Inline Edit API - v0.5

ⓘ This documentation refers to version 0.5 of the Inline Edit code. For documentation specific to trunk, please see Simple Text Inline Edit API.

Inline Edit Overview

The Inline Edit allows users to edit content within the context of their work rather than going to an "edit mode". It can be applied to any text, assuming a very simple contract is maintained:

1. The elements that are part of the Inline Edit component have some kind of container element.
2. The text you wish to make editable is within the component container.

You can optionally provide your own markup that will be used for the edit mode, but if not, default markup is provided.

Creating a single Inline Edit

```
fluid.inlineEdit(componentContainer, options);
```

Return: The Inline Edit component object.

On This Page

See Also

- Inline Edit
- Inline Edit Tutorial - v0.5

Still need help?

Join the fluid-talk mailing list and ask your questions there.

Parameters

componentContainer

The `componentContainer` parameter is a selector, a single-element jQuery, or a DOM element representing the component's container.

options

The `options` parameter is an optional collection of name-value pairs that configure the Inline Edit component:

Name	Description	Values	Default

`selectors`	Javascript object containing selectors for various fragments of the Inline Edit component.	The object can contain any subset of the following keys:
		`text`
		`editContainer`
		`edit`
		Any values not provided will revert to the default.
			`selectors: {`
			`text: ".text",`
			`editContainer: ".editContainer",`
			`edit: ".edit"`
			`}`

`styles`	Javascript object containing CSS style names that will be applied to the Inline Edit component.	The object can contain any subset of the following keys:
		`invitation`
		`defaultViewText`
		`tooltip`
		`focus`
		Any values not provided will revert to the default.
			`styles: {`
			`invitation: "inlineEdit-invitation",`
			`defaultViewText: "inlineEdit-invitation-text",`
			`tooltip: "inlineEdit-tooltip",`
			`focus: "inlineEdit-focus"`
			`}`

`paddings`	Javascript object containing pixel values that will configure the size of the edit field.	The object can contain any subset of the following keys:
		`edit`
		`minimumEdit`
		`minimumView`
		Any values not provided will revert to the default.
			`paddings: {`
			`edit: 10,`
			`minimumEdit: 80,`
			`minimumView: 60`
			`}`

`finishedEditing`	A function that will be called each time the Inline Edit component leaves edit mode. This function can be used to communicate changes in the field to the server, or to take any other action desired by the implementor.	a function	undefined

`editModeRenderer`	A function that creates the editable field based on the display text. This function is used if no edit container is provided in the markup.	a function that returns an object of the following form:
		`return {`
		`container: editContainer,`
		`field: editField`
		`};`
		where `editContainer` is a container element for the edit field itself and `editField` is the edit field itself.
			a function that creates the edit field based on the following template:
			`<span>`
			`<input type='text' />`
			`</span>`

`selectOnEdit`	Indicates whether or not to automatically select the editable text when the component switches into edit mode.	boolean	`false`
`defaultViewText`	The default text to use when filling in an empty component. Set to empty to suppress this behaviour	string	"Click here to edit"
`useTooltip`	Indicates whether or not the component should display a custom ("invitation") tooltip on mouse hover	boolean	`false`
`tooltipText`	The text to use for the tooltip to be displayed when hovering the mouse over the component	string	"Click item to edit"
`tooltipId`	The id to be used for the DOM node holding the tooltip	string	"tooltip"
`tooltipDelay`	The delay, in ms, between starting to hover over the component and showing the tooltip	number	2000

Functions

```
fluid.inlineEdit.edit();
```

Switches the component into edit mode.

```
fluid.inlineEdit.finish();
```

Switches the component out of edit mode into display mode, updating the displayed text with the current content of the edit field. Calls the `finishedEditing` function provided in the `options`, if it exists.

```
fluid.inlineEdit.render(source);
```

Prompts the component to render itself.

```
fluid.inlineEdit.tooltipEnabled();
```

Returns a boolean indicating whether or not the tooltip is enabled.

```
fluid.inlineEdit.updateModel(newValue, source);
```

Updates the component's internal representation of the text to a new value. The component will be re-rendered.

Creating Multiple Inline Edits

```
fluid.inlineEdits(componentContainer, options);
```

This function will find any elements within the given container that are identified as 'editables' and apply the Inline Edit component to them.

Parameters

componentContainer

The `componentContainer` parameter is a selector, a single-element jQuery, or a DOM element specifying the root DOM node of the Inline Edit markup.

options

The `options` parameter is an optional collection of name-value pairs that configure the Inline Edit components. The available options are the same as those for the creation of a single Inline edit (described fluid:above), with the addition of a selector for identifying the editable elements. The default selector is defined as follows:

```
selectors: {
    editables: ".inlineEditable"
}
```

Dependencies

The Inline Edit 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/jquery.tooltip.js"></script>
<script type="text/javascript" src="fluid/Fluid.js"></script>
<script type="text/javascript" src="fluid/InlineEdit.js"></script>
```