

Layout Customizer API



This page is no longer valid. For information about the Layout Reorderer, see [Layout Reorderer API](#).



This documentation refers to the trunk version of the Layout Customizer code. For documentation specific to the v0.4 release, please see [Layout Customizer API - v0.4](#).

Overview

The Layout Customizer can be used to provide the ability to change the layout of content modules, for example portlets within a portal environment. It can be used to organize arbitrary pieces of content, or "modules," provided they are laid out in columns. For details about how modules must be laid out for use with the Layout Customizer, see [Tutorial - Layout Reorderer Selectors](#).

The Layout Customizer is initialised with two data structures describing the initial layout of the modules and any restrictions on their movement. It can be customized by providing any of a number of optional parameters.

Initializing the Layout Customizer

```
fluid.initLayoutCustomizer(layout, perms,  
orderChangedCallbackUrl[, options]);
```

Parameters

On This Page

- [Overview](#)
- [Initializing the Layout Customizer](#)
- [Parameters](#)
 - [layout](#)
 - [perms](#)
 - [orderChangedCallbackUrl](#)
 - [options](#)
- [Dependencies](#)

See Also

- [Reorderer API](#)
- [Layout Reorderer API](#)
- [Tutorial - Layout Reorderer Selectors](#)

Still need help?

Join the [fluid-talk mailing list](#) and ask your questions there.

layout

The `layout` object specifies the IDs of the DOM elements and the relationships between them.

For more information about the `layout` object, see the [Layout Customizer - Layout and Permissions](#).

perms

The `perms` object is a two-dimensional array of binary numbers (i.e. 0s and 1s) describing any restrictions on where modules can be moved. If there are no restrictions, pass `null` for this parameter.

In the array,

- each row describes the permissions for one module, i.e. the first row describes where the first module can or can't go
- each column in that row describes the permissions for one of the drop targets, i.e. the first column refers to the first drop target, the second column refers to the second drop target, etc.

For details about drop targets and the `perms` object, see the [Layout Customizer - Layout and Permissions](#).

orderChangedCallbackUrl

The `orderChangedCallbackUrl` is the URL that the Layout Customizer should use to communicate changes in the `layout` to the server. The Layout Customizer will POST the updated `layout` to the URL, and expects an updated `perms` object in response. If there are no restrictions on permissions, the `perms` object should contain all 1s (for details about the structure of the `perms` object, see the [Layout Customizer - Layout and Permissions](#)).

options

The `options` parameter is an *optional* collection of name-value pairs that configure the Layout Customizer:

Selectors

Error rendering macro 'excerpt-include'

No link could be created for 'Layout Customizer Selectors'.

General options

Layout Customizer-specific options

Name	Description	Values	Default
<code>layout</code>	Javascript object. For information about the structure of this object, see Tutorial - Layout Reorderer Selectors .	<pre>var layout = { id: <id of container>, columns: [{ id:<id of column>, children: [<id of module>, <id of module>...] }, { id:<id of column>, children: [<id of module>, <id of module>...] }, ...] };</pre>	none
<code>afterMoveCallbackUrl</code>	If an URL is provided with this option, the current state of the component model will be sent to that URL after a move is carried out.	an URL	none
<code>acquireModel</code>	Valid only when the <code>afterMoveCallbackUrl</code> option is provided. <code>acquireModel</code> is a function that returns a Javascript object representing the model for the component. The model will be converted to a JSON string and sent to the URL specified by the <code>afterMoveCallbackUrl</code> option.	a function	none

Dependencies

The Layout Customizer dependencies can be met by including the minified `Fluid-all.js` file in the header of the HTML file:

```
<script type="text/javascript" src="Fluid-all.js"></script>
```

Alternatively, the individual file requirements are:

```
<script type="text/javascript" src="jquery/jquery-1.2.6.js"></script>
<script type="text/javascript" src="jquery/jARIA.js"></script>
<script type="text/javascript" src="jquery/jquery.keyboard-ally.js"></script>
<script type="text/javascript" src="jquery/ui.core.js"></script>
<script type="text/javascript" src="jquery/ui.draggable.js"></script>
<script type="text/javascript" src="json/json2.js"></script>
<script type="text/javascript" src="fluid/Fluid.js"></script>
<script type="text/javascript" src="fluid/FluidDOMUtilities.js"></script>
<script type="text/javascript" src="fluid/GeometricManager.js"></script>
<script type="text/javascript" src="fluid/Reorderer.js"></script>
<script type="text/javascript" src="fluid/LayoutCustomizer.js"></script>
<script type="text/javascript" src="fluid/ModuleLayout.js"></script>
```