Search our catalog

Architecture

PrestaShop v1.6 Architecture

PrestaShop uses Model-View-Controller (MVC) for its software architecture, which is often used in web applications where webpages are dynamically generated. MVC isolates application logic from the input and presentation, so that they can be separately developed, tested and maintained.

A **view** provides the user interface presented to the customer in the browser. A **controller** receives input from the browser, decides what to do with it and then passes information to **models** that carry out specific tasks.

In PrestaShop 1.6, the models are in the `classes` directory, the views are in the `themes` directory and the controllers are in the `controllers` directory. The files in the root directory are the public interface used to call the controllers.

Everything passes through `index.php` in the root directory to improve sharing of code and make processing friendly URLs easier. There are also `index.php` files in all subdirectories of PrestaShop, but these are used only to prevent the contents of the directories from being displayed in browsers. Some subdirectories have an .htaccess file too to allow or block access to individual files.

All the models, views and controllers have a warning at the top of them to not modify the file if the website will be upgraded to a new version of PrestaShop in the future. This is because PrestaShop allows these files to be overridden.

Classes and controllers are overridden using the `override` directory. By overriding classes and controllers instead of modifying them, they can be safely overwritten by a new version of PrestaShop in the future without losing any changes that were made.

PrestaShop's modules also have the same warning, since they can be overridden in a theme by mirroring the module's directory inside the theme, e.g. `themes/themename/modules/blockuserinfo/blockuserinfo.tpl` .

Models

The models in the `classes` directory of PrestaShop are all PHP classes that extend the `ObjectModel` class. This class gives the models a unique ID and a table name and identifier to link them to a row in the database.

For example, in the `Address` model, the ID is a positive integer, the table name is `address` and the table identifier is `id_address` . The `ObjectModel` class also defines the model's field sizes and the functions in the `Validate` class that are used to validate fields.

It also links the model to the webservice feature and provides a cache. The model class names all end in `Core` so they can be overridden in the `override/classes` directory. Here is a description of each PrestaShop model:

Model Filename	Description
order/Order.php	Customer order information.
order/OrderDetail.php	Customer order products.
order/OrderDiscount.php	**(deprecated)** Customer order vouchers.
order/OrderHistory.php	Customer order status history.
order/OrderMessage.php	Prefilled order messages that employees can use.
order/OrderReturn.php	Merchandise returns.
order/OrderReturnState.php	Merchandise return statuses.
order/OrderSlip.php	Credit slips.
order/OrderState.php	Order statuses.
range/RangePrice.php	Carrier price ranges.
range/RangeWeight.php	Carrier weight.
shop/Shop.php	Shops (multistore).
shop/ShopGroup.php	Shop groups (multistore).
shop/ShopURL.php	Shop URLs (multistore).
stock/Stock.php	Products kept in warehouses (advanced stock management).
stock/StockAvailable.php	Quantity of products kept in warehouses (advanced stock management).
stock/StockMvt.php	Stock movements (advanced stock management).
stock/StockMvtReason.php	Stock movement reasons (advanced stock management).
stock/StockMvtWs.php	Stock movements through Webservice (advanced stock management).
stock/SupplyOrder.php	Supply orders (advanced stock management).
stock/SupplyOrderDetail.php	Supply order products (advanced stock management).
stock/SupplyOrderHistory.php	Supply order status history (advanced stock management).
stock/SupplyOrderReceiptHistory.php	Supply order receipt history (advanced stock management).
stock/SupplyOrderState.php	Supply order statuses (advanced stock management).
stock/Warehouse.php	Warehouses
stock/WarehouseProductLocation.php	Warehouse product location reference codes.
tax/Tax.php	Tax names and rates.
tax/TaxRules.php	Specific tax rules.
tax/TaxRulesGroup.php	Groups of the tax rules.
webservice/WebserviceKey.php	Webservice account keys.
Address.php	Customer, manufacturer and supplier addresses.

AddressFormat.php	Address format for each country.
Alias.php	Aliases used to redirect typos in the search box in the Front Office.
Attachment.php	File attachments that are added to products.
Attribute.php	Product attributes that display as dropdowns on the product page.
AttributeGroup.php	Product attribute groups that display as labels for the dropdowns.
Carrier.php	Carriers used to ship products to customers.
Cart.php	Customer carts including those that didn't result in an order.
CartRule.php	Customer cart rules including reductions, free shipping and gifts.
Category.php	Product categories used to organise products.
CMS.php	CMS pages used to display information.
CMSCategory.php	Categories used to organise CMS pages.
CMSRole.php	Used by Advanced EU Compliance module to link CMS pages to legal roles.
Combination.php	Combinations of product attributes and their prices.
CompareProduct.php	Customer comparisons of products.
Configuration.php	Non-translated configuration settings.
ConfigurationKPI.php	Key Performance Indicators (KPIs) displayed on Back Office tabs.
Connection.php	Statistics about page visits.
ConnectionsSource.php	Statistics about the source of page visits.
Contact.php	Departments that can be selected on the contact form.
Country.php	Countries used in addresses.
County.php	**(deprecated)** Counties used in addresses.
Currency.php	Currencies used in prices.
Customer.php	Customers who have registered.
CustomerMessage.php	Individual messages sent between customers and employees.
CustomerThread.php	Threads of messages sent between customers and employees.
Customization.php	Product customization data associated with customer carts.
CustomizationField.php	Image and text customization fields displayed on product pages.
DateRange.php	Date ranges used in page visits.
Delivery.php	Links between carriers, zones and ranges.
Discount.php	**(deprecated)** Vouchers that can be used by customers.
Employee.php	Employees who can log in to the Back Office.

Feature.php	Product feature labels on the product page.
FeatureValue.php	Product feature values on the product page.
Group.php	Customer groups and display methods.
GroupReduction.php	Customer group reductions.
Guest.php	Customers who haven't registered.
Hook.php	Hooks where modules can be placed.
Image.php	Product images.
ImageType.php	Image thumbnail sizes.
Language.php	Languages used on the website.
Mail.php	Log entries of e-mails sent and received.
Manufacturer.php	Product manufacturers.
Message.php	Customer messages sent from the order history page.
Meta.php	Meta-information of PrestaShop's pages.
Pack.php	Product packs (extends Product class).
Page.php	PrestaShop's non-CMS pages.
PaymentCC.php	**(deprecated)** Order credit card information.
PrestaShopLogger.php	Log entries for employee log-ins, modifications and errors.
Product.php	Product information.
ProductDownload.php	Downloadable products.
ProductSupplier.php	Supplier product reference codes and prices.
Profile.php	Profiles used to restrict employee access.
QuickAccess.php	Shortcuts in the dropdown at the top of the Back Office.
RangePrice.php	Carrier price ranges.
RangeWeight.php	Carrier weight ranges.
Referrer.php	Affiliate program data.
RequestSql.php	Saved queries from the SQL Manager tab.
Risk.php	Business-to-Business (B2B) risk ratings.
Scene.php	Image maps that link products to banners.
SearchEngine.php	Search engines tracked when guests visit.
SpecificPrice.php	Specific prices including quantity discounts for a product.
SpecificPriceRule.php	Specific price rules applied to multiple products.
State.php	States used in addresses.
Store.php	Stores used in the store locator module.
StockMvt.php	Stock movements used to track employees' changes.
StockMvtReason.php	Stock movement reasons.
SubDomain.php	Subdomains used by PrestaShop's cookie.
Supplier.php	Product suppliers.
Tab.php	Back Office tabs and positions.
Tag.php	Product tags used by the tag cloud module.
Theme.php	Theme settings.

Zone.php	Zones used by the carriers.

Here's a description of the other PrestaShop classes that do not store data in the database:

Class Filename	Description
cache/Cache.php	Provides basic caching functionality.
cache/CacheApc.php	Performs APC caching.
cache/CacheFS.php	Performs File System caching.
cache/CacheMemcache.php	Performs Memcache caching.
cache/CacheMemcached.php	Performs Memcached caching.
cache/CacheXcache.php	Performs Xcache caching.
controller/AdminController.php	Provides basic Back Office tab functionality.
controller/Controller.php	Provides basic controller functionality such as adding CSS and JavaScript.
controller/FrontController.php	Initialises Smarty variables and other basic controller code.
controller/ModuleAdminController.php	Used by modules to create a Back Office tab.
controller/ModuleFrontController.php	Used by modules to create a Front Office page.
db/Db.php	Provides basic database functionality.
db/DbMySQLi.php	Connects to a MySQL database using MySQLi.
db/DbPDO.php	Connects to a MySQL database using PDO.
db/DbQuery.php	Generates an SQL query.
db/MySQL.php	Provides access to a MySQL database.
exception/PrestaShopDatabaseException.php	Displays a MySQL database error message.
exception/PrestaShopException.php	Displays a 500 Internal Server Error and backtrace for debugging.

exception/PrestaShopModuleException.php	Displays a module error.
exception/PrestaShopPaymentException.php	Displays a payment error.
helper/Helper.php	Provides basic helper functionality.
helper/HelperCalendar.php	Displays a calendar in the Back Office.
helper/HelperForm.php	Displays a form in the Back Office.
helper/HelperImageUploader.php	Validates image uploads.
helper/HelperKpi.php	Displays a group of Key Performance Indicators (KPIs).
helper/HelperKpiRow.php	Displays a Key Performance Indicator (KPI).
helper/HelperList.php	Displays a list of objects in the Back Office.
helper/HelperOptions.php	Displays a configuration page.
helper/HelperShop.php	Displays a dropdown of shop contexts in the Back Office.
helper/HelperTreeCategories.php	Displays a tree of categories in the Back Office.
helper/HelperTreeShops.php	Displays a tree of shops in the Back Office.
helper/HelperUploader.php	Displays a file uploader in the Back Office.
helper/HelperView.php	Displays an individual item in the Back Office.
log/AbstractLogger.php	Provides basic logging functionality.
log/FileLogger.php	Saves logs to a file.
module/CarrierModule.php	Provides basic shipping module functionality.
module/ImportModule.php	Provides basic shop import functionality.
module/Module.php	Provides basic module functionality.
module/ModuleGraph.php	Provides module graph functionality.

module/ModuleGraphEngine.php	Provides module graph engine functionality.
module/ModuleGrid.php	Provides module grid functionality.
module/ModuleGridEngine.php	Provides module grid engine functionality.
pdf/HTMLTemplate.php	Sets the data for the header and footer of PDF files.
pdf/HTMLTemplateDeliverySlip.php	Sets the data and filename of the delivery slip.
pdf/HTMLTemplateInvoice.php	Sets the data and filename of the invoice.
pdf/HTMLTemplateOrderReturn.php	Sets the data and filename of the merchandise return.
pdf/HTMLTemplateOrderSlip.php	Sets the data and filename of the credit slip.
pdf/HTMLTemplateSupplyOrderForm.php	Sets the data and filename of the supply order form.
pdf/PDF.php	Designs the PDF files using the HTML templates.
pdf/PDFGenerator.php	Generates PDF files using TCPDF.
stock/StockManager.php	The default stock manager (advanced stock management).
stock/StockManagerFactory.php	Gets a stock manager (advanced stock management).
stock/StockManagerInterface.php	The interface for managing stock (advanced stock management).
stock/StockManagerModule.php	Enables a module to manage stock (advanced stock management).
tax/AverageTaxOfProductsTaxCalculator.php	Calculates the average tax of products in an order.
tax/TaxCalculator.php	Calculates taxes.
tax/TaxManagerFactory.php	Gets a tax manager.
tax/TaxManagerInterface.php	The interface for managing taxes.

tax/TaxManagerModule.php	Enables a module to manage tax.
tax/TaxRulesTaxManager.php	The default tax manager.
tree/ITreeToolbar.php	The interface for tree toolbars in the Back Office.
tree/ITreeToolbarButton.php	The interface for tree toolbar buttons in the Back Office.
tree/Tree.php	Creates a tree in the Back Office.
tree/TreeToolbar.php	Creates a tree toolbar in the Back Office.
tree/TreeToolbarButton.php	Creates a tree toolbar button in the Back Office.
tree/TreeToolbarLink.php	Creates a tree toolbar link in the Back Office.
tree/TreeToolbarSearch.php	Creates a tree toolbar search box in the Back Office.
tree/TreeToolbarSearchCategories.php	Creates a tree toolbar search box for categories in the Back Office.
webservice/WebserviceException.php	Displays a Webservice error.
webservice/WebserviceOutputBuilder.php	Builds Webservice output.
webservice/WebserviceOutputInterface.php	The interface for displaying Webservice output.
webservice/WebserviceOutputJSON.php	Displays Webservice output in JSON format.
webservice/WebserviceOutputXML.php	Displays Webservice output in XML format.
webservice/WebserviceRequest.php	Processes a Webservice request.
webservice/WebserviceSpecificManagementImages.php	Manages images through Webservice.
webservice/WebserviceSpecificManagementInterface.php	The interface for managing specific data.
webservice/WebserviceSpecificManagementSearch.php	Manages specific queries on the data through the Webservice.

AdminTab.php	**(deprecated)** Provides basic Back Office tab functionality.
Blowfish.php	Encrypts the PrestaShop cookie with Blowfish.
Chart.php	Creates charts.
ConfigurationTest.php	Tests whether the server configuration is compatible with PrestaShop.
Context.php	Used to get all the data related to the current customer or employee.
ControllerFactory.php	**(deprecated)** Gets a controller or its override.
Cookie.php	Reads and writes PrestaShop's cookie.
CSV.php	Generates a CSV file.
Dispatcher.php	Calls a controller.
FileUploader.php	Performs file uploads.
ImageManager.php	Generates image thumbnails.
Link.php	Gets links to products, categories, CMS pages, etc.
LocalizationPack.php	Installs localisation packs.
Mail.php	Sends emails.
Media.php	Manages CSS, JavaScript and minification.
Notification.php	Handles notifications in the Back Office.
PaymentModule.php	Provides basic payment module functionality.
PrestaShopAutoload.php	Automatically loads classes when needed.
PrestaShopBackup.php	Manages PrestaShop backups.
PrestaShopCollection.php	Manages SQL queries and results.
ProductSale.php	Manages product sale numbers.
Rijndael.php	Encrypts PrestaShop's cookie using mcrypt.
Search.php	Performs searchs on products.

SmartyCacheResourceMysql.php	Tracks compiled and cached Smarty templates.
SmartyCustom.php	Manages Smarty caching and flushing.
Tools.php	Contains many useful conversion and sanitization functions.
Translate.php	Gets translations.
Upgrader.php	Compares PrestaShop version and files for changes.
Uploader.php	Handles file uploads.
Validate.php	Validates data.

Here's a description of the ObjectModel members:

Member Name	Description
$cache_objects	Whether the objects can be cached.
$db	The database manager.
$def	The current object's definition.
$definition	Defines the default table, identifier and fields of objects.
$fieldsRequired	**(deprecated)** The required fields.
$fieldsRequiredDatabase	The required fields that have been overridden by an employee.
$fieldsSize	**(deprecated)** The maximum size of each non-translatable field.
$fieldsSizeLang	**(deprecated)** The maximum size of each translatable field.
$fieldsValidate	**(deprecated)** Array of validation functions that check non-translatable fields.
$fieldsValidateLang	**(deprecated)** Array of validation functions that check translatable fields.
$force_id	Whether to use the specified ID when adding the object.
$get_shop_from_context	Whether to get the shop from the current context.
$id	The integer ID of the object, e.g. 1
$id_lang	The language ID used to select translations of the object.
$id_shop	The shop ID used to choose which shop to get data from.
$id_shop_list	The IDs of every shop the object is associated with.
$identifier	**(deprecated)** The name of the ID of the object, e.g. `id_address`
$image_dir	The image directory used when deleting objects.
$image_format	The image extension used when deleting objects.
$loaded_classes	Contains a list of loaded classes.
$table	**(deprecated)** The main table used by the object.
$tables	**(deprecated)** All the tables used by the object.
$update_fields	Used to update specific fields (or `null` for all fields).
$webserviceParameters	The webservice parameters for the object.

Here's a description of the ObjectModel functions:

Function Name	Description
__construct()	Loads an object with the specified ID, language and shop.
add()	Adds the current object to the database.
addFieldsRequiredDatabase()	Adds the fields marked as required to the database.
associateTo()	Associates the object with the specified shops.
cacheFieldsRequiredDatabase()	Caches the fields marked as required.
clearCache()	Used to clear the cache whenever an object is modified.
delete()	Deletes the current object from the database.
deleteImage()	Deletes images associated with the object.
deleteSelection()	Deletes the specified list of objects from the database.
disableCache()	Disables caching of objects.
displayFieldName()	Gets the unique name of a field.
duplicateObject()	Creates a duplicate of the object, its translations and shops.
duplicateShops()	Copies the object's shop associations to another object.
enableCache()	Enabls caching of the object.
existsInDatabase()	Checks whether the object's ID exists in the database.
formatFields()	Formats the fields based on their specified types.
formatValue()	Formats a specific value.
getAssociatedShops()	Gets all shops associated with the object.
getDefinition()	Gets the object's definition.
getFieldByLang()	Gets the value of the field for the specified language.
getFields()	Gets the non-translatable fields for the object.
getFieldsShop()	Gets fields for the current shop.
getFieldsRequiredDatabase()	Gets the fields marked as required from the database.
getTranslationFields()	**(deprecated)** Gets the translatable fields for the object.
getValidationRules()	Gets the validation rules for the object.
getWebserviceObjectList()	Gets a list of all webservice objects.
getWebserviceParameters()	Gets the webservice parameters for the object.
hasMultishopEntries()	Checks whether the object is in multiple shops.
hydrate()	Fills the object with the specified data.
hydrateCollection()	Fills a collection of objects with the specified data.
isAssociatedToShop()	Checks whether the object is associated with the specified shop.
isCurrentlyUsed()	Checks whether the object exists and is active.
isLangMultishop()	Checks whether the object has shop-specific translations.
isMultishop()	Checks whether the object is shop-specific.

isMultiShopField()	Checks whether a field is shop-specific.
makeTranslationFields()	**(deprecated)** Creates translation fields for the specified language.
save()	Saves the object, adding it or updating it as necessary.
setDefinitionRetrocompatibility()	Creates a definition using the deprecated fields.
setFieldsToUpdate()	Sets which fields to update when the `update()` function is called.
toggleStatus()	Toggles whether the object is visible on the website.
update()	Updates the current object in the database.
updateMultishopTable()	Updates the shop-specific data.
validateControler()	**(deprecated)** Checks whether required fields have valid data.
validateController()	Checks whether required fields have valid data.
validateField()	Validates a single field value.
validateFields()	Checks whether required non-translatable fields have valid data.
validateFieldsLang()	Checks whether required translatable fields have valid data.
validateFieldsRequiredDatabase()	Validates all fields marked as required in the database.

Here's a description of the validation functions:

Function Name	Description
isAbsoluteUrl()	Is an absolute URL (e.g., http://www.domain.com).
isAddress()	Is an address line (^!<>?=+@{}_$% not allowed).
isAnything()	Accepts anything without doing any validation.
isApe()	Is a valid APE code (3 or 4 numbers followed by a letter).
isArrayWithIds()	Is an array containing only positive integers.
isBirthDate()	Is empty date or a date in yyyy-mm-dd format at least 9 years ago.
isBool()	Is a Boolean value 0 or 1.
isBool_Id()	**(deprecated)** Is a Boolean id (0 or 1 followed by an underscore, then digits).
isBoolId()	Is a Boolean id (0 or 1 followed by an underscore, then digits).
isCarrierName()	Is a valid carrier name (^<>;=#{} not allowed).
isCatalogName()	Is a catalog name (^<>;=#{} not allowed).
isCityName()	Is a city name (^!<>;?=+@#"Â°{}_$% not allowed).
isCleanHtml()	Is HTML without script tags or events.
isColor()	Is a colour keyword or hexadecimal code.
isConfigName()	Is a configuration key (alphanumeric and hypens).
isControllerName()	Is a controller name (non-Unicode alphanumeric, hyphens and underscores).
isCookie()	Is an instance of the Cookie class.
isCoordinate()	Is a coordinate (up to 6 digits, a period, then up to 6 digits).

isCountryName()	Is a country name (lowercase letters, spaces and hyphens).
isDate()	Is a date in yyyy-mm-dd format.
isDateFormat()	Is a valid date in yyyy-mm-dd format.
isDirName()	Is a directory name (alphanumeric, hyphens, underscores and periods).
isDiscountName()	Is a valid voucher code (3-32 characters except ^!<>,;?=+()@"Â°{}_$%:).
isDistanceUnit()	Is a distance unit (up to 2 letters).
isDniLite()	Is a DNI number (1 to 13 alphanumeric with hyphens and periods).
isEan13()	Is a 13-digit EAN13 code.
isEmail()	Is an email of the form username@domain.tld.
isFileName()	Is a valid filename (alphanumeric, hyphens, underscores and periods).
isFloat()	Is a valid float value.
isGenericName()	Is a generic name (^<>;=#{} not allowed).
isHookName()	Is a valid hook name (lowercase alphanumeric and hyphens).
isImageSize()	Is an image size (up to 4 digits).
isImageTypeName()	Is a valid image type name (lowercase alphanumeric, spaces and hyphens).
isImageSize()	Is an image size (an integer with a length 1 and 4 numbers).
isInt()	Is an integer.
isIp2Long()	Is a valid IP address.
isLabel()	Is a customisation field label (^{}<> allowed).
isLangIsoCode()	Is a language iso code (2 or 3 alphanumeric).
isLanguageCode()	Is a language code (2 letters, a hyphen and another 2 letters).
isLanguageFileName()	Is a language filename (2 or 3 alphanumeric followed by .gzip or .tar.gz).
isLanguageIsoCode()	Is a valid language ISO code (2 or 3 lowercase letters).
isLinkRewrite()	Is a rewritten link (lowercase alphanumeric, hypens and underscores).
isLoadedObject()	Is a variable is an object and has a valid ID.
isLocalizationPackSelection()	Is "states", "taxes", "currencies", "languages" or "units".
isMailName()	Is a valid mail name (^<>;=#{} not allowed).
isMailSubject()	Is a valid mail subject (^<>{} not allowed).
isMessage()	Is a message (<>{} not allowed except <br />).
isMd5()	Is a valid MD5 hash.
isModuleName()	Is a valid module name (lowercase alphanumeric and hyphens).
isModuleUrl()	Is a valid URL to a zip, tgz or tar.gz file.
isMySQLEngine()	Is a MySQL Engine (InnoDB or MySAM)
isName()	Is a valid name (^0-9!<>,;?=+()@#"Â°{}_$%: not allowed).

isNegativePrice()	Is a positive or negative price (an optional hyphen, up to 10 digits, a period, then up to 9 digits).
isNumericIsoCode()	Is a numeric ISO code (2 or 3 digits).
isOptFloat()	Is an empty field or valid float value.
isOrderInvoiceNumber()	Is an order invoice number (the configured invoice prefix followed by numbers).
isOrderWay()	Is a valid order way 'ASC', 'DESC', 'asc' or 'desc'.
isOrderBy()	Is a valid order (lowercase alphanumeric, hyphens and underscores).
isPasswd()	Is a customer password (5-32 alphanumeric and (). -!@#$%^&*).
isPasswdAdmin()	Is an employee password (8-32 alphanumeric and (). -!@#$%^&*).
isPercentage()	Is a percentage (a decimal number between 0 and 100).
isPhpDateFormat()	Is a PHP date format (anything but < and >).
isPhoneNumber()	Is a phone number (digits, hyphens, spaces, parentheses and periods).
isPostCode()	Is a postcode (alphanumeric and hyphens).
isPrestaShopVersion()	Is a PrestaShop version (0 or 1 following by 3 one-or-two-digit numbers separated by periods).
isPrice()	Is a valid price (up to 10 digits, a period, then up to 9 digits).
isPriceDisplayMethod()	Is a price display method (`PS_TAX_EXC` or `PS_TAX_INC`).
isProductVisibility()	Is a product visibility (`both` , `catalog` , `search` or `none`)
isReductionType()	Is a reduction type "amount" or "percentage".
isReference()	Is a reference (^<>;={} not allowed).
isRoutePattern()	Is a friendly URL route pattern.
isSceneZones()	Is an array of image map zones (must have `x1` , `y1` , `width` , `height` and `id_product` keys with integer values).
isSerializedArray()	Is a serialised array.
isSha1()	Is a valid SHA1 hash.
isSiret()	Is a SIRET code.
isSortDirection()	Is a sort direction 'ASC' or 'DESC'.
isStateIsoCode()	Is a valid state ISO code (up to 4 lowercase alphanumeric).
isStockManagement()	Is a type of stock management (`WA` , `FIFO` or `LIFO`).
isString()	Is a string.
isSubDomainName()	Is a subdomain name.
isTablePrefix()	Is a table prefix (alphanumeric and underscore).
isTableOrIdentifier()	Is a table name (lowercase alphanumeric, hyphens and underscores).
isTabName()	Is a valid tab name (alphanumeric, hyphens and underscores).
isTagsList()	Is a tags list (^!<>;?=+#"Â°{}_$% not allowed).

isTplName()	Is a valid template name (lowercase alphanumeric and hyphens).
isTrackingNumber()	Is a shipping tracking number.
isUnixName()	Is a UNIX filename (non-Unicode alphanumeric, hyphens, underscores, backslashes and periods).
isUnsignedFloat()	Is a valid positive float value.
isUnsignedId()	Is a positive integer.
isUnsignedInt()	Is a positive integer.
isUrl()	Is a valid URL.
isUrlOrEmpty()	Is a valid URL or an empty string.
isUpc()	Is a 12-digit UPC code.
isValidSearch()	Is a valid search (up to 64 characters except ^<>;=#{}).
isValuesList()	**(deprecated)** Is a MySQL query values list.
isVoucherDescription()	Is a voucher description (<>{} not allowed except <br />).
isWeightUnit()	Is a weight unit (up to 3 letters).
isZipCodeFormat()	Is a zip code format (using uppercase letters N, L, C, spaces and hyphens).

Here's a list of useful static functions available in the Tools class:

Function Name	Description
addCSS	**(deprecated)** Adds a CSS file to by combined, compressed and cached.
addJS	**(deprecated)** Adds a JavaScript file to be combined, compressed and cached.
addonsRequest	Makes a request to the PrestaShop Addons website.
alignVersionNumber	Aligns two version numbers so they are easier to compare.
apacheModExists	Checks whether the specified Apache module exists.
argvToGET	Adds the specified arguments to the query string.
array_replace	Replaces values in an array.
arrayReplaceRecursive	Replaces values in an array recursively.
arrayUnique	Removes duplicate values from an array.
boolVal	Casts the value as a Boolean.
cccCss	**(deprecated)** Combines, compresses and caches the specified CSS files.
cccJS	**(deprecated)** Combines, compresses and caches the specified JavaScript files.
ceilf	Rounds up a float to the specified number of decimal places.
changeFileMTime	Changes the modification time of the specified file.
checkPhpVersion	Gets the PHP version.
chmodr	Recursively changes the permissions of the specified file or directory.
cleanNonUnicodeSupport	Deletes unicode from from regular expressions.
clearCache	Clears the cached Smarty templates in the `cache/smarty/cache` directory.

clearCompile	Clears the compiled Smarty templates in the `cache/smarty/compile` directory.
clearSmartyCache	Clears the compiled and cached Smarty templates.
clearXMLCache	Clears the cached XML files in the `config/xml` directory except for the installed themes.
completeMetaTags	**(deprecated)** Fills missing meta information with the default values.
convertBytes	Converts a shorthand byte to an integer.
convertPrice	Converts a price from one currency to another.
convertPriceFull	Converts a price from one currency to another and then rounds it.
copy	Copies a file or URL.
d	Displays an error message with a detailed object and stops execution.
dateDays	Gets a day array with the numbers 1 to 31.
dateFormat	Formats the specified date according to the current language's date format.
dateFrom	Adds 00:00:00 to the date if a time doesn't already exist.
dateMonths	Gets a month array with January to December.
dateTo	Adds 23:59:59 to the date if a time doesn't already exist.
dateYears	Gets a year array from the current year back to 1900.
debug_backtrace	Displays a stack trace for debugging.
deleteDirectory	Deletes a directory and its subdirectories.
deleteFile	Deletes a file if it isn't in the excluded array.
dieObject	Displays an error message with a detailed object.
dieOrLog	Displays an error message and/or logs it.
display404Error	**(deprecated)** Displays the "Page not found" page.
displayAsDeprecated	Displays a message that the called function is deprecated.
displayDate	Formats the specified date according to the current language's date format.
displayError	Displays the specified error message.
displayFileAsDeprecated	Displays a message that the called file is deprecated.
displayNumber	Changes the thousands separator of a number according to the specified currency.
displayParameterAsDeprecated	Displays a message that the specified parameter is deprecated.
displayPrice	Rounds and formats the price according to the specified currency.
displayPriceSmarty	Used to format the price from a Smarty template.
enableCache	Enables Smarty caching.
encrypt	Encrypts the specified password using the cookie key and MD5 hashes it.

encryptIV	Encrypts data using the cookie IV and MD5 hashes it.
error_log	Writes the specified object to the PHP error log.
fd	Displays the specified object in the Firebug console.
file_exists_cache	Checks whether the specified file exists with a cache.
file_exists_no_cache	Checks whether the specified file exists without the cache.
file_get_contents	Downloads the content from the specified URL.
fileAttachment	Gets information about file uploads.
floorf	Rounds down a float to the specified number of decimal places.
formatBytes	Shortens a number from 10,000 to 10k, for example.
generateHtaccess	Regenerates the .htaccess file in PrestaShop's root.
generateIndex	Regenerates `cache/class_index.php` .
getAdminToken	Gets the security token for a Back Office tab.
getAdminTokenLite	Gets the security token for the current employee.
getAdminTokenLiteSmarty	Gets the security token for the current employee from a Smarty template.
getAdminUrl	Makes a URL valid for use in the Back Office.
getAllValues	Gets all $_GET and $_POST variables in a single array.
getBrightness	Gets the brightness of a hex colour code.
getBytes	Gets random bytes of the specified length.
getCountry	Gets the customer's country from their entered address or IP address.
getCurrentUrlProtocolPrefix	Gets either `http://` or `https://` .
getDefaultIndexContent	Generates the content for an `index.php` file to block access to a directory.
getDescriptionClean	Removes all HTML tags from the specified description.
getFullPath	Get the breadcrumb bar for the specified category, product or CMS page.
getHomeMetaTags	**(deprecated)** Gets the meta-information for a specified page.
getHttpHost	Gets the current host with `http://` or `https://` .
getIsset	Checks whether the specified key exists in `$_POST` or `$_GET` .
getMaxUploadSize	Gets the maximum file upload size.
getMediaServer	Gets the media servers configured on the Advanced Parameters > Performance tab.
getMemoryLimit	Gets the memory limit as an octet.
getMetaTags	**(deprecated)** Gets the meta-information for the current product, category, manufacturer, supplier, cms page or cms category.
getOctets	Converts 10k to 1024, for example.

getPath	Gets part of the breadcrumb bar.
getProductsOrder	Gets the order by or way used to sort products.
getProtocol	Gets either `http://` or `https://` based on the "Use SSL" setting.
getRemoteAddr	Gets the IP address of the customer.
getSafeModeStatus	Gets whether PHP safe mode is enabled.
getServerName	The name of the current server.
getShopDomain	Gets the configured shop name when SSL is disabled.
getShopDomainSsl	Gets the configured shop name when SSL is enabled.
getShopProtocol	Gets `http://` or `https://` based on whether SSL is enabled on the server.
getToken	Gets the customer's secure token for the current page.
getUserBrowser	Gets the customer's web browser.
getUserPlatform	Gets the customer's operating system.
getValue	Gets the value from $_POST, $_GET or the specified default value.
hourGenerate	Converts values like 23 59 59 to 23:59:59.
htmlentitiesDecodeUTF8	Decodes HTML entities such as &lt; to <
htmlentitiesUTF8	Encodes HTML entities such as < to &lt;
iconv	Converts currency signs to HTML entities.
isCallable	Checks whether a function is enabled and can be called.
isEmpty	Checks whether a string is empty.
isPHPCLI	Checks whether PHP CLI is used.
isSubmit	Checks whether the specified form was submitted.
isX86_64arch	Checks whether the server is 64-bit.
jsonDecode	Decodes JSON as a data array.
jsonEncode	Encodes a data array as JSON.
link_rewrite	Converts a string to a friendly URL.
math_round	Rounds the number to the specified number of places using the specified rounding method.
minifyCSS	**(deprecated)** Minifies CSS styles.
minifyHTML	**(deprecated)** Minifies HTML content.
minifyHTMLprefCallback	**(deprecated)** Used when minifying HTML.
modRewriteActive	Checks whether mod_rewrite is active on the server.
nl2br	Converts new line characters to <br />
normalizeDirectory	Normalizes a directory.
orderbyPrice	Orders an array of product IDs by product price.
p	Displays an error with a detailed object, but doesn't stop execution.
packJS	**(deprecated)** Minifies JavaScript files.
packJSinHTML	**(deprecated)** Minifies JavaScript within HTML code.
packJSinHTMLpregCallback	**(deprecated)** Used when minifying JavaScript.
parserSQL	Parses an SQL query.

passwdGen	Generates a random password of the specified length.
pRegexp	Removes query strings and other characters from the end of URLs.
property_exists	**(deprecated)** Checks whether a class has the specified property.
ps_round	Rounds a number to the specified number of decimal places using the specified rounding method.
purifyHTML	Purifies HTML of unsafe code.
recurseCopy	Recursively copies a directory.
redirect	Redirects to a Front Office URL.
redirectAdmin	Redirects to a Back Office URL.
redirectLink	Redirects a URL that already includes the base URI.
replaceAccentedChars	Removes accents from characters.
replaceByAbsoluteURL	Replaces relative URLs in with absolute URLs.
restoreCacheSettings	Restores the cache settings.
round_helper	Used when rounding numbers.
rtrimString	Deletes the first instance of a string started from the end.
safeDefine	Defines a constant if it isn't already defined.
safeOutput	Converts HTML entities and optionally strips tags.
safePostVars	Converts HTML entities in `$_POST` .
scandir	Gets a list of all files in a directory.
secureReferrer	Makes the referrer URL secure.
setCookieLanguage	Changes the customer's language.
setCurrency	Changes the customer's currency.
simplexml_load_file	Loads an XML file.
spreadAmount	Spreads an amount over multiple lines.
str_replace_once	Replaces the first instance of a string in another string.
str2url	Converts a string to a friendly URL.
stripslashes	Unquotes backslashes in a string.
strlen	Gets the number of numbers in a string.
strpos	Gets the position of the first instance of a string in another string.
strReplaceFirst	Replaces the first instance of a string in another string.
strrpos	Gets the position of the last instance of a string in another string.
strtolower	Makes all letters in a string lowercase.
strtoupper	Makes all letters in a string uppercase.
substr	Gets part of a string.
switchLanguage	Changes the customer's language.
toCamelCase	Converts a string to camel case (first_name to firstName).
toUnderscoreCase	Converts a string to underscore case (firstName to first_name).

truncate	Truncates a string to the specified number of characters and adds a suffix.
truncateString	Truncates a string to the specified number of characters.
ucfirst	Makes the first letter uppercase.
ucwords	Makes the first letter of each word uppercase.
unSerialize	Unserializes a string.
url	Adds a query string variable to a URL.
usingSecureMode	Checks if the current page is using SSL.
version_compare	Compares version numbers.
waitUntilFileIsModified	Waits for a file to be modified for the specified number of seconds.
ZipExtract	Extracts a ZIP file to the specified directory.
ZipTest	Tests whether a ZIP file is valid.

Views

PrestaShop uses Smarty templating for its views, which supplements HTML with a flexible syntax to add variables, if statements, loops and other PHP code to the view. Here is a description of PrestaShop's default theme templates:

View Filename	Description
404.tpl	The "Page not found" page.
address.tpl	The address form.
addresses.tpl	Lists customer addresses.
authentication.tpl	The login and registration forms.
best-sales.tpl	Lists the best-selling products.
breadcrumb.tpl	The breadcrumb bar at the top of the centre column.
category-cms-tree.tpl	Used by sitemap.php to list CMS pages on the sitemap.
category-count.tpl	Displays the number of products in the current category.
category-tree-branch.tpl	Used by sitemap.php to list subcategories on the sitemap.
category.tpl	Lists a category's subcategories and products.
cms.tpl	Displays a CMS page.
contact-form.tpl	The customer service contact form.
discount.tpl	Lists vouchers available to a customer.
errors.tpl	Displays error messages.
footer.tpl	The footer at the bottom of the website.
global.tpl	Adds JavaScript to the footer of the page.
guest-tracking.tpl	The guest order tracking form.
header.tpl	The header at the top of the website.
history.tpl	Lists a customer's orders.
identity.tpl	The customer's personal information form.
index.tpl	The center column on the homepage.
layout.tpl	The main layout of the page.
maintenance.tpl	The maintenance page when the shop is disabled.
manufacturer.tpl	Lists the products of a manufacturer.
manufacturer-list.tpl	Lists manufacturers.
my-account.tpl	The account page of a customer.
new-products.tpl	Lists new products.

order-address.tpl	Used by order.php and order-opc.php for the address step.
order-address-advanced.tpl	Used by the Advanced EU Compliance module for the address step.
order-address-multishipping.tpl	Form to ship products to multiple addresses.
order-address-multishipping-products.tpl	Form to choose products for multishipping.
order-address-product-line.tpl	Displays a single product line on the multishipping form.
order-carrier.tpl	Used by order.php and order-opc.php for the carrier step.
order-carrier-advanced.tpl	Used by the Advanced EU Compliance module for the carrier step on five-step checkout.
order-carrier-opc-advanced.tpl	Used by the Advanced EU Compliance module for the carrier step on one-page checkout.
order-confirmation.tpl	The order confirmation page.
order-detail.tpl	Used by guest-tracking.php and history.php for order details.
order-follow.tpl	Lists a customer's merchandise returns.
order-opc.tpl	The one-page checkout page.
order-opc-advanced.tpl	Used by the Advanced EU Compliance module for one-page checkout.
order-opc-new-account.tpl	Used by order-opc.php for the login and new account forms.
order-opc-new-account-advanced.tpl	Used by the Advanced EU Compliance module for the new account forms.
order-payment.tpl	Used by order-opc.php for the payment step.
order-payment-advanced.tpl	Used by the Advanced EU Compliance module to display payment methods.
order-payment-classic.tpl	Displays payment methods.
order-return.tpl	Displays information about a merchandise return.
order-slip.tpl	Lists a customer's credit clips.
order-steps.tpl	Used by order.php to display the order steps.
pagination.tpl	The page navigation at the bottom of the centre column.
password.tpl	The lost password form.
prices-drop.tpl	Lists discounted products.
product-compare.tpl	Adds product comparison checkboxes to the product listings.
product-list.tpl	Lists products.
product-list-colors.tpl	Displays a colour picker on product listings.
product-sort.tpl	The "Sort by" form at the top of product listings.
product.tpl	Information about a product.
products-comparison.tpl	The product comparison page.
restricted-country.tpl	Displays a message when a customer has been restricted by country.
scenes.tpl	Used by category.php to display banners that link to products.

search.tpl	List the search results page.
shopping-cart.tpl	Used by order.php and order-opc.php to display the cart summary.
shopping-cart-advanced.tpl	Used by the Advanced EU Compliance module for the shopping cart.
shopping-cart-product-line.tpl	Used by shopping-cart.tpl to display a product line in the cart summary.
sitemap.tpl	The sitemap of the website.
store_infos.tpl	The store opening hours.
stores.tpl	The store information and Google map.
supplier-list.tpl	Lists suppliers.
supplier.tpl	Lists products from a supplier.

The following Smarty variables are accessible in all of the templates:

Smarty variable	Description
{$add_prod_display}	Whether to display the "Add to cart" button on products with attributes.
{$b2b_enable}	Whether business-to-business (B2B) mode is enabled.
{$base_dir}	The absolute base URL starting with `http://` .
{$base_dir_ssl}	The absolute base URL starting with `https://` .
{$base_uri}	The relative base URI without any protocol.
{$cart}	Used to access the customer's shopping cart.
{$cart_qties}	The number of products in the customer's shopping cart.
{$content_dir}	The content URL starting with `http://` .
{$content_only}	Whether to load the page without the header and footer (for Fancybox popups).
{$come_from}	The original URL if the customer was redirected.
{$compared_products}	The customer's chosen products for comparison.
{$comparator_max_item}	The maximum number of products that can be compared.
{$cookie}	Used to access the customer's cookie.
{$css_dir}	Absolute URL to the CSS directory `css` inside the current theme's directory.
{$currencies}	Used to access all currencies.
{$currency}	Used to access the customer's currency.
{$currencyBlank}	Whether the customer's currency has a blank space after the sign.
{$currencyFormat}	The customer's currency format.
{$currencyRate}	The conversion rate of the customer's currency.
{$currencySign}	The customer's currency sign.
{$customerName}	**(deprecated)** The customer's first and last names.
{$display_tax_label}	Whether to display tax included and excluded labels.
{$force_ssl}	Whether all URLs should use `https://` .
{$hide_left_column}	Whether the left column should be hidden.
{$hide_right_column}	Whether the right column should be hidden.
{$id_currency_cookie}	**(deprecated)** The ID of the customer's currency.

{$img_cat_dir}	The absolute URL to the category image directory `img/c` in PrestaShop's root.
{$img_col_dir}	The absolute URL to the attribute texture directory `img/co` in PrestaShop's root.
{$img_dir}	The absolute URL to the image directory `img` directory inside the current theme.
{$img_lang_dir}	The absolute URL to the language flag directory `img/l` in PrestaShop's root.
{$img_manu_dir}	The absolute URL to the manufacturer image directory `img/m` in PrestaShop's root.
{$img_prod_dir}	The absolute URL to the product image directory `img/p` in PrestaShop's root.
{$img_ps_dir}	The absolute URL to the image directory `img` in PrestaShop's root.
{$img_ship_dir}	The absolute URL to the carrier image directory `img/s` in PrestaShop's root.
{$img_store_dir}	The absolute URL to the store image directory `img/st`
{$img_sup_dir}	The absolute URL to the supplier image directory `img/su` in PrestaShop's root.
{$img_update_time}	The time the shop logo was last updated.
{$is_guest}	Whether the customer is a guest.
{$is_logged}	Whether the customer is logged in.
{$js_dir}	Absolute URL to the JavaScript directory `js` inside the current theme's directory.
{$lang_id}	The ID of the customer's language (`1`).
{$lang_iso}	The ISO code of the customer's language (`en`).
{$language_code}	The language code of the customer's language (`en-us`).
{$languages}	Used to access all languages.
{$link}	Used to generate links to various pages.
{$logged}	**(deprecated)** Whether the customer is logged in.
{$mail_dir}	The relative URL to the mails directory.
{$meta_language}	The ISO codes of all the active languages.
{$mobile_device}	Whether the website is being viewed on a phone.
{$modules_dir}	The relative URL to the modules directory.
{$opc}	Whether one-page checkout is enabled.
{$page_name}	The name of the current page (as displayed in the <body> tag).
{$pic_dir}	Absolute URL to the customized image directory `upload` inside PrestaShop's root.
{$priceDisplay}	Whether to display prices with or without tax.
{$priceDisplayPrecision}	The number of decimal places to use in prices.
{$PS_CATALOG_MODE}	Whether catalog mode is enabled.
{$PS_STOCK_MANAGEMENT}	Whether stock management is enabled.
{$quick_view}	Whether quick view is enabled.
{$request}	The pagination link for the current page.
{$roundMode}	The method of rounding prices.

{$shop_name}	The shop's name.
{$shop_phone}	The shop's phone number.
{$static_token}	The static security token for the customer.
{$show_taxes}	Whether to show taxes on a separate line.
{$time}	The current time in milliseconds since January 1, 1970 00:00:00 GMT.
{$token}	The customer's security token for the current page.
{$tpl_dir}	The absolute URL to the current theme's directory.
{$tpl_uri}	The relative URL to the current theme's directory without any protocol.
{$use_taxes}	Whether to add taxes to prices.
{$vat_management}	Whether the European VAT Number module is enabled.

Controllers

Here's a list of PrestaShop's controllers and a description of what each of them do:

Controller Name	Description
AddressController.php	Used to edit a customer's address.
AddressesController.php	Used to get a customer's addresses.
AttachmentController.php	Used to download product attachments.
AuthController.php	Used to manage customer login and registration.
BestSalesController.php	Used to get the best-selling products.
CartController.php	Used to manage the customer's cart.
CategoryController.php	Used to get product categories.
ChangeCurrencyController.php	Used to change the currency of prices.
CmsController.php	Used to get a CMS page.
CompareController.php	Used to compare products.
ContactController.php	Used to send messages to employees.
DiscountController.php	Used to get vouchers available to a customer.
GuestTrackingController.php	Used to manage guest orders.
HistoryController.php	Used to get a customer's orders.
IdentityController.php	Used to get a customer's personal information.
IndexController.php	Used to display the homepage.
ManufacturerController.php	Used to get manufacturers and their products.
MyAccountController.php	Used to display a customer's account page.
NewProductsController.php	Used to get a list of new products.
OrderConfirmationController.php	Used to manage the order confirmation page.
OrderController.php	Used to manage the five-step checkout pages.
OrderDetailController.php	Used to get a single customer order.
OrderFollowController.php	Used to get a list of a customer's merchandise returns.
OrderOpcController.php	Used to manage the one-page checkout page.
OrderReturnController.php	Used to get a single merchandise return.
OrderSlipController.php	Used to get a list of a customer's credit slips.
PageNotFoundController.php	Used to manage the "Page not found" page.
ParentOrderController.php	Used for shared ordering code.
PasswordController.php	Used to allow a customer to reset a lost password.

PdfInvoiceController.php	Used to download PDF invoices.
PdfOrderReturnController.php	Used to download PDF merchandise returns.
PdfOrderSlipController.php	Used to download PDF credit slips.
PricesDropController.php	Used to get a list of discounted products.
ProductController.php	Used to get a product.
SearchController.php	Used to perform a search and get the results.
SitemapController.php	Used to get the sitemap of the website.
StatisticsController.php	Used to log OS and browser stats and page load time.
StoresController.php	Used to get store information and Google map coordinates.
SupplierController.php	Used to get suppliers and their products.

All controllers in PrestaShop extend the FrontController class. Here's a description of the FrontController members:

Member Name	Description
$auth	Whether a customer must log in before they can access the page.
$authRedirection	The page redirected to after logging in.
$cart	**(deprecated)** The current customer's or guest's cart.
$cookie	**(deprecated)** Used to access PrestaShop's cookie variables.
$currentCustomerGroups	Caches the current customer's groups.
$display_column_left	Whether to display the left column on the page.
$display_column_right	Whether to display the right column on the page.
$errors	An array of errors that have occurred.
$guestAllowed	Whether a customer who has signed out can access the page.
$initialized	Whether the init() function has been called.
$iso	The iso code of the currently selected language.
$link	**(deprecated)** Used to generate PrestaShop links.
$maintenance	Forces the maintenance page to be displayed.
$n	The number of items per page.
$nb_items_per_page	The number of items per page.
$orderBy	The field used to sort.
$orderWay	Whether to sort ascending or descending ("ASC" or "DESC").
$p	The current page number.
$restrictedCountry	Forces the restricted country page to be displayed.
$smarty	**(deprecated)** Used to access Smarty variables and functions.
$ssl	Whether the page is loaded using SSL when it is available.

Here's a description of the controller functions in the order they are executed:

Function Name	Description
__construct()	Sets all the controller's member variables.
setMedia()	Adds all JavaScript and CSS specific to the page so it can be combined, compressed and cached.

initHeader()	Initialises the header variables.
initContent()	Initialises the content variables.
initFooter()	Initialises the footer variables.
display()	Calls the Smarty templates and displays the page.

PRODUCTS

Our Modules

Merchant Guide

Developer Guide

OUR COMPANY

Terms & Conditions

Donate

Contact us

My account

ABOUT NETHERCOTT CONSTRUCTIONS

We are the Australian PrestaShop Ambassador and a Superhero Seller. We have been regularly providing advice and support through the PrestaShop forums to many merchants and developers since 2009.

We are available for custom themes and modules, easy to install and highly configurable modules that improve your online shop, and advice on PrestaShop setup and performance optimisation.

© 2009-2018 Nethercott Constructions