Table of Contents Levels

The Levels subcomponent is used by Table of Contents API to render the generated list of links to headers into a template.

Do not use this component directly. The information on this page should be used when customizing Table of Contents.

See Also
Table of Contents API

Supported Events	*none*
Methods	*none*

Options	`model`
	An array of information about the headings in the document.
	
	`produceTree`
	A function that will return a Renderer Component Tree.
	
	`rendererFnOptions`
	Options passed on to the renderer function.
	
	`rendererOptions`
	Options passed on to the renderer.
	
	`repeatingSelectors`
	An array of selector names for elements that will be repeated.
	
	`resources`
	Information about resources that need to be loaded.
	
	selectors
	See below.

Selectors	`level1 through level6`
	Identify the elements in the template for the levels.
	
	`items1 through items6`
	Identify the elements in the template for the items at each level.
	
	`link1 through link6`
	Identify the elements in the template for the links to the headings.

Options

back to top The second argument to the creator function is the options argument. This is a JavaScript object containing name/value pairs: The name is the name of the option and the value is the desired setting. Components define their own default values for options, but integrators can override these defaults by providing new values using the options argument. For technical information about how options are merged with defaults, see Options Merging.

model

Description	An object containing an array of information about the headings in the document. It should have the following form:

	``
	{
	    heading: [{
	        text: <heading text>,
	        url: <link to anchor associated with the heading>
	        headings: [<array of heading info for subheadings>]
	    },
	    ...
	    ]
	}
	``
	
	By default, this information is generated by the component as part of its initialization process.

| Default | ```
{
    heading: []
}
``` |
| --- | --- |

produceTree

Description	A function that will return a [Renderer Component Tree](#) for the table of contents.
Default	`"fluid.tableOfContents.levels.produceTree"`
| Example | ```
produceTree: "customTreeProducerFn"
``` |
| See also | [Renderer Components](#) |

rendererFnOptions

Description	This option is an object containing options that will be passed on to the renderer function created for this component.
| Default | ```
{
    noexpand: true
}
``` |
| Example | ```
rendererFnOptions: {
    noexpand: true
}
``` |
| See also | [Renderer Components](#) |

rendererOptions

Description	This option is an object containing options that will be passed on to the Renderer.
| Default | ```
{
    debugMode: false
}
``` |
| Example | ```
rendererOptions: {
    debugMode: true
}
``` |
| See also | [Renderer Components](#) |

repeatingSelectors

Description	This option is an array of the selector names that identify elements in the template that will be replicated by the renderer as needed, based on the data being rendered.
| Default | ```
["level1", "level2", "level3", "level4", "level5", "level6",
  "items1", "items2", "items3", "items4", "items5", "items6"]
``` |

resources

Description	An object describing any resources that need to be loaded.

Default	```
	{
	    template: {
	        forceCache: true,
	        url: "../html/TableOfContents.html"
	    }
	}
	``

Example	```
	resources: {
	    template: {
	        url: "../templates/CustomToCTemplate.html"
	    }
	}
	``

See also	Renderer Components

Selectors

back to top One of the options that can be provided to the component is a set of CSS-based selectors identifying where in the DOM different elements can be found. The value for the option is itself a Javascript object containing name/value pairs:

```
selectors: {
    selector1Name: "selector 1 string",
    selector2Name: "selector 2 string",
        ...
}
```

The component defines defaults for these selectors: If you use those defaults in your markup, you do not need to specify the selectors option. If you do choose to override any of the selectors, you can specify your custom selector using this option.

level1 through level6

Description	These selectors identify the elements in the template that serve as containers for the different levels of header links. In the default template, these selectors identify `<ul>` elements.
Default	`".flc-toc-levels-level1"` through `".flc-toc-levels-level6"`

Example	```
	selectors: {
	    level1: "ol.level1",
	    level2: "ol.level2",
	    level3: "ol.level3",
	    ...
	}
	``

items1 through items6

Description	These selectors identify the elements in the template that serve as containers for the links at each level. In the default template, these selectors identify `<li>` elements.
Default	`".flc-toc-levels-items1"` through `".flc-toc-levels-items6"`

Example	``` selectors: { items1: ".level1 div", items2: ".level2 div", items3: ".level3 div", ... } ```

link1 through link6

Description	These selectors identify the link elements at each level. In the default template, these selectors identify `<a>` elements.
Default	`".flc-toc-levels-link1"` through `".flc-toc-levels-link6"`
Example	``` selectors: { link1: "a.tocLinks" } ```