Короткие URL

> Нужно в этом решнии предусмотреть проверку на совпадение URL товаров в разных группах.

1. ТДС интернет магазин, настройки страницы, вставить перед:

```
$Shop_Controller_Show
    ->limit($limit)
    ->parseUrl();
```

Код:

```
$newPath = Core::$url['path'];
$partsPath = explode('/', $newPath);
$briefUrl = false;

if(count($partsPath) == 4)
{
    $partsPath = array_reverse($partsPath);
    $oItems = Core_Entity::factory('shop_item');
    $oItems->queryBuilder()
        ->where('shop_id', '=', 1)
        ->where('path', '=', $partsPath[1]);
    $aoItems = $oItems->findAll();

    if(count($aoItems) == 1)
    {
        $oldPath = $aoItems[]->Shop->Structure->getPath() .
$aoItems[]->getPath();
        Core::$url['path'] = urldecode($oldPath);
        $briefUrl = true;
    }
    elseif(count($aoItems) > 1)
    {
        $oldPath = $aoItems[]->Shop->Structure->getPath() .
$aoItems[]->getPath();
        Core::$url['path'] = urldecode($oldPath);
        $briefUrl = true;

        Core_Log::instance()
            ->clear()
            ->status(4)
            ->write("Найдено более одного товара с путём $partsPath[1], выводится
первый из них");
    }
```

```
}
```

После:

```
$Shop_Controller_Show
    ->limit($limit)
    ->parseUrl();
```

Добавить:

```
Core::$url['path'] = $newPath;
```

Что бы сделать редирект на новый адрес. Нужно после

```
$Shop_Controller_Show
    ->limit($limit)
    ->parseUrl();
```

Добавить:

```
if($Shop_Controller_Show->item && $briefUrl == false )
{
    $view_item = $Shop_Controller_Show->item;
    $item_path = Core_Entity::factory('Shop_Item', $view_item)->path;
    $shop_path = $oShop->Structure->path;
    header("Location: /$shop_path/{$item_path}/");
    exit();
}
```

2. Меняем в нужных XSLT шаблонах в `shop_item` ссылки. Вместо `{url}` вставить `{/shop/url}{path}/`.

3. Для того чтобы на карте сайта (/sitemap/) выводились товары с изменённым url, необходимо в ТДС Google SiteMap в настройках страницы создать новый класс, который наследуется от `Core_Sitemap`, и переопределить метод `_structure` в котором изменяется часть с url товара.

```
class Core_Sitemap_Ixta extends Core_Sitemap
{
    protected function _structure($structure_id = )
    {
        $oSite = $this->getSite();

        $aStructure = $this->_selectStructuresByParentId($structure_id);

        $dateTime = Core_Date::timestamp2sql(time());

        $oSite_Alias = $oSite->getCurrentAlias();

        foreach ($aStructure as $oStructure)
        {
```

```
            $sProtocol = $oStructure->https
                ? 'https://'
                : 'http://';

            $this->addNode($sProtocol . $oSite_Alias->name .
$oStructure->getPath(), $oStructure->changefreq, $oStructure->priority);

            // Informationsystem
            if ($this->showInformationsystemGroups &&
isset($this->_Informationsystems[$oStructure->id]))
            {
                $oInformationsystem =
$this->_Informationsystems[$oStructure->id];

                $offset = ;

                do {
                    $oInformationsystem_Groups =
$oInformationsystem->Informationsystem_Groups;
                    $oInformationsystem_Groups->queryBuilder()
                        ->select('informationsystem_groups.id',
                            'informationsystem_groups.informationsystem_id',
                            'informationsystem_groups.parent_id',
                            'informationsystem_groups.path'
                            )
->where('informationsystem_groups.siteuser_group_id', 'IN',
$this->_aSiteuserGroups)
                        ->where('informationsystem_groups.active', '=', 1)
                        ->where('informationsystem_groups.indexing', '=', 1)
                        ->offset($offset)->limit($this->limit);

                    $aInformationsystem_Groups =
$oInformationsystem_Groups->findAll(FALSE);

                    $path = $sProtocol . $oSite_Alias->name .
$oInformationsystem->Structure->getPath();

                    foreach ($aInformationsystem_Groups as
$oInformationsystem_Group)
                    {
                        $this->addNode($path .
$oInformationsystem_Group->getPath(), $oStructure->changefreq,
$oStructure->priority);
                    }
                    $offset += $this->limit;
                }
                while (count($aInformationsystem_Groups));

                // Informationsystem's items
                if ($this->showInformationsystemItems)
                {
```

```
                    $offset = ;

                    do {
                        $oInformationsystem_Items =
$oInformationsystem->Informationsystem_Items;
                        $oInformationsystem_Items->queryBuilder()
                            ->select('informationsystem_items.id',
'informationsystem_items.informationsystem_id',
'informationsystem_items.informationsystem_group_id',
                                'informationsystem_items.shortcut_id',
                                'informationsystem_items.path'
                                )
                            ->open()
->where('informationsystem_items.start_datetime', '<', $dateTime)
                            ->setOr()
->where('informationsystem_items.start_datetime', '=', '0000-00-00
00:00:00')
                            ->close()
                            ->setAnd()
                            ->open()
                            ->where('informationsystem_items.end_datetime',
'>', $dateTime)
                            ->setOr()
                            ->where('informationsystem_items.end_datetime',
'=', '0000-00-00 00:00:00')
                            ->close()
->where('informationsystem_items.siteuser_group_id', 'IN',
$this->_aSiteuserGroups)
                            ->where('informationsystem_items.active', '=',
1)
                            ->where('informationsystem_items.shortcut_id',
'=', )
                            ->where('informationsystem_items.indexing', '=',
1)
                            ->offset($offset)->limit($this->limit);

                        $aInformationsystem_Items =
$oInformationsystem_Items->findAll(FALSE);
                        foreach ($aInformationsystem_Items as
$oInformationsystem_Item)
                        {
                            $this->addNode($path .
$oInformationsystem_Item->getPath(), $oStructure->changefreq,
$oStructure->priority);
                        }

                        $offset += $this->limit;
                    }
                    while (count($aInformationsystem_Items));
                }
            }
```

```
            // Shop
            if ($this->showShopGroups &&
isset($this->_Shops[$oStructure->id]))
            {
                $oShop = $this->_Shops[$oStructure->id];

                $offset = ;

                do {
                    $oShop_Groups = $oShop->Shop_Groups;
                    $oShop_Groups->queryBuilder()
                        ->select('shop_groups.id',
                            'shop_groups.shop_id',
                            'shop_groups.parent_id',
                            'shop_groups.path'
                            )
                        ->where('shop_groups.siteuser_group_id', 'IN',
$this->_aSiteuserGroups)
                        ->where('shop_groups.active', '=', 1)
                        ->where('shop_groups.indexing', '=', 1)
                        ->offset($offset)->limit($this->limit);

                    $aShop_Groups = $oShop_Groups->findAll(FALSE);

                    $path = $sProtocol . $oSite_Alias->name .
$oShop->Structure->getPath();
                    foreach ($aShop_Groups as $oShop_Group)
                    {
                        $this->addNode($path . $oShop_Group->getPath(),
$oStructure->changefreq, $oStructure->priority);
                    }

                    $offset += $this->limit;
                }
                while (count($aShop_Groups));

                // Shop's items
                if ($this->showShopItems)
                {
                    $offset = ;

                    do {
                        $oShop_Items = $oShop->Shop_Items;
                        $oShop_Items->queryBuilder()
                            ->select('shop_items.id',
                                'shop_items.shop_id',
                                'shop_items.shop_group_id',
                                'shop_items.shortcut_id',
                                'shop_items.modification_id',
                                'shop_items.path'
```

```
                                    )
                                ->open()
                                ->where('shop_items.start_datetime', '<',
$dateTime)
                                ->setOr()
                                ->where('shop_items.start_datetime', '=',
'0000-00-00 00:00:00')
                                ->close()
                                ->setAnd()
                                ->open()
                                ->where('shop_items.end_datetime', '>',
$dateTime)
                                ->setOr()
                                ->where('shop_items.end_datetime', '=',
'0000-00-00 00:00:00')
                                ->close()
                                ->where('shop_items.siteuser_group_id', 'IN',
$this->_aSiteuserGroups)
                                ->where('shop_items.active', '=', 1)
                                ->where('shop_items.shortcut_id', '=', )
                                ->where('shop_items.indexing', '=', 1)
                                ->offset($offset)->limit($this->limit);

                            // Modifications
                            if (!$this->showModifications)
                            {
                                $oShop_Items->queryBuilder()
                                    ->where('shop_items.modification_id', '=',
);
                            }

                            $aShop_Items = $oShop_Items->findAll(FALSE);
                            foreach ($aShop_Items as $oShop_Item)
                            {
                                $this->addNode($path . $oShop_Item->path . '/',
$oStructure->changefreq, $oStructure->priority);
                            }

                            $offset += $this->limit;
                        }
                        while (count($aShop_Items));
                    }
                }

                // Structure
                $this->_structure($oStructure->id);
            }

            return $this;
        }
}
```

И ниже уже используем новый класс:

```
$oCore_Sitemap = new Core_Sitemap_Ixta($oSite);
```

Вместо старого:

```
$oCore_Sitemap = new Core_Sitemap($oSite);
```

Страницу создал Константин Сериков 18.10.16 в 17:07