

Импорт товаров из CSV-файла с идентификацией по артикулу и производителю

При импорте товаров из CSV-файла учитывается только артикул товара, и несколько товаров с одинаковым артикулом будут импортированы, как один товар. Артикул может совпадать у товар разных производителей, а у одного производителя все артикулы обычно уникальны, поэтому можно внести модификации в импорт для того, чтобы в идентификации товара участвовал не только артикул, но и производитель товара.

Для реализации такого подхода можно воспользоваться нижеприведенным решением, оно позволяет импортировать товары с одинаковым артикулом и разными производителями, при этом в импортируемом CSV-файле *первым должен быть столбец с производителем, а вторым столбец с артикулом*.

Устанановка решения:

1. Создать файл наблюдателя:

```
<?php

/*
 * B bootstrap.php:
 *
 // Поиск по артикулу и производителю при импорте CSV-файла
 Core_Event::attach('ImportShopItems.onBeforeFindByMarking',
 array('Shop_Item_Import_Observer_SearchByProdAndVendor',
 'onBeforeFindByMarking'));
 Core_Event::attach('ImportShopItems.onAfterFindByMarking',
 array('Shop_Item_Import_Observer_SearchByProdAndVendor',
 'onAfterFindByMarking'));
 *
 */

defined('HOSTCMS') || exit('HostCMS: access denied.');
```

```
class Shop_Item_Import_Observer_SearchByProdAndVendor
{
    static public function onBeforeFindByMarking($object, $args)
    {
        // Запоминаем товар с производителем
        Core_Registry::instance()->set('shop_item_with_prod', clone
$args[1]);
    }

    static public function onAfterFindByMarking($object, $args)
    {
        // Новый товар
```

```

        $oItemWithProd =
Core_Registry::instance()->get('shop_item_with_prod');

        // Результат поиска по артикулу
        $oItemWithMarking = $args[1];

        $oShop_Item = Core_Entity::factory('shop_item');
        $oShop_Item->queryBuilder()
            ->where('marking', '=', $oItemWithMarking->marking)
            ->where('shop_producer_id', '=',
$oItemWithProd->shop_producer_id)
            ->limit(1)
        ;
        $aoShop_Item = $oShop_Item->findAll();

        if (count($aoShop_Item) > )
        {
            $object->setCurrentItem($aoShop_Item[]);
        }
        else
        {
            $oItemWithProd->marking = $oItemWithMarking->marking;
            $object->setCurrentItem($oItemWithProd);
        }
    }
}

```

1. Зарегистрировать в файле bootstrap.php следующие события:

```

// Поиск по артикулу и производителю при импорте CSV-файла
Core_Event::attach('Shop_Item_Import_Csv_Controller.onBeforeFindByMarking', array('Shop_Item_Import_Observer_SearchByProdAndVendor', 'onBeforeFindByMarking'));
Core_Event::attach('Shop_Item_Import_Csv_Controller.onAfterFindByMarking', array('Shop_Item_Import_Observer_SearchByProdAndVendor', 'onAfterFindByMarking'));

```