Developer tips and tricks

Table of contents

Developer tips and tricks

How to check whether a user is logged to PrestaShop or not

You can easily perform this check by using the `isLoggedBack()` method provided in the `Employee` class (`/classes/Employee.php`).

This example assumes your file is located in the administration folder:

```
<?php

define('_PS_ADMIN_DIR_', getcwd());
include(_PS_ADMIN_DIR_.'/../config/config.inc.php');

if (Context::getContext()->employee->isLoggedBack())
{
   /* Your code here, user is logged */
}

?>
```

The `isLoggedBack()` method was located in the `Cookie` class in PrestaShop 1.4.x, and was moved to `Employee` class in v1.5.x.

How to find the speed bottleneck on a site

Try to enable the profiling feature, it will highlight which part of the code is slowing down your server.

1. Put your store in maintenance mode.
2. In your PrestaShop v1.5.x store, edit the `/config/defines.inc.php` file.
3. On line 44, replace `define('_PS_DEBUG_PROFILING_', false);` by `define('_PS_DEBUG_PROFILING_', true);`
4. Go to your store homepage and reload the page.

You should now be able to see how much time is consuming each function and SQL query.

How to retrieve the carrier's id

In PrestaShop 1.4, you could use `{id_carrier}`:

```
{if $id_carrier == "1"}
  {* Do some stuff here *}
{/if}
```

The same result can be achieved in PrestaShop 1.5+ with `{$cart->id_carrier}`:

```
{if isset($cart->id_carrier) && $cart->id_carrier == 1}
   {* Do some stuff here *}
{/if}
```

How to retrieve the cart's id

You have several ways to retrieve the Cart ID from the current visitor, the easiest one is to use the Context.

Step 1

Open the `/modules/blockcart/ajax-cart.js` file and look for this line:

```
$(document).ready(function(){
```

Below this line, add this:

```
$.ajax({
    type: 'GET',
    url: baseDir + 'modules/blockcart/ajax.php' + '?retrieve_cart_id=1',
    success: function(result_cart_id)
    {
        alert(result_cart_id);
        /* my_id_cart = parseInt(result_cart_id); Uncomment this line to store the value into a JS variable */
    }
});
```

Step 2

Create a file named `ajax.php` in `/modules/blockcart/` with the following code inside:

```
<?php

include(dirname(__FILE__).'/../../config/config.inc.php');
include(dirname(__FILE__).'/../../init.php');

$context = Context::getContext();
if (Tools::getValue('retrieve_cart_id') == 1)
    echo isset($context->cookie->id_cart) ? (int)$context->cookie->id_cart : 0;
```

You might want to consider the fact that by retrieving this value in JavaScript, it will be publicly available to the visitor. Depending on how secure your code and payment modules are, this could be an issue.

How to import products from a XML file

With PrestaShop you can easily import your data using one of these options:

1. By using the "CSV Import" feature in your Back-office
2. By using the PrestaShop Web-service (custom development required)
3. By using the existing classes (custom development required)

Below is a quick code snippet that is working with a specific XML document (stored as a heredoc string) . It will create or update all the products, and take into account their price, availability, name, description, weight, etc.

Please note that:

- This code MUST be adapted to your own XML needs!
- This code will is not designed to work with combinations (color, sizes, etc.)
- This code does not create categories for you, and is adding all the products to the "Home" category

```php
<?php

include(dirname(__FILE__).'/config/config.inc.php');
include(dirname(__FILE__).'/init.php');

$xml_string = <<<XML
<?xml version="1.0" encoding="UTF-8"?>
<Document>
  <Products>
    <Reference>1101TEST</Reference>
    <Valid_internet_product>1</Valid_internet_product>
    <Products_name>Test product</Products_name>
    <Price>49.99</Price>
    <Active_product>1</Active_product>
    <SupplierNo>8</SupplierNo>
    <Weight>5</Weight>
    <Description>My long product description</Description>
    <Short_Description>Product desc.</Short_Description>
    <MinOrderQty>1</MinOrderQty>
    <Categories>
      <Category>
        <CategoryID>3</CategoryID>
          <CategoryName>Home\Prod</CategoryName>
          <Active_category>1</Active_category>
          <Changed>0</Changed>
      </Category>
    </Categories>
    <Tax_Class_ID>1</Tax_Class_ID>
    <Discount>
      <Discount_percentage>percentage</Discount_percentage>
      <discountprice_ex_vat>0</discountprice_ex_vat>
      <Discountprice_include_vat>0</Discountprice_include_vat>
      <Pct_ReductionPercent>0</Pct_ReductionPercent>
    </Discount>
  </Products>
</Document>
XML;

$xml = simplexml_load_string($xml_string);
foreach ($xml->Products as $product_xml)
{
    if ($product_xml->Valid_internet_product == 1)
    {
        /* Update an existing product or Create a new one */
        $id_product = (int)Db::getInstance()->getValue('SELECT id_product FROM '._DB_PREFIX_.'product WHERE
reference = \''.pSQL($product_xml->Reference).'\'');
        $product = $id_product ? new Product((int)$id_product, true) : new Product();
        $product->reference = $product_xml->Reference;
        $product->price = (float)$product_xml->Price;
        $product->active = (int)$product_xml->Active_product;
        $product->weight = (float)$product_xml->Weight;
        $product->minimal_quantity = (int)$product_xml->MinOrderQty;
        $product->id_category_default = 2;
        $product->name[1] = utf8_encode($product_xml->Products_name);
        $product->description[1] = utf8_encode($product_xml->Description);
        $product->description_short[1] = utf8_encode($product_xml->Short_Description);
        $product->link_rewrite[1] = Tools::link_rewrite($product_xml->Products_name);
        if (!isset($product->date_add) || empty($product->date_add))
            $product->date_add = date('Y-m-d H:i:s');
        $product->date_upd = date('Y-m-d H:i:s');
        $id_product ? $product->updateCategories(array(2)) : $product->addToCategories(array(2));
        $product->save();

        echo 'Product <b>'.$product->name[1].'</b> '.($id_product ? 'updated' : 'created').'<br />';
    }
}
```