

Скрыть товары и модификации товаров, которых нет на основном складе

Товары

Хотим скрыть товары, которых нет в наличии на текущем складе. Назначаем наблюдатель для события `onBeforeRedeclaredShow` класса `Shop_Controller_Show`:

- В `bootstrap.php` добавляем код:

```
// Идентификатор интернет-магазина
define('REGION_PRICES_SHOP_ID', 4);

// KAD: Скрываем товары, которых нет в наличии на текущем складе
Core_Event::attach('Shop_Controller_Show.onBeforeRedeclaredShow',
array('Kad_Shop_Item_Observers_Hidenotavailableitems',
'onBeforeRedeclaredShow'));
```

- В директории `/modules/kad/shop/item/observers/` создаем файл `hidenotavailableitems.php` со следующим содержимым:

```
<?php

/*
 * @author Maxim Zasorin, KAD Systems (©) 2015
 * @date 18-11-2015
 *
 * В bootstrap.php:
 *
 // KAD: Скрываем товары, которых нет в наличии на текущем складе
Core_Event::attach('Shop_Controller_Show.onBeforeRedeclaredShow',
array('Kad_Shop_Item_Observers_Hidenotavailableitems',
'onBeforeRedeclaredShow'));

* Копирование и использование файлов модуля
* в коммерческих целях ЗАПРЕЩЕНО
*
*/
defined('HOSTCMS') || exit('HostCMS: access denied.');
```

```
class Kad_Shop_Item_Observers_Hidenotavailableitems
{
    static public function onBeforeRedeclaredShow($object, $args)
    {
        if (
            (
                defined('REGION_PRICES_SHOP_ID')
                &&
```

```

        REGION_PRICES_SHOP_ID == $object->getEntity()->id
    )
    &&
    !$object->item
)
{
    $oShop = Core_Entity::factory('Shop',
REGION_PRICES_SHOP_ID);

    $oCurrent_Warehouse =
$oShop->Shop_Warehouses->getDefault();

    // Вариант 1. Некорректный, но менее затратный вариант, в этом случае
ярлыки товаров не будут
    // проверяться на наличие на текущем складе, а будут выводится все
время
    $object->shopItems()
        ->queryBuilder()
        ->leftJoin('shop_warehouse_items',
'shop_warehouse_items.shop_item_id', '=', 'shop_items.id')
        ->open()
        ->where('shop_warehouse_items.count', '>', )
        ->where('shop_warehouse_items.shop_warehouse_id',
'=', $oCurrentWarehouse->id)
        ->setOr()
        ->where('shop_items.shortcut_id', '!=', )
        ->close()
    ;

    /// Вариант 2. Корректный, с LEFT JOIN, но более затратный, в этом
случае все ярлыки будут корректно
    /// обрабатываться условием наличия на складе. Данный вариант
предусматривает сложное объединение таблиц через LEFT JOIN ... OR
    /// и может выполняется значительное время, особенно на объемных
интернет-магазинах
    // $object->shopItems()
    //     ->queryBuilder()
    //     ->leftJoin('shop_warehouse_items',
'shop_warehouse_items.shop_item_id', '=', 'shop_items.id', array(
    //         array('OR' =>
array('shop_warehouse_items.shop_item_id', '=',
Core_QueryBuilder::expression(`shop_items`.`shortcut_id`))),
    //     )
    //     )
    //     ->where('shop_warehouse_items.count', '>', 0)
    //     ->where('AND' =>
array('shop_warehouse_items.shop_warehouse_id', '=',
$oCurrentWarehouse->id))
    //     ;

    // var_dump($object->shopItems()->queryBuilder()->build());

```

```
}  
}  
}
```

- Решение предусматривает обработку ярлыков, но с ними могут быть проблемы в интернет-магазинах с большим количеством товаров, из-за чего в наблюдателе предусмотрено два варианта выборки, один из них отбрасывает ярлыки при сравнении, а другой корректно обрабатывает их, но может выполняться значительно время (см. комментарии в наблюдателе).

Модификации

Хотим скрыть модификации товаров, которых нет в наличии на текущем складе. Назначаем наблюдатель для события `onBeforeShowXmlModifications` сущности `shop_item`:

- В `bootstrap.php` добавляем код:

```
// KAD: Скрываем товары, которых нет в наличии на текущем складе  
Core_Event::attach('shop_item.onBeforeShowXmlModifications',  
array('Kad_Shop_Item_Observers_Hidenotavailablemods',  
'onBeforeShowXmlModifications'));
```

- В директории `/modules/kad/shop/item/observers/` создаем файл `hidenotavailablemods.php` со следующим содержимым:

```
<?php  
  
/*  
 * @author Maxim Zasorin, KAD Systems (©) 2015  
 * @date 18-11-2015  
 *  
 * В bootstrap.php:  
 *  
 // KAD: Скрываем модификации, которых нет в наличии на текущем складе  
 Core_Event::attach('shop_item.onBeforeShowXmlModifications',  
 array('Kad_Shop_Item_Observers_Hidenotavailablemods',  
 'onBeforeShowXmlModifications'));  
  
 * Копирование и использование файлов модуля  
 * в коммерческих целях ЗАПРЕЩЕНО  
 *  
 */  
defined('HOSTCMS') || exit('HostCMS: access denied.');
```

```
class Kad_Shop_Item_Observers_Hidenotavailablemods  
{  
    static public function onBeforeShowXmlModifications($object, $args)  
    {  
        if (  
            (  

```

```

        defined('REGION_PRICES_SHOP_ID')
        &&
        REGION_PRICES_SHOP_ID == $object->shop_id
    )
    )
    {
        $oShop = Core_Entity::factory('Shop',
REGION_PRICES_SHOP_ID);

        $oCurrent_Warehouse =
$oShop->Shop_Warehouses->getDefault();

        $args[]->queryBuilder()
            ->select('shop_items.*')
            ->leftJoin('shop_warehouse_items',
'shop_warehouse_items.shop_item_id', '=', 'shop_items.id', array(
                array(
                    'AND' =>
array('shop_warehouse_items.shop_warehouse_id', '=',
$oCurrent_Warehouse->id)
                )
            )
        )
        ->where('shop_warehouse_items.count', '>', );
    }
}
}

```

Константа REGION_PRICES_SHOP_ID задает идентификатор интернет-магазина, для которого хотим скрыть товары и модификации товаров.

Если наблюдатель не срабатывает на странице интернет-магазина, то возможно в ТДС используется класс, наследованный от Shop_Controller_Show, например, с названием My_Shop_Controller_Show, или другой. Для такого класса нужно назначить наблюдатель отдельно. В bootstrap.php добавляем код:

```

Core_Event::attach('My_Shop_Controller_Show.onBeforeRedeclaredShow',
array('Kad_Shop_Item_Observers_Hidenotavailableitems',
'onBeforeRedeclaredShow'));

```