

Расчет веса товаров в корзине

Данное решение позволяет модифицировать расчет веса товаров в корзине при расчете стоимости доставки, решение учитывает установленный вес товаров, вес, заданный через свойство в группе и вес по умолчанию.

Установка решения.

1. Добавляем следующий контроллер.

```
<?php

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

*/***
** Контроллер для расчета веса товаров в корзине.*

** @author Maxim Zasorin, KAD Systems (©) 2018*
** @date 15-02-2018*
**/*

```
class Shop_Item_Weight_Controller extends Core_Servant_Properties
{
    protected $_allowedProperties = array(
        'groupProperty',
        'defaultWeight',
    );

    protected $_ShopCarts;
    protected $_Shop;

    public function __construct(Shop_Model $oShop)
    {
        parent::__construct();

        $this->_Shop = $oShop;
        $this->_setShopCarts();

        $this->defaultWeight = 200; // Вес в граммах
    }

    public function getItemWeight(Shop_Cart_Model $oShopCart)
    {
        $oShopGroupPropertyList =
        Core_Entity::factory('Shop_Group_Property_List', $this->_Shop->id);
        $oWeightProperty =
        $oShopGroupPropertyList->Properties->getByTagName('average_weight');

        $itemWeight = (float) $oShopCart->Shop_Item->weight;

        if (!$itemWeight && $oWeightProperty)
```

```

        {
            $oShopGroup = $oShopCart->Shop_Item->Shop_Group;

            do
            {
                $aoWeightPropertyValues =
$WeightProperty->getValues($oShopGroup->id);

                if (count($aoWeightPropertyValues))
                {
                    $itemWeight = (float)
$aoWeightPropertyValues[]->value;
                }

                $oShopGroup = $oShopGroup->getParent();
            }
            while (!$itemWeight && ($oShopGroup && $oShopGroup->id));
        }

        if (!$itemWeight)
        {
            $itemWeight = $this->defaultWeight;
        }

        return $itemWeight;
    }

    public function getTotalWeight()
    {
        $totalWeight = ;

        foreach ($this->_ShopCarts as $oShopCart)
        {
            if ($oShopCart->Shop_Item->id && $oShopCart->postpone == )
            {
                $totalWeight += $this->getItemWeight($oShopCart) *
$oShopCart->quantity;
            }
        }

        return $totalWeight;
    }

    protected function _setShopCarts()
    {
        $this->_ShopCarts =
Shop_Cart_Controller::instance()->getAll($this->_Shop);
    }
}

```

1. В ТДС корзины при расчете способов доставки после вызова
\$Shop_Delivery_Controller_Show→setUp добавляем следующий код:

```
$oShopItemWeightController = new Shop_Item_Weight_Controller($oShop);  
$oShopItemWeightController  
    ->groupProperty('average_weight')  
    ->defaultWeight(3000);  
  
$Shop_Delivery_Controller_Show  
    ->totalWeight(  
        $oShopItemWeightController->getTotalWeight()  
    );
```