

# Добавление минимальных и максимальных значений свойств товаров и ярлыков

## Описание

У нас уже имеется решение которое корректирует работу фильтров для ярлыков. Однако фильтры бывают не всегда стандартными и для их корректной работы необходимы минимальное и максимальное значение какого-либо свойства. В стандартном контроллере уже есть метод `addMinMaxPrice()` который добавляет минимальное и максимальное значение цены товаров в группе. Однако метода, который бы добавлял минимальное и максимальное значение свойства товара или ярлыка нет.

Вот пример такого фильтра, для корректной работы которого необходимо знать минимальное и максимальное значение дополнительных свойств:

Цена от: 212 до: 1016

Диаметр колеса, мм от: 61 до: 104

Грузоподъемность колеса, кг от: 46 до: 58

Значения этих свойств добавляется при помощи метода `addMinMaxProperty`:

```
public function addMinMaxProperty($aProperties_id)
{
    $iCurrentShopGroup = intval($this->group); //Получаем номер текущей группы
    $arItemsId = array();
    $oShortcuts = Core_Entity::factory('shop_item');

    //Находим все ярлыки
    $oShortcuts->queryBuilder()
        ->where('shop_items.shortcut_id', '!=', )
        ->where('shop_items.shop_group_id', '=', $iCurrentShopGroup);
    $aoShortcuts = $oShortcuts->findAll();

    //Записываем в массив id всех ярлыков
    $aShortcutsId = array();
    foreach ($aoShortcuts as $oShortcut)
    {
        $aShortcutsId[] = $oShortcut->shortcut_id;
    }
}
```

```

$shop_Items = Core_Entity::factory('shop_item');

//Если есть ярлыки
if(count($aShortcutsId) > )
{
    $shop_Items->queryBuilder()
        ->where('shop_items.shop_group_id', '=', $iCurrentShopGroup)
        ->where('shop_items.shortcut_id', '=', )
        ->setOr()
        ->where('shop_items.id', 'IN', $aShortcutsId);
}
else
{
    $shop_Items->queryBuilder()
        ->where('shop_items.shop_group_id', '=', $iCurrentShopGroup);
}

$aShop_Items = $shop_Items->findAll();

foreach($aProperties_id as $property_id)
{
    $aPropertyValues = '';
    $oProperty = Core_Entity::factory('Property', $property_id);

    foreach($aShop_Items as $oShop_Item)
    {
        $oPropertyValue = $oProperty->getValues($oShop_Item->id, FALSE);

        if(isset($oPropertyValue[]))
        {
            $aPropertyValues[] = $oPropertyValue[]->value;
        }
    }

    if(is_array($aPropertyValues))
    {
        $minPropertyValue = min($aPropertyValues);
        $maxPropertyValue = max($aPropertyValues);

        $this->addEntity(
            Core::factory('Core_Xml_Entity')
                ->name('minPropertyValue')
                ->value($minPropertyValue)
                ->addAttribute('id', $property_id)
        )
        ->addEntity(
            Core::factory('Core_Xml_Entity')
                ->name('maxPropertyValue')
                ->value($maxPropertyValue)
                ->addAttribute('id', $property_id)
        );
    }
}

```

```
    }  
}  
  
return $this;  
}
```

Этот метод принимает массив идентификаторов свойств, которые необходимо узнать, а возвращает в XML тэги `<minPropertyValue id="">` и `<maxPropertyValue id="">`, где `id` это идентификатор свойства, который мы передаем в качестве аргумента.

## Пример использования

Вызов в макете или ТДС:

```
$Kad_Shop_Controller_Show  
->group($iCurrentShopGroup) // Указываем группу  
->->addMinMaxProperty(array(423,424))
```

В результате работы в XML появятся теги с минимальным и максимальным значением свойств, `id` которых были переданы в массиве в качестве аргумента:

```
<minPropertyValue id="423">50</minPropertyValue>  
<maxPropertyValue id="423">125</maxPropertyValue>  
<minPropertyValue id="424">35</minPropertyValue>  
<maxPropertyValue id="424">80</maxPropertyValue>
```

Страницу создал Егор Козырев 10.10.17 в 10:55