



UXPACK

UX PACK

Jiří Chmiel

© Copyright 2013 - ARTIO International Co.

UX Pack

ARTIO

Publication date: 27th March 2013

Version: 1.0.0

Abstract

User documentation for ARTIO Booking & Reservations (Book-it!) component for CMS Joomla!. This document describes the component requirements, installation and usage. All main component parts are described in more details, including illustration images and examples. It specifies server and browser requirements and information about environment constraints. The documentation is helpful for both web developers and website administrators / owners.

Table of Contents

1. Overview	1
2. Compatibility	2
3. Installation	3
4. Settings	4
4.1. Activation	4
4.2. Basic Settings	4
4.3. Extended Settings	4
4.4. Endless page Settings	5

Chapter 1. Overview

Magento module Artio UX Pack provides the AJAX wrapper for layer navigation on catalog category view page, search result page and advanced search result page.

This module also replaces the standard pager system by "endless page system", where the pages are continuously loaded when user scrolls the page in his browser.

Chapter 2. Compatibility

Magento Community Edition 1.4.1.0 - 1.7.0.2 Magento Enterprise Edition 1.12.0.2

NOTE: If you use Magento 1.4.x or 1.5.x we recommend (but not required) the upgrade JS Prototype framework to version 1.7 or using X-UA Compatibility Meta Tag IE8

Chapter 3. Installation

Extract distribution package (ZIP) to root of your website and refresh standard Magento cache.

Chapter 4. Settings

You can find all configuration settings in System/Configuration/Catalog/UXPack.

There are four configuration sections. First "Activation" and second "Basic Settings" for common users. Third "Extended Settings" and fourth "Endless page Settings" is intended for Magento Developers which can make compatibility with the various design templates. There assumes a basic knowledge of HTML, CSS and Magento design developing.

4.1. Activation

Activation Key - here enter your activation key, which was sent with installation package.

Note: Activation key is related to your domain. If you use it for differ domain the activation key will not be accepted.

Note: Activation key is related to Magento version. If you use key for CE version on EE version or vice versa the activation key will not be accepted.

4.2. Basic Settings

Enabled AJAX loading products - enable or disable AJAX loading. This is the main settings of this module. If you choose "No" module will be deactivated.

Note: You can activate this module only if you entered the correct activation key.

Enabled endless pages - enable or disable "endless page system", where products page are continuously loaded when user scrolls page in his browser. It can be enabled only if previous settings is set to "Yes".

Note: If you want to use endless page for grid view mode then set the number of products per page as integer multiple by number of columns (3,6,9 for 3 columns), otherwise there will appear holes (empty blocks). The number of columns you can change at "System/Configuration/Catalog/Frontend"

4.3. Extended Settings

AJAX action - internally identification of "category view" action. For most cases there should be set to "catalog_category_view". Change this only if you use some module 3rd party which rewrites standard "Catalog_Category" controller.

AJAX search action - internally identification of "search result" action. For most cases there should be set to "catalogsearch_result_index". Change it only if you use some module 3rd party which rewrites standard "CatalogSearch_Result" controller.

AJAX advanced search action - internally identification of "advanced search result" action. For most cases there should be set to "catalogsearch_advanced_result". Change it only if you use some module 3rd party which rewrites standard "CatalogSearch_Advanced" controller.

AJAX blocks - the list of the loaded blocks by AJAX. It says which block from the layout will be placed into which DOM element in HTML. This settings has following format:

layout_name1 => css_selector1;layout_name2 => css_selector2 ...

Remember these rules:

- "enterprisecatalog.leftnav=>div.block-layered-nav;" means replace DOM element <DIV> with class `.block-layered-nav` by block with layout name `enterprisecatalog.leftnav`
- "product_list=>!div.category-view" means insert block with layout name `product_list` into DOM element <DIV> with class `.category-view`
- "!" at the start CSS selector means place the block INTO element, not REPLACE the element.

Default settings for basic design template of CE edition: "catalog.leftnav=>div.block-layered-nav;catalogsearch.leftnav=>div.block-layered-nav;product_list=>div.category-products;search_result_list=>div.category-products"

Default settings for basic design template of EE edition: "enterprisecatalog.leftnav=>div.block-layered-nav;enterprisearch.leftnav=>div.block-layered-nav;product_list=>!div.category-view;search_result_list=>!div.results-view"

CSS selectors of control elements - defines the list of elements which can initiate AJAX loading.

There is supported the events `click` on elements <A> and the events `change` on elements <SELECT>. If user click on <A> then the list of products will be loaded from URL defined by HREF attribute of it. If user change the value of <SELECT> product list will be loaded from URL defined by VALUE attribute of chosen value.

Default settings for basic design template of CE edition: "div.block-layered-nav a,div.category-products div.toolbar a,div.category-products div.toolbar select"

Default settings for basic design template of EE edition: "div.block-layered-nav a,div.toolbar a,div.toolbar select"

4.4. Endless page Settings

Layout names of main block - defines which blocks represents block with product list. The values separate by comma. Default settings is "product_list,search_result_list". Change it only if you are using other layout names for these blocks.

Note if you change this value do not forget to change also AJAX blocks setting in previous section.

CSS selector of main block - defines the DOM element with product list. Default settings is "div.category-products"

Relative level to activate AJAX loading (%) - defines when should be loaded the next page. If you choose 100 then next page will be loaded if bottom line (100% of height) of the product list block is visible (it appears in view rectangle of browser). Default settings is 66.

CSS selector of container element (grid), Target position in container (grid) - defines the target position of product list in view mode GRID. First value defines DOM element, the second value defines position. For example settings "div.products" and "Bottom" says: "Place product list in view mode GRID at the bottom of element <DIV> with class "products". This controls WHERE will be block placed. The place depends on your design template.

Default settings for basic design template of CE edition:

"div.toolbar-bottom" and "Before"

Default settings for basic design template of EE edition:

"div.category-products" and "Bottom"

Before HTML (grid), Start substring (grid), Start substring type (grid), End substring (grid), End substring type (grid), After HTML (grid) - previous settings means WHERE will be block placed, this settings means WHAT will be placed. Suppose your template for product list in view mode GRID would looks like this:

```
<p>My Products</p>
<ul class="product-list">
  <li class="product">
    <span>Product 1</span>
    <span>100.00</span>
  </li>
  <li class="product">
    <span>Product 2</span>
    <span>400.00</span>
  </li>
  <li class="product">
    <span>Product 3</span>
    <span>400.00</span>
  </li>
  <li class="product">
    <span>Product 4</span>
    <span>400.00</span>
  </li>
</ul>
<p>This was our products</p>
```

We need place only the list of elements, not wrapper and <P> around them. Therefore we set "Start substring" to "<li" and "End substring" to "". This extracts from your template all between first occurrence "<li" and last occurrence "".

The settings "Start substring type" and "End substring type" determines whether HTML includes border substrings or not.

By settings "Before HTML" and "After HTML" you can add some HTML before or after product list HTML. It is good for placing for example some Java Scripts does some corrections (adding CSS class first/last, etc.).

Default settings for basic design template of CE and EE edition:

Before HTML (grid):	"<script type="text/javascript">\$\$('ul.products-grid').each(function(el) { el.removeClassName('odd'); el.removeClassName('even'); el.removeClassName(' first'); el.removeClassName('last'); });</script>"
Start substring (grid):	"<ul"
Start substring type (grid):	"Outer"
End substring (grid):	""
End substring type (grid):	"Inner"
After HTML (grid):	"<script type="text/javascript">decorateGeneric(\$\$('ul.products-grid'), ['o dd','even','first','last'])</script>"

CSS selector of container element (list) AND Target position in container (list) - this is the same meaning also "CSS selector of container element (grid)" and "Target position in container (grid)" but for view mode LIST.

Default settings for basic design template of CE and EE edition:

"ol#products-list" and "Bottom"

Before HTML (list), Start substring (list), Start substring type (list), End substring (list), End substring type (list), After HTML (list) - this is the same meaning also in previous case but for view mode LIST.

Default settings for basic design template of CE and EE edition:

Before HTML (list): `"<script type='text/javascript'>$$('ol#products-list li').each(function(e l){ el.removeClassName('even'); el.removeClassName('odd'); el.removeClassName('last'); });</script>"`

Start substring (list): `"<li"`

Start substring type (list): `"Outer"`

End substring (list): `""`

End substring type (list): `"Inner"`

After HTML (list): `"<script type='text/javascript'>decorateList('products-list', 'none-recursive')</script>"`

Layout names of blocks to remove - defines blocks which will not be generated to next page. There is need to remove for example "Toolbar Block", because on page must be only once at the start page and once the bottom page, not between pages. The default settings is "product_list_toolbar".

CSS selector of elements to hide - defines DOM elements which will be hidden after page loading. There is need to hide the pager and the select for choosing the number of page at page (these element has no sense for "endless page"). The default value is "div.limiter,div.pages".

Products counters - defines the way to show information about the number of displayed products and total number of products. This settings has follow format:

`css_selector1=>format1;css_selector2=>format2;`

For example "span.counter=>There is %n / %t" means that into element with class "counter" will be displayed text "There is x / y", where x is the number of currently displayed products and y is number of total products.

Default settings for basic design template of CE edition: `"p.amount => Items 1 to %n of %t total"`

Default settings for basic design template of EE edition: `"p.amount => Items 1-%n of %t"`