | Нет          |                |
| <b>lang_id</b> | int(11)      |                 |          | Нет  | Нет          |                |
| <b>caption</b> | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 26                   | A         |      |
| var_id      | BTREE | Да         | Нет      | var_id  | 13                   | A         |      |
|             |       |            |          | lang_id | 26                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **var\_id** – идентификатор варианта опции продукта из таблицы [proshop\\_products\\_options\\_vars](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано название **caption**.
- **caption** – название варианта **var\_id** опции продукта на языке **lang\_id**.

## proshop\_products\_pictures

Назначение:

- Хранение информации о дополнительных изображениях продуктов.

Структура:

| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| prod_id   | int(11)      |                 |          | Нет  | 0            |                |
| name      | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| pos       | int(11)      |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 1                    | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#), имя файла изображения которого хранится в **name**.
- **name** – имя файла изображения продукта **prod\_id**. Файлы изображений продуктов хранятся в директории, путь к которой прописан в константе [NOWM\\_PRODS\\_PICS](#) и [WM\\_PRODS\\_PICS](#) (если включено наложение водяного знака). Данное поле может быть пустым.
- **pos** – позиция данного изображения относительно других дополнительных изображения продукта **prod\_id**. Используется для сортировки вывода изображений.

## proshop\_products\_pictures\_texts

Назначение:

- Хранение альтернативных текстов (для атрибута **alt** тэга **img**) дополнительных изображений продуктов, определенных в таблице [proshop\\_products\\_pictures](#), на разных языках.

Структура:

| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| pic_id    | int(11)      |                 |          | Нет  | 0            |                |
| lang_id   | int(11)      |                 |          | Нет  | 0            |                |
| alt       | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| pic_id      | BTREE | Да         | Нет      | pic_id  | 1                    | A         |      |
|             |       |            |          | lang_id | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **pic\_id** – идентификатор изображения продукта из таблицы [proshop\\_products\\_pictures](#), для которого задан альтернативный текст **alt**.

- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), для которого хранится текст **alt**.
- **alt** – альтернативный текст изображения **pic\_id** на языке **lang\_id**.

## proshop\_products\_pins

Назначение:

- Хранение секретных кодов цифровых продуктов для их продажи (например, коды карт оплаты мобильной связи).

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>prod_id</b> | int(11)      |                 |          | Нет  | Нет          |                |
| <b>code</b>    | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 6                    | A         |      |
| <b>prod_id</b> | BTREE | Да         | Нет      | prod_id | 2                    | A         |      |
|                |       |            |          | code    | 6                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор цифрового продукта из таблицы [proshop\\_products](#).
- **code** – значение секретного кода.

## proshop\_products\_recommended

Назначение:

- Хранение идентификаторов рекомендуемых продуктов для определенного продукта. Рекомендуемый продукт (назовем **продукт2**) отображается в карточке продукта (назовем **продукт1**). При этом **продукт1** не отображается в карточке **продукта2**.

Структура:

| Поле            | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>       | int(11) |           |          | Нет  | Нет          | auto_increment |
| <b>prod_id</b>  | int(11) |           |          | Нет  | 0            |                |
| <b>rprod_id</b> | int(11) |           |          | Нет  | 0            |                |
| <b>pos</b>      | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 5                    | A         |      |
| <b>prod_id</b> | BTREE | Нет        | Нет      | prod_id | 1                    | A         |      |



Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#), для которого задан рекомендуемый продукт **rprod\_id**.
- **rprod\_id** – идентификатор рекомендуемого продукта (из таблицы [proshop\\_products](#)) для продукта **prod\_id**.
- **pos** – позиция рекомендуемого продукта **rprod\_id** по отношению к другим рекомендуемым продуктам в пределах продукта **prod\_id**. Используется для сортировки.

## proshop\_products\_related

Назначение:

- Хранение идентификаторов родственных продуктов для определенного продукта. Родственный продукт (назовем **продукт2**) отображается в карточке продукта (назовем **продукт1**). При этом **продукт1** также отображается в карточке **продукта2**.

Структура:

| Поле            | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>       | int(11) |           |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>prod_id</b>  | int(11) |           |          | Нет  | 0            |                |
| <b>rprod_id</b> | int(11) |           |          | Нет  | 0            |                |
| <b>pos</b>      | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса     | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-----------------|-------|------------|----------|----------|----------------------|-----------|------|
| <b>PRIMARY</b>  | BTREE | Да         | Нет      | id       | 9                    | A         |      |
| <b>prod_id</b>  | BTREE | Нет        | Нет      | prod_id  | 9                    | A         |      |
| <b>rprod_id</b> | BTREE | Нет        | Нет      | rprod_id | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#), для которого задан родственный продукт **rprod\_id**.
- **rprod\_id** – идентификатор родственного продукта (из таблицы [proshop\\_products](#)) для продукта **prod\_id**.
- **pos** – позиция родственного продукта **rprod\_id** по отношению к другим родственным продуктам в пределах продукта **prod\_id**. Используется для сортировки.

## proshop\_products\_texts

Назначение:

- Хранение подробных описаний продуктов на разных языках.

Структура:

| Поле              | Тип     | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|---------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(32) |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>product_id</b> | int(11) |                 |          | Нет  | 0            |                |
| <b>lang_id</b>    | int(11) |                 |          | Нет  | 0            |                |
| <b>descr</b>      | text    | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса       | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY           | BTREE | Да         | Нет      | id         | 222                  | A         |      |
| <b>product_id</b> | BTREE | Да         | Нет      | product_id | 0                    | A         |      |
|                   |       |            |          | lang_id    | 222                  | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **product\_id** – идентификатор продукта из таблицы [proshop\\_products](#), для которого задано описание на языке **lang\_id**.
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано описание продукта **product\_id**.
- **descr** - полное описание продукта **product\_id** на языке **lang\_id**.

## proshop\_products\_variants

Назначение:

- Хранение информации о вариантах продуктов.

Структура:

| Поле           | Тип           | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|---------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)       |                 |          | Нет  | <i>Нет</i>   | AUTO_INCREMENT |
| <b>prod_id</b> | int(11)       |                 |          | Нет  | <i>Нет</i>   |                |
| <b>weight</b>  | decimal(12,2) |                 |          | Нет  | <i>Нет</i>   |                |
| <b>sku</b>     | varchar(100)  | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>qty</b>     | varchar(32)   | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>img</b>     | varchar(100)  | utf8_general_ci |          | Нет  |              |                |
| <b>thumb</b>   | varchar(100)  | utf8_general_ci |          | Нет  |              |                |
| <b>pos</b>     | int(11)       |                 |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| <b>prod_id</b> | BTREE | Нет        | Нет      | prod_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (варианта продукта).
- **prod\_id** – идентификатор продукта (из таблицы [proshop\\_products](#)), вариант которого описан в данной записи.

- **weight** – вес одного экземпляра данного варианта продукта в кг.
- **sku** – артикул данного варианта продукта.
- **qty** – доступное кол-во экземпляров данного варианта продукта на складе. Если равно пустой строке, кол-во считается бесконечным. Если равно 0, считается, что данный вариант отсутствует на складе.
- **img** – имя файла изображения данного варианта продукта. Файлы изображений продуктов хранятся в директории, путь к которой прописан в константе [NOWM\\_PRODS\\_PICS](#) и [WM\\_PRODS\\_PICS](#) (если включено наложение водяного знака). Данное поле может быть пустым.
- **thumb** – имя файла уменьшенного изображения данного варианта продукта. Файлы изображений продуктов хранятся в директории, путь к которой прописан в константе [NOWM\\_PRODS\\_PICS](#) и [WM\\_PRODS\\_PICS](#) (если включено наложение водяного знака). Данное поле может быть пустым.
- **pos** – позиция варианта для сортировки вывода вариантов в пределах конкретного продукта.

## proshop\_products\_wished

Назначение:

- Хранение списков желаний (отложенных товаров) зарегистрированных покупателей.

Структура:

| Поле       | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------|---------|-----------|----------|------|--------------|----------------|
| <u>id</u>  | int(11) |           |          | Нет  | Нет          | auto_increment |
| user_id    | int(11) |           |          | Нет  | 0            |                |
| product_id | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 4                    | A         |      |
| user_id     | BTREE | Нет        | Нет      | user_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **user\_id** – идентификатор зарегистрированного покупателя из таблицы [proshop\\_users](#), для которого задан идентификатор отложенного продукта **product\_id**.
- **product\_id** – идентификатор отложенного продукта из таблицы [proshop\\_products](#) для зарегистрированного покупателя **user\_id**.

## proshop\_reviews

Назначение:

- Хранение отзывов о магазине (сообщения гостевой книги).

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>author</b>   | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>email</b>    | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>comment</b>  | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>lang_id</b>  | int(11)      |                 |          | Нет  | Нет          |                |
| <b>date</b>     | datetime     |                 |          | Нет  | Нет          |                |
| <b>approved</b> | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 3                    | A         |      |
| lang_id     | BTREE | Нет        | Нет      | lang_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **author** – имя автора отзыва.
- **email** – e-mail автора отзыва.
- **comment** – текст отзыва.
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), который был выбран в момент написания отзыва.
- **date** – дата и время написания отзыва по часам сервера.
- **approved** – флаг того, что данный отзыв одобрен: **Y** – отзыв одобрен, **N** – отзыв еще не одобрен.

## proshop\_rtshippings

Назначение:

- Хранение информации о службах доставки с возможностью подсчета стоимости доставки в режиме он-лайн (таких, как **FedEx**).

Структура:

| Поле                  | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>             | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>group_id</b>       | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>           | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>caption</b>        | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>delivery_time1</b> | tinyint(4)   |                 |          | Нет  | 0            |                |
| <b>delivery_time2</b> | tinyint(4)   |                 |          | Нет  | 0            |                |
| <b>is_active</b>      | char(1)      | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 85                   | A         |      |
| group_id    | BTREE | Нет        | Нет      | group_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **group\_id** – идентификатор группы службы доставки из таблицы [proshop\\_rtshippings\\_groups](#), которой принадлежит данная служба доставки.
- **name** – служебное имя данной службы доставки.
- **caption** – название данной службы доставки.
- **delivery\_time1** – начальное значение диапазона возможного срока доставки заказа данной службой, в днях.
- **delivery\_time2** – конечное значение диапазона возможного срока доставки заказа данной службой, в днях.
- **is\_active** – флаг активности данной службы: **Y** – служба активна и может применяться, **N** – служба временно выключена (администратором).

## proshop\_rtshippings\_groups

Назначение:

- Хранение информации о группах служб доставок, которые позволяют рассчитать стоимость доставки заказа в режиме он-лайн (таких, как **FedEx**).

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>    | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>caption</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя службы доставки.
- **caption** – название службы доставки (например, **FedEx**).

## proshop\_rtshippings\_groups\_settings

Назначение:

- Хранение настроек служб доставки, определенных в таблице [proshop\\_rtshippings\\_groups](#).

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>group_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>      | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>value</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>maxlength</b> | tinyint(3)   |                 | UNSIGNED | Нет  | 254          |                |

Индексы:

| Имя индекса     | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-----------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY         | BTREE | Да         | Нет      | id       | 37                   | A         |      |
| <b>group_id</b> | BTREE | Нет        | Нет      | group_id | 1                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **group\_id** – идентификатор группы служб доставки из таблицы [proshop\\_rtshippings\\_groups](#).
- **name** – служебное имя настройки.
- **value** – значение настройки.
- **maxlength** – максимально возможная длина значения **value** настройки.

## proshop\_ruspost\_options

Назначение:

- Хранение некоторых настроек для службы доставки Почта России.

Структура:

| Поле         | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>    | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>name</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>value</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 6                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя настройки. Может принимать следующие значения:
  - **avia\_cost** – стоимость отправления заказа воздушным путем в единицах базовой валюты магазина;
  - **attention\_mark** – отправлять ли заказ с отметкой «Осторожно»: **Y** – отметка «Осторожно» присутствует, **N** – отметка отсутствует;
  - **max\_post\_weight** – максимальный вес заказа для отправления Почтой России, кг.;
  - **hard\_weight** – вес усложненной тарификации, кг.;

- **defined\_value\_cost** – размер платы за объявленную ценность в единицах базовой валюты магазина;
- **avia\_cost2** – надбавка за отправление заказа воздушным путем при усложненной тарификации в единицах базовой валюты магазина.
- **value** – значение настройки.

## proshop\_ruspost\_regions

Назначение:

- Хранение стоимости пересылки заказа для каждого магистрального пояса при пересылке заказа Почтой России.

Структура:

| Поле              | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|-------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)     |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>region_num</b> | tinyint(4)  |                 |          | Нет  | 0            |                |
| <b>cost</b>       | varchar(16) | utf8_general_ci |          | Нет  |              |                |
| <b>add_cost</b>   | varchar(16) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса       | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------------|-------|------------|----------|------------|----------------------|-----------|------|
| <b>PRIMARY</b>    | BTREE | Да         | Нет      | id         | 5                    | A         |      |
| <b>region_num</b> | BTREE | Да         | Нет      | region_num | 5                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **region\_num** – номер магистрального пояса.
- **cost** – стоимость пересылки заказа для данного магистрального пояса, если вес заказа не превышает 500 г., в базовой валюте магазина
- **add\_cost** – плата за каждые последующие полные/неполные 500 г. посылки для данного магистрального пояса, в базовой валюте магазина.

## proshop\_settings

Назначение:

- Хранение настроек магазина.

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>group_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>type_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>     | varchar(254) | utf8_general_ci |          | Да   | NULL         |                |
| <b>value</b>    | text         | utf8_general_ci |          | Да   | NULL         |                |
| <b>caption</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>descr</b>    | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>module</b>   | varchar(80)  | utf8_general_ci |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 94                   | A         |      |
| name        | BTREE | Да         | Нет      | name     | 94                   | A         | YES  |
| group_id    | BTREE | Нет        | Нет      | group_id | 18                   | A         |      |
|             |       |            |          | type_id  | 31                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **group\_id** – идентификатор группы настроек из таблицы [proshop\\_settings\\_groups](#), которой принадлежит данная настройка.
- **type\_id** – идентификатор типа настройки из таблицы [proshop\\_settings\\_types](#).
- **name** – служебное имя настройки.
- **value** – значение настройки. Должно соответствовать типу **type\_id**.
- **caption** – название настройки, отображаемое в админке в модуле настроек.
- **descr** – описание настройки, отображаемое в админке в модуле настроек.
- **module** – название модуля (из таблицы [proshop\\_modules](#)), которому принадлежит данная настройка (или названия разных модулей, разделенные точкой с запятой).

## proshop\_settings\_groups

Назначение:

- Хранение групп настроек, определенных в таблице [proshop\\_settings](#).

Структура:

| Поле            | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>       | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>group_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>  | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле     | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id       | 21                   | A         |      |
| group_id    | BTREE | Нет        | Нет      | group_id | 5                    | A         |      |



Поля:

- **id** – уникальный идентификатор записи (группы настроек).
- **group\_id** – идентификатор группы настроек из этой же таблицы, которая является родительской по отношению к данной группе.
- **caption** – название данной группы настроек.

## proshop\_settings\_types

Назначение:

- Хранение типов настроек, определенных в таблице [proshop\\_settings](#).

Структура:

| Поле        | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u>   | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <u>name</u> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 10                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи (типа настроек).
- **name** – служебное имя типа настройки. Может принимать следующие значения:
  - **string** – настройка данного типа является строкой, которая не может быть пустой. Выводится в админке в модуле настроек как `<input type="text" />`.
  - **checkbox** – настройка данного типа является флагом, принимающим значения **on** или **off**. Выводится в админке в модуле настроек как `<input type="checkbox" />`.
  - **country** – настройка данного типа является идентификатором страны из таблицы [proshop\\_countries](#). Выводится в админке в модуле настроек как `<select>`.
  - **email** – настройка данного типа является правильным e-mail адресом (в правильном формате). Выводится в админке в модуле настроек как `<input type="text" />`.
  - **natural\_number** – настройка данного типа является натуральным числом (от 1 до бесконечности). Выводится в админке в модуле настроек как `<input type="text" />`.
  - **positive\_float\_number** – настройка данного типа является позитивным числом с плавающей точкой. Выводится в админке в модуле настроек как `<input type="text" />`.
  - **state** – настройка данного типа является идентификатором области/провинции/штата из таблицы [proshop\\_states](#). Выводится в админке в модуле настроек как `<select>`.
  - **empty\_string** – настройка данного типа является строкой, которая может быть пустой. Выводится в админке в модуле настроек как `<input type="text" />`.
  - **number** – настройка данного типа является любым целым числом. Выводится в админке в модуле настроек как `<input type="text" />`.

- **empty\_textarea** – настройка данного типа является длинной строкой, которая может быть пустой. Выводится в админке в модуле настроек как **<textarea>**.

## proshop\_slider

Назначение:

- Хранение списка идентификаторов продуктов, связанных со слайдером витрины.

Структура:

| Поле      | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|---------|-----------|----------|------|--------------|----------------|
| <u>id</u> | int(11) |           |          | Нет  | Нет          | AUTO_INCREMENT |
| prod_id   | int(11) |           |          | Нет  | Нет          |                |
| pos       | int(11) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 6                    | A         |      |
| pos         | BTREE | Нет        | Нет      | pos     | 0                    | A         |      |
| prod_id     | BTREE | Нет        | Нет      | prod_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).
- **pos** – позиция расположения продукта в слайдере относительно других продуктов.

## proshop\_shippings

Назначение:

- Хранение информации о статических методах доставки (методах доставки, которые имеют фиксированные стоимость и время доставки заказа, не зависящие от веса заказа).

Структура:

| Поле           | Тип        | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|------------|-----------------|----------|------|--------------|----------------|
| <u>id</u>      | int(11)    |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| delivery_time1 | tinyint(4) |                 |          | Нет  | 0            |                |
| delivery_time2 | tinyint(4) |                 |          | Нет  | 0            |                |
| pos            | int(11)    |                 |          | Нет  | 0            |                |
| is_active      | char(1)    | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.

- **delivery\_time1** – начальное значение диапазона возможного срока доставки заказа данным методом, в днях.
- **delivery\_time2** – конечное значение диапазона возможного срока доставки заказа данным методом, в днях.
- **pos** – значение позиции для сортировки вывода статических доставок.
- **is\_active** – флаг активности данного метода: **Y** – метод активен и может применяться, **N** – метод временно выключен (администратором).

## proshop\_shippings\_costs

Назначение:

- Хранение диапазонов стоимости доставки (для доставок из таблицы [proshop\\_shippings](#)) в абсолютных и относительных величинах.

Структура:

| Поле        | Тип           | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------|---------------|-----------|----------|------|--------------|----------------|
| <u>id</u>   | int(11)       |           |          | Нет  | Нет          | auto_increment |
| shipping_id | int(11)       |           |          | Нет  | Нет          |                |
| low_band    | decimal(12,2) |           |          | Нет  | Нет          |                |
| cost        | decimal(12,2) |           |          | Нет  | Нет          |                |
| per         | decimal(12,2) |           |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id          | 6                    | A         |      |
| shipping_id | BTREE | Нет        | Нет      | shipping_id | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **shipping\_id** – идентификатор доставки из таблицы [proshop\\_shippings](#).
- **low\_band** – нижняя граница («от») диапазона суммы заказа в базовой валюте, либо веса заказа, если стоимость доставки рассчитывается по весу.
- **cost** – стоимость доставки при сумме (весе) заказа от **low\_band** и выше в базовой валюте (килограммах).
- **per** – значение процента от суммы заказа при сумме (весе) заказа от **low\_band** и выше.

## proshop\_shippings\_payments

Назначение:

- Хранение связей способов доставки со способами оплаты (для обеспечения возможности ограниченного выбора (в пределах способа доставки) способа оплаты при оформлении заказа покупателем).

Структура:

| Поле                 | Тип        | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------------|------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>            | int(11)    |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>shipping_type</b> | varchar(1) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>shipping_id</b>   | int(11)    |                 |          | Нет  | Нет          |                |
| <b>payment_id</b>    | int(11)    |                 |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле          | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id            | 5                    | A         |      |
| payment_id  | BTREE | Нет        | Нет      | payment_id    | 0                    | A         |      |
| sit         | BTREE | Нет        | Нет      | shipping_id   | 0                    | A         |      |
|             |       |            |          | shipping_type | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **shipping\_type** – тип способа доставки, указанного в поле **shipping\_id**: **s** – статический, **r** – реального времени, **c** – типа **custom**.
- **shipping\_id** – идентификатор способа доставки. В зависимости от значения в поле **shipping\_type**, может ссылаться на одну из следующих таблиц: [proshop\\_shippings](#), [proshop\\_rtshippings](#), [proshop\\_cshippings](#).
- **payment\_id** – идентификатор способа оплаты из таблицы [proshop\\_payments](#).

## proshop\_shippings\_texts

Назначение:

- Хранение названий статических методов доставки, определенных в таблице [proshop\\_shippings](#), на разных языках.

Структура:

| Поле               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>          | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>shipping_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b>     | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле        | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id          | 4                    | A         |      |
| shipping_id | BTREE | Да         | Нет      | shipping_id | 0                    | A         |      |
|             |       |            |          | lang_id     | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **shipping\_id** – идентификатор статического метода доставки из таблицы [proshop\\_shippings](#).

- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано название **caption**.
- **caption** – название метода **shipping\_id** на языке **lang\_id**.

## proshop\_shortcuts

Назначение:

- Хранение информации об URL-ярлыках в панели ярлыков, доступной в админке, для каждого администратора в отдельности.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>user_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>url</b>     | text         | utf8_general_ci |          | Нет  | Нет          |                |
| <b>caption</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 2                    | A         |      |
| <b>user_id</b> | BTREE | Нет        | Нет      | user_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **user\_id** – идентификатор администратора из таблицы [proshop\\_users](#), для которого предназначен данный ярлык.
- **url** – URL-адрес страницы, на которую указывает ярлык.
- **caption** – название ссылки (ярлыка), которая указывает на **url**.

## proshop\_smsgates

Назначение:

- Хранение информации о SMS-шлюзах, применяемых для оповещения администратора о поступлении заказа.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>      | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>is_active</b> | char(1)      | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя SMS-шлюза.
- **is\_active** – флаг активности флюза: **Y** – активен, **N** – не активен. В таблице может содержаться только одна запись с активным шлюзом. Остальные шлюзы должны быть неактивными.

## proshop\_smsgates\_settings

Назначение:

- Хранение настроек SMS-шлюзов, определенных в таблице [proshop\\_smsgates](#).

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>gate_id</b>   | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>      | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>value</b>     | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>maxlength</b> | tinyint(3)   |                 | UNSIGNED | Нет  | 254          |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 11                   | A         |      |
| <b>gate_id</b> | BTREE | Нет        | Нет      | gate_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **gate\_id** – идентификатор шлюза из таблицы [proshop\\_smsgates](#).
- **name** – служебное имя настройки.
- **value** – значение настройки.
- **maxlength** – максимально возможная длина значения **value** настройки.

## proshop\_states

Назначение:

- Хранение информации об областях/штатах/провинциях разных стран, определенных в таблице [proshop\\_countries](#).

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | <i>Нет</i>   | AUTO_INCREMENT |
| <b>country_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>       | varchar(254) | utf8_general_ci |          | Нет  | <i>Нет</i>   |                |
| <b>code</b>       | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>code_edost</b> | varchar(10)  | utf8_general_ci |          | Да   | NULL         |                |
| <b>pos</b>        | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |
| <b>region_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>avail_avia</b> | char(1)      | utf8_general_ci |          | Нет  | Y            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id         | 151                  | A         |      |
| country_id  | BTREE | Нет        | Нет      | country_id | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **country\_id** – идентификатор страны из таблицы [proshop\\_countries](#), которой принадлежит данная область.
- **name** – название области.
- **code** – обозначение области (например, для штата США данное обозначение состоит из двух букв, а для русского региона – из двух цифр).
- **code\_edost** – код области для сервиса расчета стоимости доставки eDost.
- **pos** – позиция данной области по отношению к другим областям страны **country\_id** в списке. Используется для сортировки вывода списка областей страны **country\_id**.
- **region\_id** – идентификатор магистрального пояса (для Почты России) из таблицы [proshop\\_ruspost\\_regions](#), которому принадлежит данная область. Должен быть равен 0 для областей, не принадлежащих территории РФ.
- **avail\_avia** – флаг авиадоставки заказа в данную область (для Почты России): **Y** - авиадоставка включена, **N** – авиадоставка выключена. Должен быть равен **N** для областей, не принадлежащих территории РФ.

## proshop\_stats\_cats\_views

Назначение:

- Хранение статистической информации о кол-ве просмотров различных товарных категорий магазина посетителями, определенных в таблице [proshop\\_categories](#).

Структура:

| Поле          | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>     | int(11) |           |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>cat_id</b> | int(11) |           |          | Нет  | 0            |                |
| <b>date</b>   | date    |           |          | Нет  | 0000-00-00   |                |
| <b>count</b>  | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле   | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|--------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id     | 257                  | A         |      |
| <b>cat_id</b>  | BTREE | Нет        | Нет      | cat_id | 32                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **cat\_id** – идентификатор категории из таблицы [proshop\\_categories](#), для которой хранится кол-во просмотров **count** в пределах даты **date**.
- **date** – дата просмотра категории **cat\_id**.
- **count** – кол-во просмотров категории **cat\_id** посетителями магазина в дату **date**.

## proshop\_stats\_search

Назначение:

- Хранение статистической информации о поисковых запросах, которые вводились в форму поиска в магазине.

Структура:

| Поле               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|--------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>          | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>search_text</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>date</b>        | date         |                 |          | Нет  | 0000-00-00   |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **search\_text** – текст поискового запроса.
- **date** – дата, когда был произведен данный поиск.

## proshop\_tags\_products

Назначение:

- Хранение связей продуктов с тэгами, определенными в таблице [proshop\\_tags\\_texts](#).

Структура:



| Поле           | Тип     | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|---------|-----------|----------|------|--------------|----------------|
| <b>id</b>      | int(11) |           |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>tag_id</b>  | int(11) |           |          | Нет  | 0            |                |
| <b>prod_id</b> | int(11) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 29                   | A         |      |
| <b>tag_id</b>  | BTREE | Нет        | Нет      | tag_id  | 0                    | A         |      |
| <b>prod_id</b> | BTREE | Нет        | Нет      | prod_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **tag\_id** – идентификатор тэга из таблицы [proshop\\_tags\\_texts](#).
- **prod\_id** – идентификатор продукта из таблицы [proshop\\_products](#).

## proshop\_tags\_texts

Назначение:

- Хранение тэгов магазина (для обеспечения работы облака тэгов).

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | <i>Нет</i>   | auto_increment |
| <b>lang_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>name</b>    | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>count</b>   | int(11)      |                 |          | Нет  | <i>Нет</i>   |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY        | BTREE | Да         | Нет      | id      | 10                   | A         |      |
| <b>lang_id</b> | BTREE | Да         | Нет      | lang_id | 0                    | A         |      |
|                |       |            |          | name    | 10                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **lang\_id** – идентификатор языка (из таблицы [proshop\\_languages](#)), которому принадлежит данный тэг.
- **name** – текст тэга на языке **lang\_id**.
- **count** – кол-во связей данного тэга с различными элементами (например, с продуктами в таблице [proshop\\_tags\\_products](#)).

## proshop\_taxes

Назначение:

- Хранение информации о налогах на продукты.

Структура:

| Поле          | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|---------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>     | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>type</b>   | varchar(254) | utf8_general_ci |          | Нет  |              |                |
| <b>value</b>  | varchar(32)  | utf8_general_ci |          | Нет  |              |                |
| <b>charge</b> | varchar(1)   | utf8_general_ci |          | Нет  | Y            |                |
| <b>no_vat</b> | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (налога).
- **type** – тип налога: **abs** – абсолютный в единицах базовой валюты магазина, **per** – в процентах.
- **value** – неотрицательное значение величины налога в зависимости от типа **type**.
- **charge** – если данное поле равно **Y**, налог считается начисляемым. Если **N** – налог выделяемый. Выделяемый налог – это налог, который уже входит в цену любого продукта, заданного в таблице [proshop\\_products](#).
- **no\_vat** – если данное поле равно **Y**, форма этого налога – «Без НДС». Значение **value** при этом должно быть равно 0.

## proshop\_taxes\_texts

Назначение:

- Хранение названий налогов, определенных в таблице [proshop\\_taxes](#), на разных языках.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>tax_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 6                    | A         |      |
| tax_id      | BTREE | Да         | Нет      | tax_id  | 0                    | A         |      |
|             |       |            |          | lang_id | 6                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **tax\_id** – идентификатор налога из таблицы [proshop\\_taxes](#).

- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано название **caption** налога.
- **caption** – название налога **tax\_id** на языке **lang\_id**.

## proshop\_url\_name\_history

Назначение:

- Хранение истории изменений имен в URL для таких объектов, как продукт, категория, статья и т.п. (для всех объектов, которые могут иметь имя в URL для modRewrite). История изменений предназначена для правильной переадресации пользователя или поисковой системы на новый URL-адрес объекта в случае смены URL-адреса этого объекта.

Структура:

| Поле             | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>        | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |
| <b>obj_id</b>    | int(11)      |                 |          | Нет  | Нет          |                |
| <b>obj_type</b>  | varchar(2)   | utf8_general_ci |          | Нет  | Нет          |                |
| <b>url_name</b>  | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>timestamp</b> | int(11)      |                 |          | Нет  | Нет          |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле      | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|-----------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id        | 6                    | A         |      |
| url_name    | BTREE | Нет        | Нет      | url_name  | 6                    | A         |      |
|             |       |            |          | obj_type  | 6                    | A         |      |
| timestamp   | BTREE | Нет        | Нет      | timestamp | 6                    | A         |      |
| obj_id      | BTREE | Нет        | Нет      | obj_id    | 3                    | A         |      |
|             |       |            |          | obj_type  | 3                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **obj\_id** – идентификатор объекта, тип которого указан в поле **obj\_type**.
- **obj\_type** – тип объекта (a – статья, ac – рубрика статьи, c – категория, p – продукт, m – производитель).
- **url\_name** – старое значение имени в URL объекта.
- **timestamp** – метка времени UNIX, когда произошла смена имени в URL объекта со значения **url\_name** на новое значение.

## proshop\_users

Назначение:

- Хранение информации о зарегистрированных пользователях магазина (в т.ч. администраторах).

Структура:

# Руководство разработчика Fast-Sales Pro

|                          | Поле             | Тип           | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  | Действие |  |  |  |  |  |
|--------------------------|------------------|---------------|-----------------|----------|------|--------------|----------------|----------|--|--|--|--|--|
| <input type="checkbox"/> | id               | int(11)       |                 |          | Нет  | Нет          | AUTO_INCREMENT |          |  |  |  |  |  |
| <input type="checkbox"/> | type_id          | int(11)       |                 |          | Нет  | 0            |                |          |  |  |  |  |  |
| <input type="checkbox"/> | group_id         | int(11)       |                 |          | Нет  | 0            |                |          |  |  |  |  |  |
| <input type="checkbox"/> | hash             | varchar(64)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | salt1            | varchar(16)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | salt2            | varchar(16)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | title            | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | first_name       | varchar(254)  | utf8_general_ci |          | Нет  |              |                |          |  |  |  |  |  |
| <input type="checkbox"/> | last_name        | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | second_name      | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | gender           | char(1)       | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | company_name     | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | ogrn             | varchar(60)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | inn              | varchar(60)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | kpp              | varchar(60)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | contact_person   | varchar(100)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | legal_address    | varchar(200)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | bank_name        | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | bank_bik         | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | bank_rs          | varchar(100)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | bank_ks          | varchar(100)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | phone            | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | mobile_phone     | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | fax              | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | address          | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | ship_address     | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | zip_code         | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | ship_zip_code    | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | city             | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | ship_city        | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | country          | int(11)       |                 |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | ship_country     | int(11)       |                 |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | state            | int(11)       |                 |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | ship_state       | int(11)       |                 |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | email            | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | website          | varchar(254)  | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | lang_id          | int(11)       |                 |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | curr_id          | int(11)       |                 |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | discount_id      | int(11)       |                 |          | Нет  | 0            |                |          |  |  |  |  |  |
| <input type="checkbox"/> | add_date         | date          |                 |          | Нет  | 0000-00-00   |                |          |  |  |  |  |  |
| <input type="checkbox"/> | partner_id       | int(11)       |                 |          | Нет  | 0            |                |          |  |  |  |  |  |
| <input type="checkbox"/> | total_balance    | decimal(12,2) |                 |          | Нет  | 0.00         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | cur_balance      | decimal(12,2) |                 |          | Нет  | 0.00         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | total_orders_sum | decimal(12,2) |                 |          | Нет  | 0.00         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | pwd_recover_key  | varchar(60)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | activation_key   | varchar(60)   | utf8_general_ci |          | Да   | NULL         |                |          |  |  |  |  |  |
| <input type="checkbox"/> | person_type      | varchar(1)    | utf8_general_ci |          | Нет  | N            |                |          |  |  |  |  |  |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 3                    | A         |      |
| type_id     | BTREE | Нет        | Нет      | type_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **type\_id** – идентификатор типа пользователя из таблицы [proshop\\_users\\_types](#).
- **group\_id** – идентификатор группы администраторов из таблицы [proshop\\_admins\\_groups](#). Должен быть равен 0 для суперадминистратора (у которого идентификатор равен 1) и для пользователей, не являющихся администраторами.
- **hash** – хэш-значение пароля пользователя.
- **salt1** – соль №1 для пароля пользователя.
- **salt2** – соль №2 для пароля пользователя.
- **title** – обращение к пользователю. Может принимать следующие значения: **MR** – господин, **MS** – госпожа. Данное поле недействительно для администраторов.
- **first\_name** – имя пользователя.
- **last\_name** – фамилия пользователя. Данное поле недействительно для администраторов.
- **second\_name** – отчество пользователя. Данное поле недействительно для администраторов.
- **gender** – пол пользователя: **M** – мужской, **F** – женский. Данное поле недействительно для администраторов.
- **company\_name** – название организации пользователя. Данное поле недействительно для администраторов.
- **ogrn** – ОГРН организации пользователя. Данное поле недействительно для администраторов.
- **inn** – ИНН организации пользователя. Данное поле недействительно для администраторов.
- **kpp** – КПП организации пользователя. Данное поле недействительно для администраторов.
- **contact\_person** – контактное лицо организации (если пользователь – юр. лицо). Данное поле недействительно для администраторов.
- **legal\_address** – юридический адрес организации (если пользователь – юр. лицо). Данное поле недействительно для администраторов.
- **bank\_name** – наименование банка пользователя (его организации). Данное поле недействительно для администраторов.
- **bank\_bik** – БИК банка покупателя. Данное поле недействительно для администраторов.
- **bank\_rs** – р/с банка покупателя. Данное поле недействительно для администраторов.
- **bank\_ks** – к/с банка покупателя. Данное поле недействительно для администраторов.
- **phone** – номера телефонов пользователя. Данное поле недействительно для администраторов.
- **mobile\_phone** – номер мобильного телефона пользователя. Данное поле недействительно для администраторов.
- **fax** – номер факса пользователя. Данное поле недействительно для администраторов.
- **address** – адрес (улица, номер дома и квартиры) пользователя. Данное поле недействительно для администраторов.

- **ship\_address** – другой адрес (улица, номер дома и квартиры) пользователя – для доставки заказа, если адрес проживания/офиса отличается от места доставки. Данное поле недействительно для администраторов.
- **zip\_code** – почтовый индекс адреса пользователя. Данное поле недействительно для администраторов.
- **ship\_zip\_code** – почтовый индекс другого адреса доставки заказа пользователю (если адрес проживания/офиса отличается от места доставки). Данное поле недействительно для администраторов.
- **city** – город местонахождения пользователя. Данное поле недействительно для администраторов.
- **ship\_city** – другой город для доставки заказа пользователю (покупателю). Данное поле недействительно для администраторов.
- **country** – идентификатор страны (из таблицы [proshop\\_countries](#)) местонахождения пользователя. Данное поле недействительно для администраторов.
- **ship\_country** – идентификатор другой страны (из таблицы [proshop\\_countries](#)) для доставки заказа пользователю (покупателю). Данное поле недействительно для администраторов.
- **state** – идентификатор области (из таблицы [proshop\\_states](#)) страны **country** местонахождения пользователя. Данное поле недействительно для администраторов.
- **ship\_state** – идентификатор области другой страны (из таблицы [proshop\\_states](#)) **ship\_country** для доставки заказа пользователю (покупателю). Данное поле недействительно для администраторов.
- **email** – адрес электронной почты пользователя. Используется в качестве логина.
- **website** – адрес сайта пользователя. Данное поле недействительно для администраторов.
- **lang\_id** – идентификатор предпочитаемого языка пользователя на витрине (из таблицы [proshop\\_languages](#)). Данное поле недействительно для администраторов.
- **curr\_id** – идентификатор предпочитаемой валюты пользователя на витрине (из таблицы [proshop\\_currencies](#)). Данное поле недействительно для администраторов.
- **discount\_id** – идентификатор скидки пользователя из таблицы [proshop\\_discounts](#). Данное поле недействительно для администраторов.
- **add\_date** – дата добавления данной записи о пользователе в таблицу.
- **partner\_id** – идентификатор пользователя из данной таблицы, который пригласил данного пользователя на сайт по партнерской программе. Данное поле недействительно для администраторов.
- **total\_balance** – общий объем заработков данного пользователя по партнерской программе в единицах базовой валюты магазина. Данное поле недействительно для администраторов.
- **cur\_balance** – текущий баланс данного пользователя в единицах базовой валюты магазина (для партнерской программы). Данное поле недействительно для администраторов.
- **total\_orders\_sum** – общая сумма всех завершенных заказов данного зарегистрированного покупателя в базовой валюте (с момента регистрации в магазине).

- **pwd\_recover\_key** – ключ восстановления пароля для данного пользователя. Генерируется только в том случае, когда пользователю на витрине требуется поменять пароль. В остальных случаях данное поле должно быть пустым. Данное поле недействительно для администраторов.
- **activation\_key** – ключ активации данного пользователя при его регистрации. Если данное поле не пусто, пользователь считается неактивированным. Данное поле недействительно для администраторов.
- **person\_type** – тип покупателя: **A** – юридическое лицо, **N** – физическое лицо.

## proshop\_users\_fields

Назначение:

- Хранение информации о полях профайла пользователя (покупателя) - их доступность, обязательность.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>name</b>       | varchar(254) | utf8_general_ci |          | Нет  | Нет          |                |
| <b>avail</b>      | varchar(1)   | utf8_general_ci |          | Нет  | Нет          |                |
| <b>required</b>   | varchar(1)   | utf8_general_ci |          | Нет  | Нет          |                |
| <b>additional</b> | varchar(1)   | utf8_general_ci |          | Нет  | N            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id   | 46                   | A         |      |
| <b>name</b>    | BTREE | Да         | Нет      | name | 46                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **name** – служебное имя поля профайла.
- **avail** – флаг доступности данного поля в профайле пользователя при его регистрации или редактировании информации о нем: **Y** – данное поле доступно, **N** – поле выключено.
- **required** – флаг обязательности заполнения данного поля в профайле пользователя при его регистрации или редактировании информации о нем: **Y** – поле обязательно для заполнения, **N** – поле не обязательно для заполнения.
- **additional** – флаг дополнительного поля (добавляемого администратором в модуле изменения полей профайлов). Зарезервирован на будущее. Должен быть всегда равен **N**.

## proshop\_users\_types

Назначение:

- Хранение типов пользователей, которые определены в таблице [proshop\\_users](#).

Структура:

| Поле        | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>   | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>type</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **type** – служебное обозначение типа пользователя. Может принимать следующие значения:  
ADMIN – администратор, CUSTOMER – покупатель.

## proshop\_visitors

Назначение:

- Хранение информации о посещениях магазина.

Структура:

| Поле              | Тип          | Сравнение       | Атрибуты | Null | По умолчанию        | Дополнительно  |
|-------------------|--------------|-----------------|----------|------|---------------------|----------------|
| <b>id</b>         | int(11)      |                 |          | Нет  | Нет                 | AUTO_INCREMENT |
| <b>user_id</b>    | int(11)      |                 |          | Нет  | 0                   |                |
| <b>partner_id</b> | int(11)      |                 |          | Нет  | 0                   |                |
| <b>ip</b>         | varchar(70)  | utf8_general_ci |          | Нет  |                     |                |
| <b>uagent</b>     | varchar(254) | utf8_general_ci |          | Нет  |                     |                |
| <b>referer</b>    | varchar(254) | utf8_general_ci |          | Нет  |                     |                |
| <b>first_time</b> | datetime     |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>last_time</b>  | datetime     |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>last_act</b>   | datetime     |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>count</b>      | int(11)      |                 |          | Нет  | 1                   |                |
| <b>unique</b>     | varchar(1)   | utf8_general_ci |          | Нет  | Y                   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 58                   | A         |      |
| ip          | BTREE | Да         | Нет      | ip      | 0                    | A         |      |
|             |       |            |          | uagent  | 58                   | A         |      |
| user_id     | BTREE | Нет        | Нет      | user_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **user\_id** – идентификатор пользователя из таблицы [proshop\\_users](#), которым является данный посетитель. Если данный посетитель не зарегистрирован в магазине как покупатель, данное поле должно быть равно 0.



- **partner\_id** – идентификатор пользователя-партнера из таблицы [proshop\\_users](#), который привлек данного пользователя на сайт.
- **ip** – IP-адрес посетителя (или цепочка IP-адресов, разделенных запятой, если посетитель пришел через прокси-сервер).
- **uagent** – строка User Agent браузера посетителя.
- **referer** – URL-адрес, с которого данный посетитель перешел в магазин.
- **first\_time** – дата и время первого посещения магазина данным посетителем.
- **last\_time** – дата и время последнего посещения магазина данным посетителем.
- **last\_act** – дата и время последнего перехода данным посетителем с одной страницы магазина на другую страницу магазина. Может использоваться для вычисления приблизительно времени пребывания данного посетителя на сайте.
- **count** – кол-во заходов данного посетителя на сайт (для подсчета кол-ва неуникальных посещений).

## proshop\_votes

Назначение:

- Хранение информации о голосованиях.

Структура:

| Поле          | Тип      | Сравнение       | Атрибуты | Null | По умолчанию        | Дополнительно  |
|---------------|----------|-----------------|----------|------|---------------------|----------------|
| <b>id</b>     | int(11)  |                 |          | Нет  | Нет                 | auto_increment |
| <b>date</b>   | datetime |                 |          | Нет  | 0000-00-00 00:00:00 |                |
| <b>active</b> | char(1)  | utf8_general_ci |          | Нет  | N                   |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id   | 2                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **date** – дата и время создания голосования по часам сервера.
- **active** – флаг активности данного голосования: **Y** – голосование активно, **N** – не активно. В таблице может содержаться только одно активное голосование.

## proshop\_votes\_antiflood

Назначение:

- Хранение информации о фактах голосования различными посетителями. Используется для обеспечения блокировки голосования одним и тем же пользователем более чем одного раза.

Структура:

| Поле       | Тип      | Сравнение | Атрибуты | Null | По умолчанию        | Дополнительно  |
|------------|----------|-----------|----------|------|---------------------|----------------|
| <u>id</u>  | int(11)  |           |          | Нет  | Нет                 | auto_increment |
| vote_id    | int(11)  |           |          | Нет  | 0                   |                |
| visitor_id | int(11)  |           |          | Нет  | 0                   |                |
| vote_time  | datetime |           |          | Нет  | 0000-00-00 00:00:00 |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле       | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|------------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id         | 6                    | A         |      |
| vote_id     | BTREE | Да         | Нет      | vote_id    | 0                    | A         |      |
|             |       |            |          | visitor_id | 6                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **vote\_id** – идентификатор голосования из таблицы [proshop\\_votes](#).
- **visitor\_id** – идентификатор проголосовавшего посетителя из таблицы [proshop\\_visitors](#).
- **vote\_time** – дата и время, когда посетитель **visitor\_id** оставил свой голос.

## proshop\_votes\_texts

Назначение:

- Хранение названий голосований, определенных в таблице [proshop\\_votes](#), на разных языках.

Структура:

| Поле      | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|-----------|--------------|-----------------|----------|------|--------------|----------------|
| <u>id</u> | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| vote_id   | int(11)      |                 |          | Нет  | 0            |                |
| lang_id   | mediumint(9) |                 |          | Нет  | 0            |                |
| caption   | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|-------------|-------|------------|----------|---------|----------------------|-----------|------|
| PRIMARY     | BTREE | Да         | Нет      | id      | 4                    | A         |      |
| vote_id     | BTREE | Да         | Нет      | vote_id | 0                    | A         |      |
|             |       |            |          | lang_id | 4                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **vote\_id** – идентификатор голосования из таблицы [proshop\\_votes](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задано название **caption** голосования.
- **caption** – название голосования **vote\_id** на языке **lang\_id**.

## proshop\_votes\_vars

Назначение:

- Хранение информации о вариантах ответов голосований, определенных в таблице [proshop\\_votes](#).

Структура:

| Поле           | Тип          | Сравнение | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |           |          | Нет  | Нет          | auto_increment |
| <b>vote_id</b> | int(11)      |           |          | Нет  | 0            |                |
| <b>votes</b>   | mediumint(9) |           |          | Нет  | 0            |                |
| <b>pos</b>     | mediumint(9) |           |          | Нет  | 0            |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 7                    | A         |      |
| <b>vote_id</b> | BTREE | Нет        | Нет      | vote_id | 0                    | A         |      |

Поля:

- **id** – уникальный идентификатор записи (варианта ответа голосования).
- **vote\_id** – идентификатор голосования из таблицы [proshop\\_votes](#), для которого определен данный вариант ответа.
- **votes** – кол-во голосов за данный вариант ответа.
- **pos** – позиция данного варианта ответа по отношению к другим вариантам ответа голосования **vote\_id**.

## proshop\_votes\_vars\_texts

Назначение:

- Хранение информации о текстах вариантов ответов (для голосований), определенных в таблице [proshop\\_votes\\_vars](#), на разных языках.

Структура:

| Поле           | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  |
|----------------|--------------|-----------------|----------|------|--------------|----------------|
| <b>id</b>      | int(11)      |                 |          | Нет  | Нет          | auto_increment |
| <b>var_id</b>  | int(11)      |                 |          | Нет  | 0            |                |
| <b>lang_id</b> | int(11)      |                 |          | Нет  | 0            |                |
| <b>caption</b> | varchar(254) | utf8_general_ci |          | Нет  |              |                |

Индексы:

| Имя индекса    | Тип   | Уникальный | Упакован | Поле    | Уникальных элементов | Сравнение | Null |
|----------------|-------|------------|----------|---------|----------------------|-----------|------|
| <b>PRIMARY</b> | BTREE | Да         | Нет      | id      | 14                   | A         |      |
| <b>var_id</b>  | BTREE | Да         | Нет      | var_id  | 0                    | A         |      |
|                |       |            |          | lang_id | 14                   | A         |      |

Поля:

- **id** – уникальный идентификатор записи.
- **var\_id** – идентификатор варианта ответа из таблицы [proshop\\_votes\\_vars](#).
- **lang\_id** – идентификатор языка из таблицы [proshop\\_languages](#), на котором задан текст **caption** варианта ответа **var\_id**.
- **caption** – текст варианта ответа **var\_id** на языке **lang\_id**.

## 8.8 Описание конфигурационных файлов

В CMS Fast-Sales Pro имеется два конфигурационных файла, содержимое которых можно изменять на свое усмотрение в разумных пределах:

1. `includes/config.php` (далее просто **config.php**)
2. `includes/defs.php` (далее просто **defs.php**)

### 8.8.1 includes/config.php

В файле **config.php** определяются и хранятся:

- версия и модификация CMS (см. описание констант [SHOP\\_VERSION](#) и [SHOP\\_MODIF](#));
- глобальные массивы таблиц БД и их связей (см. описание [\\$TABLES](#) и [\\$TABLES\\_RELS](#));
- префикс таблиц БД (в переменной **\$\_tbl\_pref**);
- различные [константы](#), определяющие пути к различным директориям, и прочие константы;
- установки директив **php.ini**, необходимых для корректной работы CMS.

Файл **config.php** чаще всего изменяется по следующим причинам:

- установка обновления CMS через модуль обновления;
- добавление/изменение таблицы в БД;
- добавление/изменение пути к какой-либо директории.

Без необходимости данный файл изменять не рекомендуется.

### 8.8.2 includes/defs.php

В данном файле хранятся настройки в виде констант, которые разработчик, в отличие от [config.php](#), может свободно менять:

- настройки подключения к БД;
- включение/отключение режима отладки;

- включение/отключение кэширования;
- прочие настройки.

Файл **defs.php** редко затрагивается модулем обновления.