

Rich Text Inline Edit API

Rich Text Inline Edit Overview

The Rich Text Inline Edit is a specific integration of the Inline Edit component that uses a rich text editor to allow users to quickly make edits to formatted text. Two versions of Rich Text Inline Edit are provided, each using a different rich text editor:

1. CKEditor
2. TinyMCE

Creation

Note: The CKEditor integration is **new in v1.2**. It replaces the deprecated FCKEditor integration, which is officially removed as of v1.4.

```
fluid.inlineEdit.CKEditor(container, options);
```

```
fluid.inlineEdit.TinyMCE(container, options);
```

Return: The Inline Edit component object.

Status

This component is in [Sneak Peek status](#)

On This Page

- [Rich Text Inline Edit Overview](#)
- [Creation](#)
- [Configuration](#)
 - [Pre-configuration](#)
 - [CKEditor integration:](#)
 - [TinyMCE integration:](#)
- [Dependencies](#)

See Also

- [Inline Edit](#)
- [Rich Text Inline Edit](#)
- [Simple Text Inline Edit API](#)
- [Dropdown Inline Edit API](#)
- [Fluid Component API](#)

Still need help?

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

Configuration

The Rich Text Inline Edit is a specific configuration of the Inline Edit component. It's API is fundamentally the same as the [Simple Text Inline Edit API](#), but some of the options are pre-configured.

Pre-configuration

The configuration of the Rich Text Inline Edit integrations is accomplished through a `fluid.defaults` object (for more information about component defaults, see [Framework Functions#Defaults](#)). The defaults for the two available integrations are described below.

NOTE: If you override any of these defaults, the Rich Text Inline Edit may not function properly.

CKEditor integration:

Note: The CKEditor integration is **new in v1.2**. It replaces the deprecated FCKEditor integration.

```
fluid.defaults("fluid.inlineEdit.CKEditor", {
  selectors: {
    edit: "textarea"
  },

  styles: {
    invitation: null
  },

  displayAccessor: {
    type: "fluid.inlineEdit.richTextViewAccessor"
  },
  editAccessor: {
    type: "fluid.inlineEdit.CKEditor.viewAccessor"
  },
  lazyEditView: true,
  blurHandlerBinder: fluid.inlineEdit.CKEditor.blurHandlerBinder,
  editModeRenderer: fluid.inlineEdit.CKEditor.editModeRenderer
});
```

selectors	The CKEditor uses a <code><textarea></code> element, and so the default selector for this field is set to "textarea."
styles	
displayAccessor	The standard display accessor uses the jQuery functions <code>val</code> or <code>text</code> depending on the tag type. The <code>richTextAccessor</code> returns the HTML contents of the element instead.
editAccessor	The standard edit accessor uses the jQuery functions <code>val</code> or <code>text</code> depending on the tag type. The <code>CKEditor.viewAccessor</code> function uses the CKEditor API to access the edit field.
lazyEditView	Loading a rich text editor is expensive, and so lazy loading is enabled.
editModeRenderer	The CKEditor-specific edit mode renderer uses the CKEditor itself.

TinyMCE integration:

```
fluid.defaults("fluid.inlineEdit.tinyMCE", {
  useTooltip: true,
  selectors: {
    edit: "textarea"
  },

  styles: {
    invitation: null
  },
  displayAccessor: {
    type: "fluid.inlineEdit.richTextViewAccessor"
  },
  editAccessor: {
    type: "fluid.inlineEdit.tinyMCE.viewAccessor"
  },
  lazyEditView: true,
  blurHandlerBinder: fluid.inlineEdit.tinyMCE.blurHandlerBinder,
  editModeRenderer: fluid.inlineEdit.tinyMCE.editModeRenderer
});
```

selectors	TinyMCE uses a <code><textarea></code> element, and so the default selector for this field is set to "textarea."
-----------	------------------------------------------------------------------------------------------------------------------------

styles	
displayAccess or	The standard display accessor uses the jQuery functions val or text depending on the tag type. The richTextAccessor returns the HTML contents of the element instead.
editAccessor	The standard edit accessor uses the jQuery functions val or text depending on the tag type. The tinyMCE.viewAccessor function uses the TinyMCE API to access the edit field.
lazyEditView	Loading a rich text editor is expensive, and so lazy loading is enabled.
editModeRende rer	The TinyMCE-specific edit mode renderer uses TinyMCE itself.

Dependencies

The Rich Text Inline Edit dependencies can be met by including the minified `InfusionAll.js` file in the header of the HTML file, along with your choice of rich text editor:

CKEditor Inetgration:

```
<script type="text/javascript" src="http://ckeditor-fluid.appspot.com/ckeditor.js"></script>
<script type="text/javascript" src="InfusionAll.js"></script>
```

TinyMCE Inetgration:

```
<script type="text/javascript" src="http://tinymce-fluid.appspot.com/tiny_mce.js"></script>
<script type="text/javascript" src="InfusionAll.js"></script>
```

Alternatively, the individual file requirements if using CKEditor are:

```
<script type="text/javascript" src="http://ckeditor-fluid.appspot.com/ckeditor.js"></script>
<script type="text/javascript" src="lib/jquery/core/js/jquery.js"></script>
<script type="text/javascript" src="lib/jquery/ui/js/jquery.ui.core.js"></script>
<script type="text/javascript" src="lib/jquery/plugins/tooltip/js/jquery.tooltip.js"></script>
<script type="text/javascript" src="framework/core/js/FluidDocument.js"></script> <!-- New in v1.3 -->
<script type="text/javascript" src="framework/core/js/jquery.keyboard-ally.js"></script>
<script type="text/javascript" src="framework/core/js/Fluid.js"></script>
<script type="text/javascript" src="components/inlineEdit/js/InlineEdit.js"></script>
<script type="text/javascript" src="components/inlineEdit/js/InlineEditIntegrations.js"></script>
```

Otherwise, the individual file requirements if using TinyMCE are:

```
<script type="text/javascript" src="http://tinymce-fluid.appspot.com/tiny_mce.js"></script>
<script type="text/javascript" src="lib/jquery/core/js/jquery.js"></script>
<script type="text/javascript" src="lib/jquery/ui/js/jquery.ui.core.js"></script>
<script type="text/javascript" src="lib/jquery/plugins/tooltip/js/jquery.tooltip.js"></script>
<script type="text/javascript" src="framework/core/js/FluidDocument.js"></script> <!-- New in v1.3 -->
<script type="text/javascript" src="framework/core/js/jquery.keyboard-ally.js"></script>
<script type="text/javascript" src="framework/core/js/Fluid.js"></script>
<script type="text/javascript" src="components/inlineEdit/js/jquery.tinymce.js"></script>
<script type="text/javascript" src="components/inlineEdit/js/InlineEdit.js"></script>
<script type="text/javascript" src="components/inlineEdit/js/InlineEditIntegrations.js"></script>
```