ADOBE AEM: RICH TEXT EDITOR IN CLASSIC UI. ADDING JUSTIFY FUNCTIONALITY.

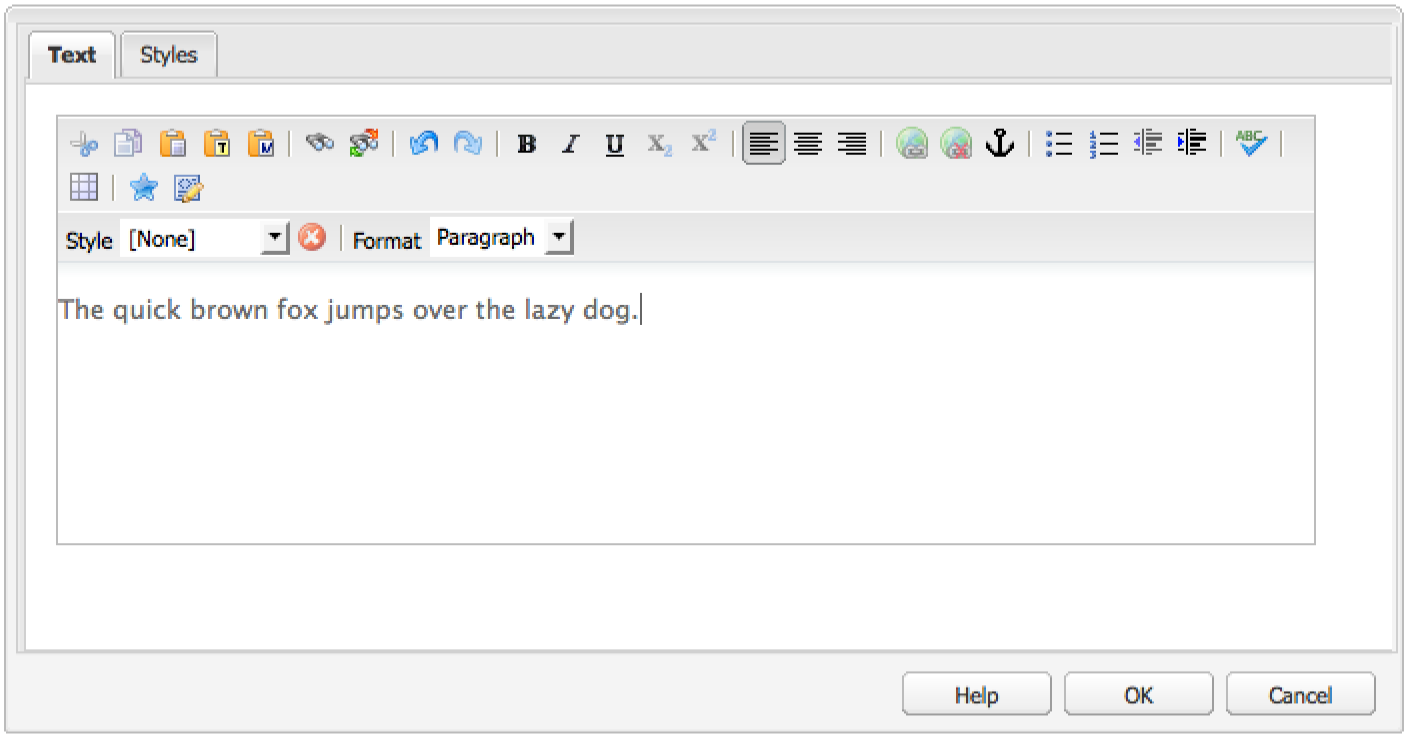
Problem:

The Rich Text Editor is a basic building block for inputting textual content into AEM. It forms the basis of various components, including:

- Text
- Text Image
- Table

RICH TEXT EDITOR - CLASSIC UI

The WYSIWYG editing dialog provides a wide range of functionality:



The quick brown fox jumps over the lazy dog.

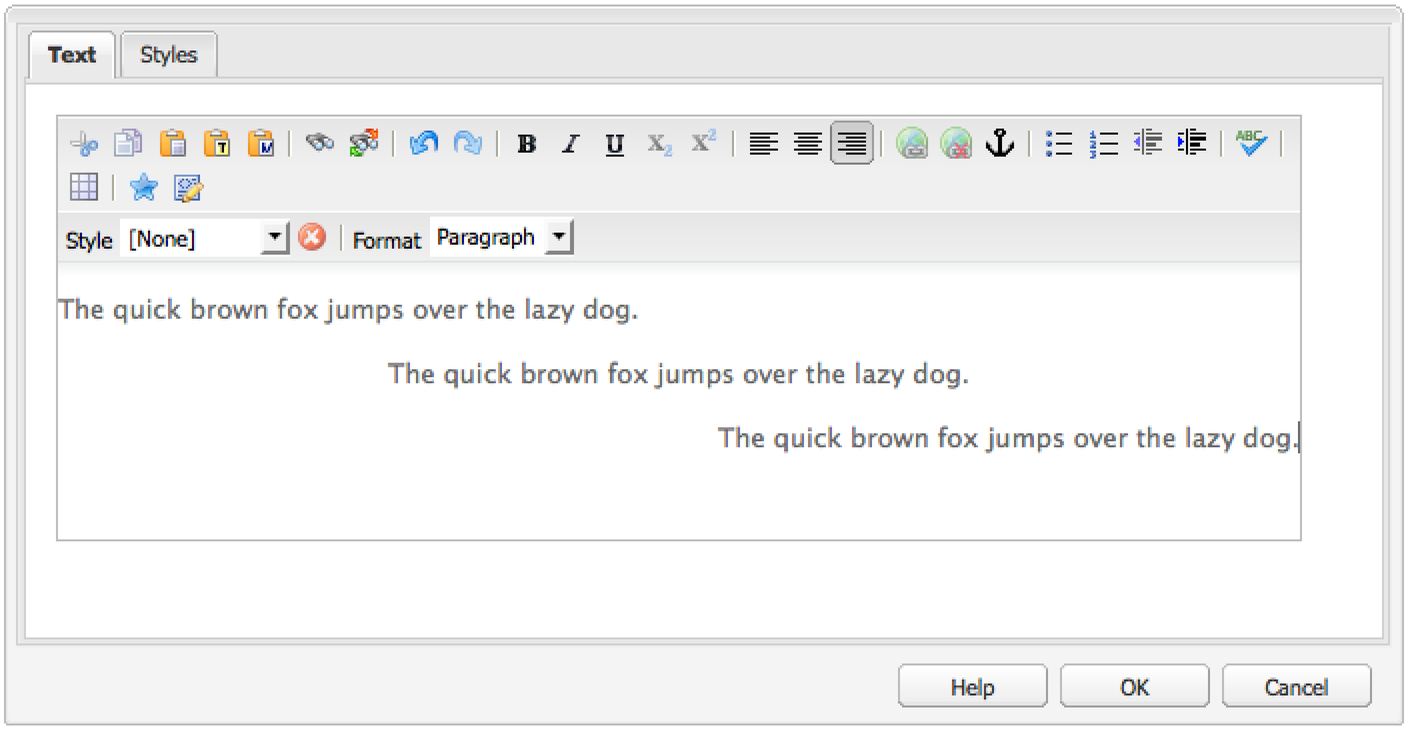
FEATURES OF THE RICH TEXT EDITOR

The Rich Text Editor provides a range of featues, these depend on the configuration of the individual component. The features are available for both the touch-optimized and classic UI.

This publication is focused on the functionality of:

Alignment

Your text can be either left, center or right aligned:

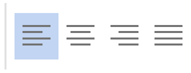


The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

But for comfort working of editor we do not see the full set of tools, like in MS Word for example:



But it is not a problem for AEM developer to add this feature. Let’s see how it is possible.

Initial Data.

- Needed functionality is provided by justify plugin of RTE;
- The source code of justify plugin is located here: /libs/cq/ui/rte/core/plugins/JustifyPlugin.js (just open http://localhost:4502/crx/de/index.jsp)

Let’s start.

Let’s have a look at code:

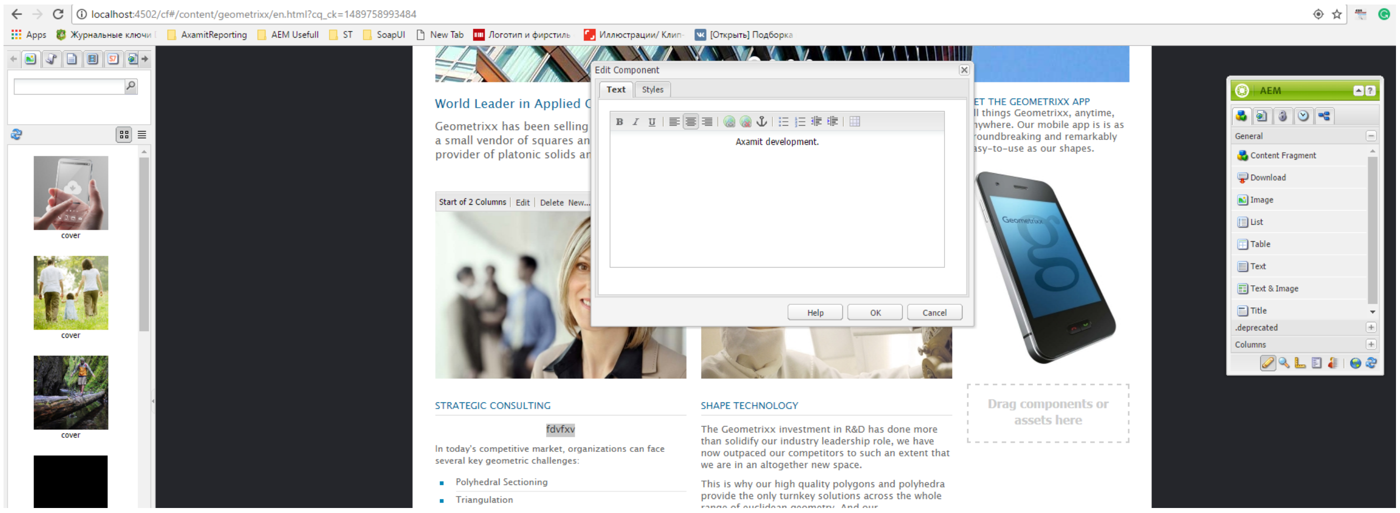
```
CUI.rte.plugins.JustifyPlugin = new Class({

    toString: "JustifyPlugin",

    extend: CUI.rte.plugins.SimpleFormatPlugin,

    _init: function(editorKernel) {
        var plg = CUI.rte.plugins;
        plg.JustifyPlugin.prototype.superClass._init.call(this, editorKernel, "justify",
                plg.Plugin.SORT_JUSTIFY, [ "justifyleft", "justifycenter", "justifyright"]);
    },

    notifyPluginConfig: function(pluginConfig) {
        pluginConfig = pluginConfig || { };
        CUI.rte.Utils.applyDefaults(pluginConfig, {
            "features": "*",
            "tooltips": {
                "justifyleft": {
                    "title": CUI.rte.Utils.i18n("plugins.justify.leftTitle"),
                    "text": CUI.rte.Utils.i18n("plugins.justify.leftText")
                },
                "justifycenter": {
                    "title": CUI.rte.Utils.i18n("plugins.justify.centerTitle"),
                    "text": CUI.rte.Utils.i18n("plugins.justify.centerText")
                },
                "justifyright": {
                    "title": CUI.rte.Utils.i18n("plugins.justify.rightTitle"),
                    "text": CUI.rte.Utils.i18n("plugins.justify.rightText")
                }
            }
        });
        this.config = pluginConfig;
    }

});


// register plugin
CUI.rte.plugins.PluginRegistry.register("justify", CUI.rte.plugins.JustifyPlugin);
```

Here we can see how features of plugin are set:

```
[ "justifyleft", "justifycenter", "justifyright"]
```

Actually the code of plugin is some kind of initial settings of justify plugin (features, localization, registration in browser, etc.)

Let’s add any text component on any available page (for example Geometrixx demo sites). Input some text.

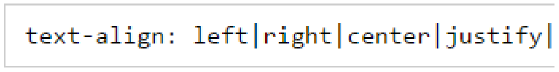


Then playing *justify features* inspect the result html code through browser tools (do not forget to enter preview mode).

```
    <link rel="stylesheet" href="/etc/clientlibs/wcm/foundation/accessibility.css" type="text/css" id="cq-gen397">
    <p style="text-align: center;" id="cq-gen398">Axamit development</p> == $0
  </div>
```

As we see the names of justify plugin’s features transforms into value of *css style* “text-align”.

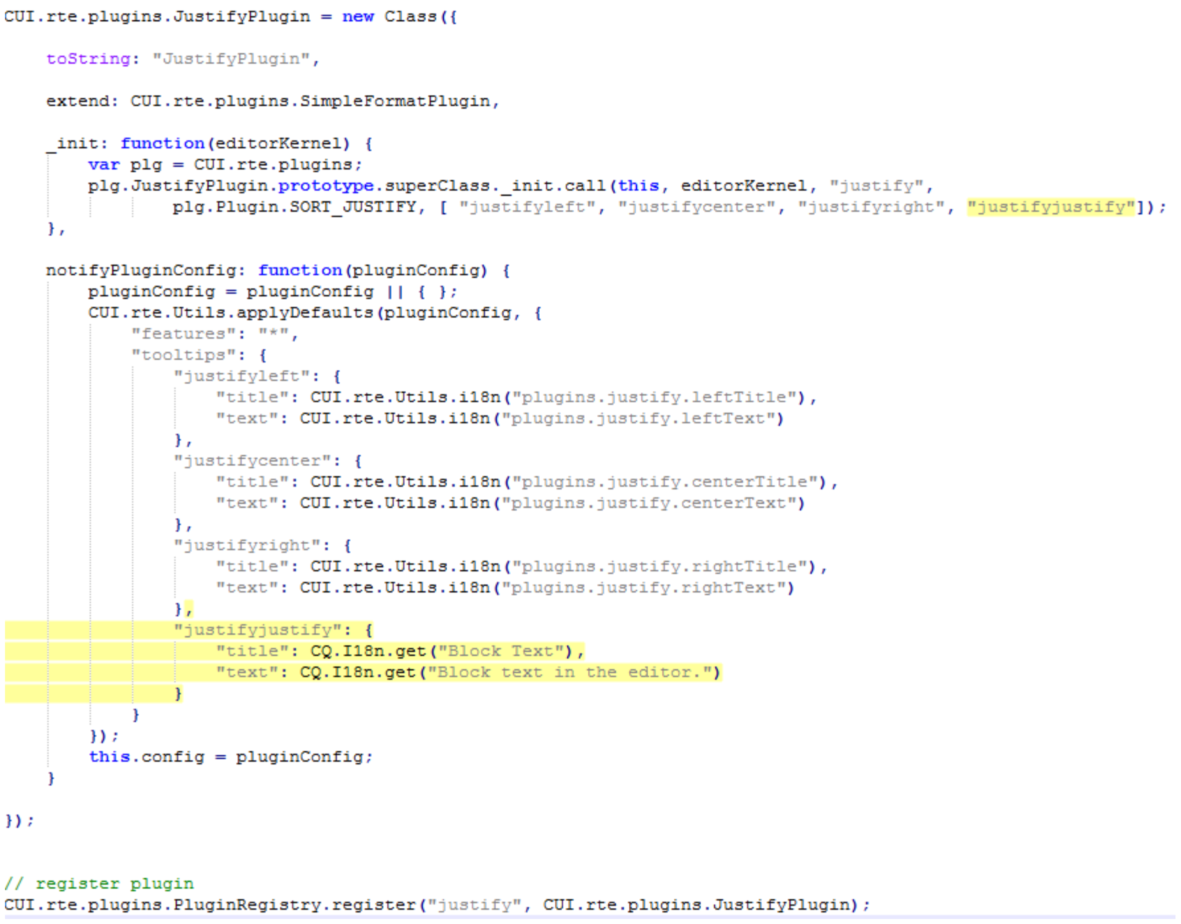
Let’s inspect all the possible values of this style:



```
text-align: left|right|center|justify|
```

Here we can see that *left*, *right* and *center* values are used. *Justify* value is ignored. But if we add *justifyjustify* feature to the plugin we’ll get what we want.

Now let’s skip localization issues and the possible variants of overwriting default justify plugin and move to source code:



```
CUI.rte.plugins.JustifyPlugin = new Class({

    toString: "JustifyPlugin",

    extend: CUI.rte.plugins.SimpleFormatPlugin,

    _init: function(editorKernel) {
        var plg = CUI.rte.plugins;
        plg.JustifyPlugin.prototype.superClass._init.call(this, editorKernel, "justify",
                plg.Plugin.SORT_JUSTIFY, [ "justifyleft", "justifycenter", "justifyright", "justifyjustify"]);
    },

    notifyPluginConfig: function(pluginConfig) {
        pluginConfig = pluginConfig || { };
        CUI.rte.Utils.applyDefaults(pluginConfig, {
            "features": "*",
            "tooltips": {
                "justifyleft": {
                    "title": CUI.rte.Utils.i18n("plugins.justify.leftTitle"),
                    "text": CUI.rte.Utils.i18n("plugins.justify.leftText")
                },
                "justifycenter": {
                    "title": CUI.rte.Utils.i18n("plugins.justify.centerTitle"),
                    "text": CUI.rte.Utils.i18n("plugins.justify.centerText")
                },
                "justifyright": {
                    "title": CUI.rte.Utils.i18n("plugins.justify.rightTitle"),
                    "text": CUI.rte.Utils.i18n("plugins.justify.rightText")
                },
                "justifyjustify": {
                    "title": CQ.I18n.get("Block Text"),
                    "text": CQ.I18n.get("Block text in the editor.")
                }
            }
        });
        this.config = pluginConfig;
    }

});


// register plugin
CUI.rte.plugins.PluginRegistry.register("justify", CUI.rte.plugins.JustifyPlugin);
```

That’s it. Next steps – safely include this class to project.

Full source code of extending justify plugin is included.

Conclusion.

As we see extending AEM functionality is not so complicated task. But our case was facilitated with fact of presence needed value of css style. In some other cases we would make much more work on front-end and back-end side.

Used sources.

1. https://docs.adobe.com/docs/en/aem/6-2/author/page-authoring/rich-text-editor.html
2. https://www.w3schools.com/cssref/pr_text_text-align.asp
3. http://localhost:4502/crx/de/index.jsp#/libs/cq/ui/rte/core/plugins/JustifyPlugin.js