Synchronization via Hooks

Synchronization via Hooks (eBay)

This article was written by Fabien Serny, and was fist published on June 22nd, 2011, on the PrestaShop blog.

Introduction

To learn more about hooks, I advise you to read Julien Breux's article here: Understanding and using hooks

Sometimes we need to set up an automated task (also called "cron job"), especially for certain modules that synchronize their product catalog to a marketplace.

Let's take a look at the eBay module as an example. The simplest solution would have been to set up an automated task regularly resynchronizing its product catalog with the marketplace. However, some problems may arise with this solution:

- If you run the task daily, then inventory synchronization will not be in real time. This can become complicated for merchants who have many customers per day and a low quantity of certain products.
- If you run the task every 30 minutes, it quickly becomes resource intensive, especially if merchants have over 5000 products.

The best solution to solve this problem is to use hooks.

How do I use hooks?

If we study the problem, we realize we can simply synchronize the global catalog when installing the module, and then resynchronize only the products that are likely to have been changed.

In the eBay module, a button exists for the initial synchronization. Subsequent synchronizations occur only when a product is added, modified, or removed, or its quantity decreases following an order.

The hooks that can be used are:

Hook Name	Description
Addproduct	adds a product
Updateproduct	Updates the product
updateProductAttribute	Updates a product's attribute
deleteproduct	Deletes a product
newOrder	for a new order

Let's look at, for example, the `addproduct` hook, the `$params` parameter that contains the `Product` object, which will allow us to easily find the product has updated.

```
public function hookaddproduct($params)
{
    // We check that the product identification is present
    if (!isset($params['product']->id))
        return false;

    // And we check that this one is also valid
    $id_product = $params['product']->id;
    if ((int)$id_product < 1)
        return false;

    // Now we just have to launch the product synchronization on eBay
    if ($this->_syncProducts(array('id_product' => (int)$id_product)))
        return true;

    return false;
}
```

Since the `hookupdateproduct`, `hookUpdateProductAttribute`, and `hookdeleteproduct` methods are all the same, you just need to add a few lines:

```
public function hookupdateproduct($params) { $this>hookaddproduct($params);  }
public function hookupdateProductAttribute($params) { $this->hookaddproduct($params); }
public function hookdeleteproduct($params) { $this->hookaddproduct($params); }
```

With this method, inventories are synchronized in real time, and products that do not need resynchronization are excluded from the process.

Please note: For those interested in the operation of `hooknewOrder` in this module, I suggest you go and look directly at the eBay module's code 😋

And what about actions without a hook?

For actions without hooks (eg. importing eBay orders to PrestaShop), it is better to use an automated task. However, this is not always an option (if you're on a shared server for example).

There is still a little trick that simulates an automated task using your visitors. Just hook the module to the front (preferably with a hook called 'appeal to all pages.'), and then simulate an automated task inside.

```
public function hookheader($params)
{
    // We check that the module is configured correctly
    if (!Configuration::get('EBAY_CONFIGURED')
        return false;

    // If the orders were never imported
    // We fill in the last import date with today's date
    if (!Configuration::get('EBAY_ORDER_LAST_UPDATE'))
        Configuration::updateValue('EBAY_ORDER_LAST_UPDATE',
        date('Y-m-d').'T'.date('H:i:s').'.000Z');

    // We check that it's been 30 minutes since the last import
    $dateNow = date('Y-m-d').'T'.date('H:i:s').'.000Z';
    $date30mn = date('Y-m-d', strtotime('-45 minutes')).'T'.
            date('H:i:s', strtotime('-45 minutes')).'.000Z');
    if (Configuration::get('EBAY_ORDER_LAST_UPDATE') < $date30mn)
    {
        // We retrieve the orders from eBay
        $ebay = new eBayRequest();
        $orderList = $ebay->getOrders(
                Configuration::get('EBAY_ORDER_LAST_UPDATE'),
                $dateNow);

        // We put them into the database
        // ...

        // We update the last import date
        Configuration::updateValue('EBAY_ORDER_LAST_UPDATE', $dateNew);

        return true;
    }

    return false;
}
```

The only issue is that if you do not have traffic on your site, the task is never executed.

Please note: You must also make sure that the task doesn't take too long; otherwise the client that has triggered the hook will be negatively affected.