

```

<?php

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

*/**
 * Наблюдатель для работы с наценкой товара.
 /

```

class Shop_Order_Item_Observer_AdditionalFields
{
    /**  

     * Метод обратного вызова для кастомного поля shop_order_item.markup  

     *  

     * @param Shop_Order_Item_Model $oShopOrderItem  

     * @param array $aArgs  

     * @return string  

     */
    static public function onCallmarkup(Shop_Order_Item_Model
$oShopOrderItem, array $aArgs)
    {
        $markup = self::_getMarkup($oShopOrderItem);

        if ($markup)
        {
            return $markup . ' ' .
$oShopOrderItem->Shop_Order->Shop->Shop_Currency->name;
        }
        else
        {
            return '-';
        }
    }

    /**  

     * Метод обратного вызова для кастомного поля  

     shop_order_item.price_without_markup  

     *  

     * @param Shop_Order_Item_Model $oShopOrderItem  

     * @param array $aArgs  

     * @return string  

     */
    static public function onCallprice_without_markup(Shop_Order_Item_Model
$oShopOrderItem, array $aArgs)
    {
        return $oShopOrderItem->price - self::_getMarkup($oShopOrderItem);
    }

    /**  

     * Добавляет наценку товара в заказе  

     *  

     * @param Shop_Order_Item_Model $oShopOrderItem  

     * @param array $aArgs

```

```

    * @return void
    */
    static public function onBeforeRedeclaredGetXml(Shop_Order_Item_Model
$ShopOrderItem, array $aArgs)
    {
        $ShopOrderItem
            ->addEntity(
                Core::factory('Core_Xml_Entity')
                    ->name('markup')
                    ->value(self::_getMarkup($ShopOrderItem))
            );
    }

    /**
    * Возвращает наценку для товара в заказе
    *
    * @param Shop_Order_Item_Model $ShopOrderItem
    * @return number
    */
    static protected function _getMarkup(Shop_Order_Item_Model
$ShopOrderItem)
    {
        if (!$ShopOrderItem->shop_item_id)
        {
            return ;
        }

        $ShopOrder = $ShopOrderItem->Shop_Order;

        $ShopOrderPropertyList =
Core_Entity::factory('Shop_Order_Property_List', $ShopOrder->Shop->id);
        $oProperty =
$ShopOrderPropertyList->Properties->getByTagName('markups');

        if (!$oProperty)
        {
            return ;
        }

        $aoPropertyValues = $ShopOrder->getPropertyValues(TRUE,
array($oProperty->id));

        if (!count($aoPropertyValues))
        {
            return ;
        }

        $aMakups = @json_decode($aoPropertyValues[]->value, TRUE);

        if (isset($aMakups[$ShopOrderItem->shop_item_id]))
        {

```

```
        return $aMakups[$oShopOrderItem->shop_item_id];  
    }  
  
    return ;  
}  
}
```

Страницу создал Максим Засорин 23.01.17 в 14:43