Layout Customizer API - v0.4

This documentation refers to the v0.4 released version of the Layout Customizer code. For documentation specific to the trunk, please see Layout Customizer API.

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 Layout Customizer Integration - Layout and Permissions - v0.4.

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

See Also

- Layout Customizer Integration - Layout and Permissions - v0.4
- ModuleLayoutHandler API - v0.4
- Reorderer API - v0.4

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 Integration - Layout and Permissions - v0.4.

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 Integration - Layout and Permissions - v0.4.

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 Integration - Layout and Permissions - v0.4).

options - optional

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

Name	Description	Values	Default

`role`	TODO: explain the use of ARIA roles, and the implications of the defaults	`fluid.roles.LIST`
		`fluid.roles.GRID`
		`fluid.roles.REGIONS`
		other:
		`var myRole = {`
		`container: <ariaRole>,`
		`item: <ariaRole>`
		`};`
			`fluid.roles.REGIONS`

`keysets`	an object containing sets of keycodes to use for directional navigation, and for the modifier key used for moving a module.	The object must be a list of objects containing the following keys:
		`up`
		`down`
		`right`
		`left`
		`modifier` : a function that returns `true` or `false`, indicating whether or not the required modifier(s) are activated
			`fluid.defaultKeysets = [{ up : fluid.keys.UP, down : fluid.keys.DOWN, right : fluid.keys.RIGHT, left : fluid.keys.LEFT, modifier : function (evt) { return evt.ctrlKey; } }, { up : fluid.keys.i, down : fluid.keys.m, right : fluid.keys.k, left : fluid.keys.j, modifier : function (evt) { return evt.ctrlKey; } }];`

`cssClassNames`	an object containing class names for styling the Layout Customizer. For a discussion of CSS styling of the Layout Customizer, see the CSS Classes documentation for the Reorderer API - v0.4.	The object may contain any of the keys defined in the `defaultCssClassNames` (show to the right). Class names not provided will revert to the default.	`var defaultCssClassNames = {`
			
			`defaultStyle: "orderable-default",`
			`selected: "orderable-selected",`
			`dragging: "orderable-dragging",`
			`mouseDrag: "orderable-dragging",`
			`hover: "orderable-hover",`
			
			`dropMarker: "orderable-drop-marker",`
			`avatar: "orderable-avatar"`
			`};`

`avatarCreator`	a function that returns a valid DOM node to be used as the dragging avatar		The item being dragged will be cloned
`dropWarningId`	the ID of an element on the page that should be displayed if users attempt to move an item to a location where movement is not permitted		(none)
`instructionMessageId`	the ID of an element on the page containing any instructional messages		"message-bundle:"

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-a11y.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="jquery/ui.droppable.js"></script>
<script type="text/javascript" src="fluid/Fluid.js"></script>
<script type="text/javascript" src="fluid/Reorderer.js"></script>
<script type="text/javascript" src="fluid/LayourCustomizer.js"></script>
```