

La synchronisation via les Hooks

La synchronisation via les Hooks (eBay)

Cet article a été écrit par Fabien Serny, et [publié sur le blog de PrestaShop le 22 juin 2011.](#)

Préambule

Pour savoir ce qu'est un hook, je vous conseille de vous reporter à l'article de Julien Breux ici : [Mieux comprendre et utiliser les hooks](#)

Il est parfois nécessaire de mettre en place une tâche automatisée (appelée également tâche cron) notamment pour certains modules permettant de synchroniser son catalogue produit avec une marketplace.

Nous allons prendre le cas du module eBay. La solution de simplicité aurait été de mettre en place une tâche automatisée permettant de resynchroniser régulièrement son catalogue produits avec la place de marché. Cependant certains problèmes peuvent se poser avec cette solution :

- Si l'on met une tâche journalière, la synchronisation des stocks ne sera pas en temps réel, ce qui peut devenir compliqué pour un marchand ayant beaucoup de clients par jour et peu de quantité pour certains produits.
- Si l'on exécute la tâche toutes les 30mn, cela devient vite gourmand en ressource notamment si le nombre de produits dépassent les 5000. La meilleure solution pour palier à ce problème est l'utilisation des Hooks.

Comment faire ?

Si l'on étudie le problème, on se rend compte qu'il suffit de faire une synchronisation globale du catalogue à l'installation du module, puis de ne resynchroniser que les produits qui sont susceptibles d'avoir été modifiés.

Dans le cas du module eBay, un bouton permettant une première synchronisation est présent. Les synchronisations suivantes n'ont lieu que lorsqu'un produit est ajouté, modifié, effacé ou que sa quantité a diminué (passage d'une commande de ce produit).

Les hooks à utiliser sont donc les suivants :

Nom du hook	Description
addproduct	« hook » d'action « Ajout de produit »
updateproduct	« hook » d'action « Mise à jour de produit »
updateProductAttribute	« hook » d'action « Mise à jour d'attribut produit »
deleteproduct	« hook » d'action « Effacement d'un produit »
newOrder	« hook » d'action « Nouvelle commande »

Prenons l'exemple du hook addproduct, le paramètre \$params contient l'objet Product, ce qui va nous permettre de trouver facilement le produit à actualiser.

```

public function hookaddproduct($params)
{
    // On vérifie que l'identifiant produit est bien présent
    if (!isset($params['product']->id))
        return false;

    // Et on vérifie que celui-ci est bien valide
    $id_product = $params['product']->id;
    if ((int)$id_product < 1)
        return false;

    // Il n'y a plus qu'à lancer la synchronisation du produit sur eBay
    if ($this->_syncProducts(array('id_product' => (int)$id_product)))
        return true;

    return false;
}

```

Les méthodes `hookupdateproduct`, `hookUpdateProductAttribute`, `hookdeleteproduct` étant les mêmes, il suffira de rajouter ces quelques lignes :

```

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

```

En utilisant cette méthode, les stocks sont synchronisés en temps réel et cela évite de resynchroniser des produits qui n'en ont pas besoin.

NB : Pour les intéressés par le fonctionnement du `hooknewOrder` dans ce module, je vous laisse aller regarder directement dans le code du module eBay 🍷

Et dans le cas des actions sans hook ?

Dans le cas des actions sans hook (par exemple l'import de commandes eBay vers PrestaShop), il faut mieux utiliser une tâche automatisée. Mais il arrive que vous n'en ayez pas la possibilité (si vous êtes sur un serveur mutualisé par exemple).

Il existe tout de même une petite astuce qui permet de simuler une tâche automatisée à l'aide de vos visiteurs. Il vous suffit de hooker le module sur le front (de préférence un hook appelé à toutes les pages. Vous pouvez ensuite simuler une tâche automatisée à l'intérieur.

```

public function hookheader($params)
{
    // On vérifie que le module est bien configuré
    if (!Configuration::get('EBAY_CONFIGURED'))
        return false;

    // Dans le cas où les commandes n'ont jamais été importés
    // On remplit la dernière date d'import à la date du jour
    if (!Configuration::get('EBAY_ORDER_LAST_UPDATE'))
        Configuration::updateValue('EBAY_ORDER_LAST_UPDATE',
            date('Y-m-d').'T'.date('H:i:s').'000Z');

    // On vérifie qu'il s'est bien écoulé 30mn depuis le dernier 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)
    {
        // On récupère les commandes auprès d'eBay
        $ebay = new eBayRequest();
        $orderList = $ebay->getOrders(
            Configuration::get('EBAY_ORDER_LAST_UPDATE'),
            $dateNow);

        // On les insère en base de données
        // ...

        // On met à jour la dernière date d'import
        Configuration::updateValue('EBAY_ORDER_LAST_UPDATE', $dateNew);

        return true;
    }

    return false;
}

```

Seule limite : si vous n'avez pas de trafic sur votre site, la tâche n'est jamais exécutée.

NB : Il faut également faire attention à ce que la tâche ne prenne pas trop de temps, sinon c'est le client qui aura déclenché le hook qui va en pâtir.

- « Ce module eBay, intégré en natif dans PrestaShop et sur Addons, a été développé dans le cadre et dans l'esprit du partenariat eBay-PrestaShop : créer des services qui permettent aux marchands de développer leur activité e-commerce. En effet, le module eBay permet aux marchands de mettre en ligne facilement leurs produits sur eBay.fr et ainsi de bénéficier de ses 12 millions de visiteurs uniques/mois afin de conquérir de nouveaux clients et de gagner en visibilité. »
Lucile CHANEL, Partnership Manager eBay France