

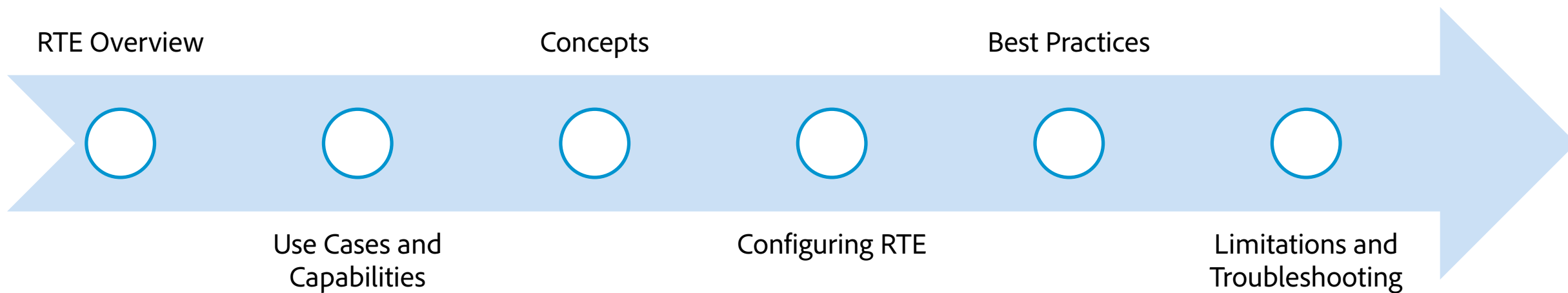


AEM Rich Text Editor (RTE) Deep Dive

Hanish Bansal | Computer Scientist

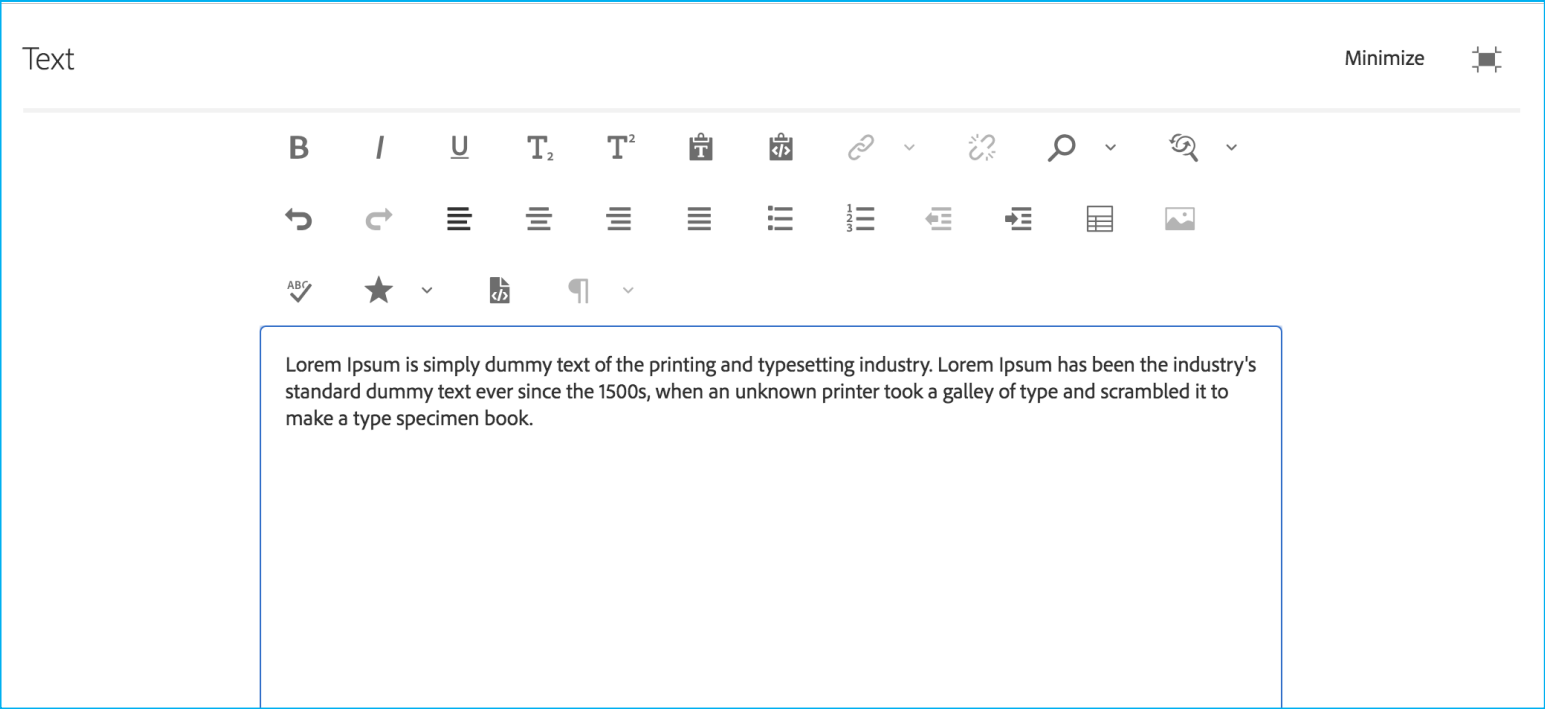


Agenda



Rich Text Editor Overview

- WYSIWIG editing for Authors
- Consistent authoring experience for all authors across browsers



Rich Text Editor Overview

- Rich set of OOTB plugins available for various formatting
- Granular configurations
- Common Configuration for both Classic UI and Touch UI

The screenshot displays the configuration interface for a 'Text' component, divided into two main sections: 'Policy' and 'Properties'.

Policy Section:

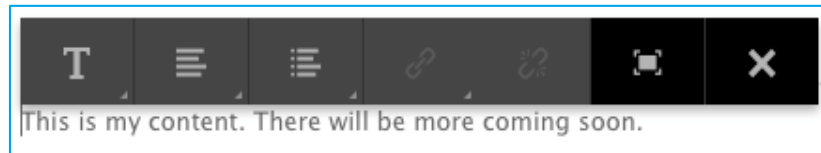
- Policy:** First choose a policy to apply. Policies can be shared across templates.
- Select policy:** A dropdown menu currently shows 'Text policy'.
- Policy Title *:** A text input field containing 'Text policy'.
- Policy Description:** A text area with the placeholder text 'Add a description'.
- Other templates also using the selected policy:** A list showing 'Content Page' with a small thumbnail image.

Properties Section:

- Properties:** Then adjust the settings of the selected policy to configure the component.
- Plugins / Styles:** Two tabs are visible, with 'Styles' currently selected.
- Formatting Options (all checked):**
 - Show table options
 - Show list options (bullet, number, indent, unindent)
 - Show custom alignment options (left aligned, centered, right aligned)
 - Show bold option
 - Show italic option
 - Show underline option
 - Show link insertion options (link, unlink)
 - Show superscript and subscript options
- Paragraph Styles:** A section header for further configuration options.

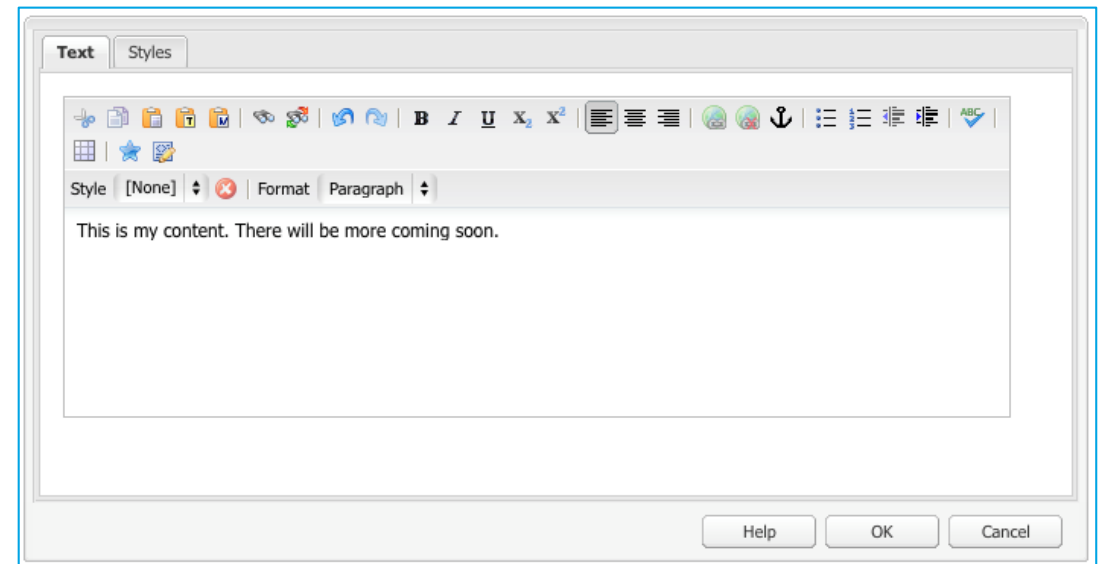
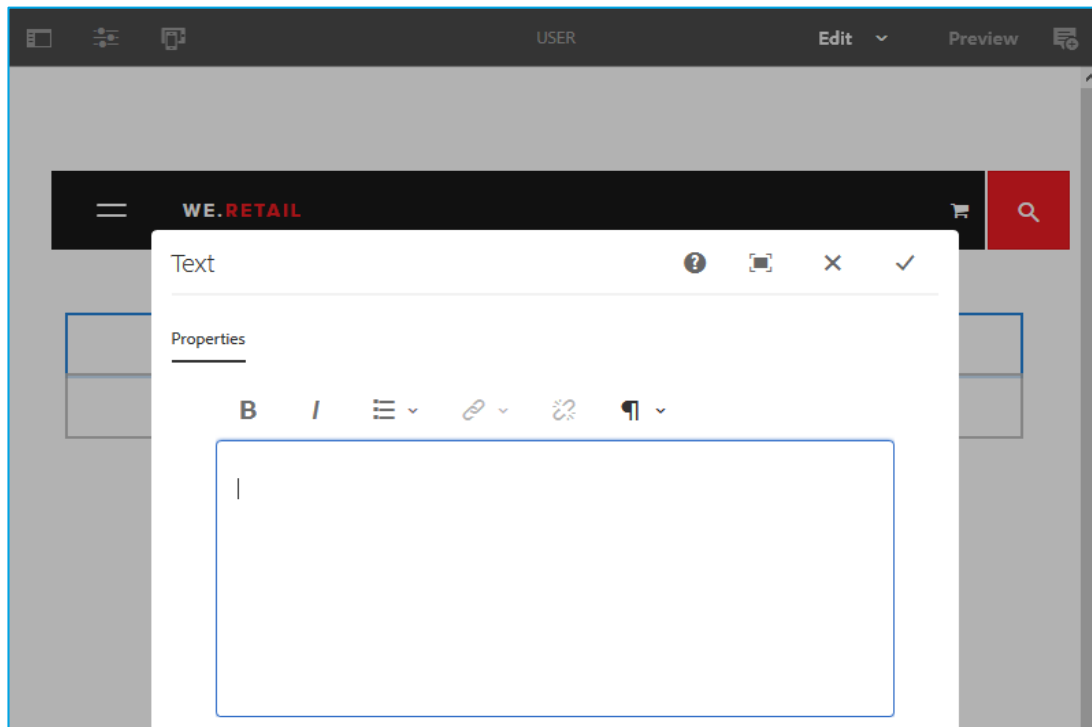
Authoring Use Cases and RTE Capabilities

- Author content in various modes: **Inline**
 - Use case : Quick WYSIWIG editing for authors with simple and commonly used features



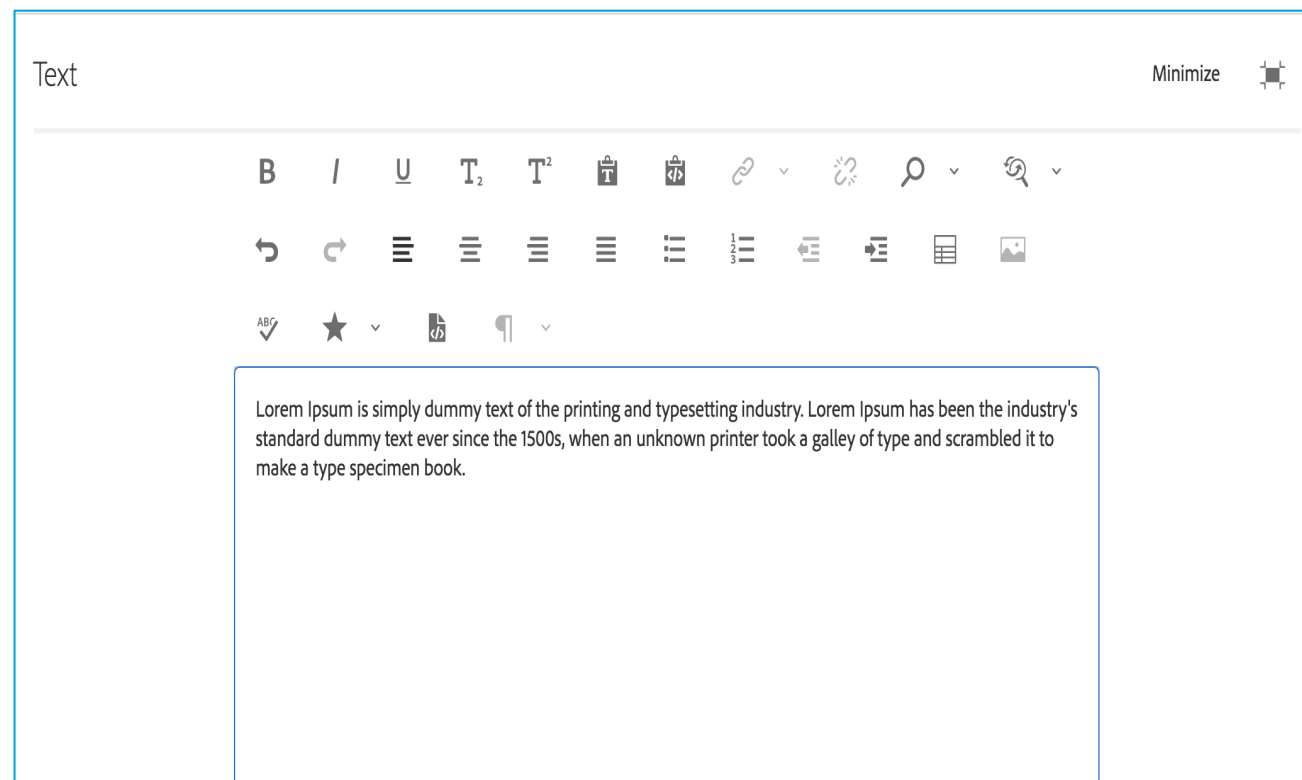
Authoring Use Cases and RTE Capabilities

- Author content in various modes: **Dialog**
 - Use case: Rich Text property field for Component Dialog



Authoring Use Cases and RTE Capabilities

- Author content in various modes:
 - **Fullscreen**
 - **DialogFullscreen**
- Use case: Text Editing using all available features (simple or complex) in a full width screen



Authoring Use Cases and RTE Capabilities

- Available as AEM Core Text component
 - Highly Recommended, available since 6.3
 - Open Sourced
 - RTE Plugins can be enabled/disabled using [Design Dialog](#) for Text Component via Template Editor
- Shipped OOTB via Foundation Text, Text&Image components
 - Deprecated in 6.5
- Can be leveraged in custom AEM components
 - In any mode

Using RTE for different modes in an AEM component

Editing mode	Location for Touch UI	Location for Classic UI	How to include RTE in this mode
Inline	cq:editConfig		editorType = 'text'
Full screen	cq:editConfig	Not applicable	
Dialog	cq:dialog	dialog	<i>Touch UI :</i> sling:resourceType = 'cq/gui/components/authoring/dialog/richtext' useFixedInlineToolbar = true <i>Classic UI :</i> xtype = 'richtext'
Full screen dialog	cq:dialog	Not applicable	

Authoring Use Cases and RTE Capabilities

- AEM Content Fragments leverage RTE

Cancel Save

This content fragment is already referenced on one or more pages. Changes will affect the referenced pages as well. Close

Story

T [] [] [] [] []

Aloha spirits in Northern Norway

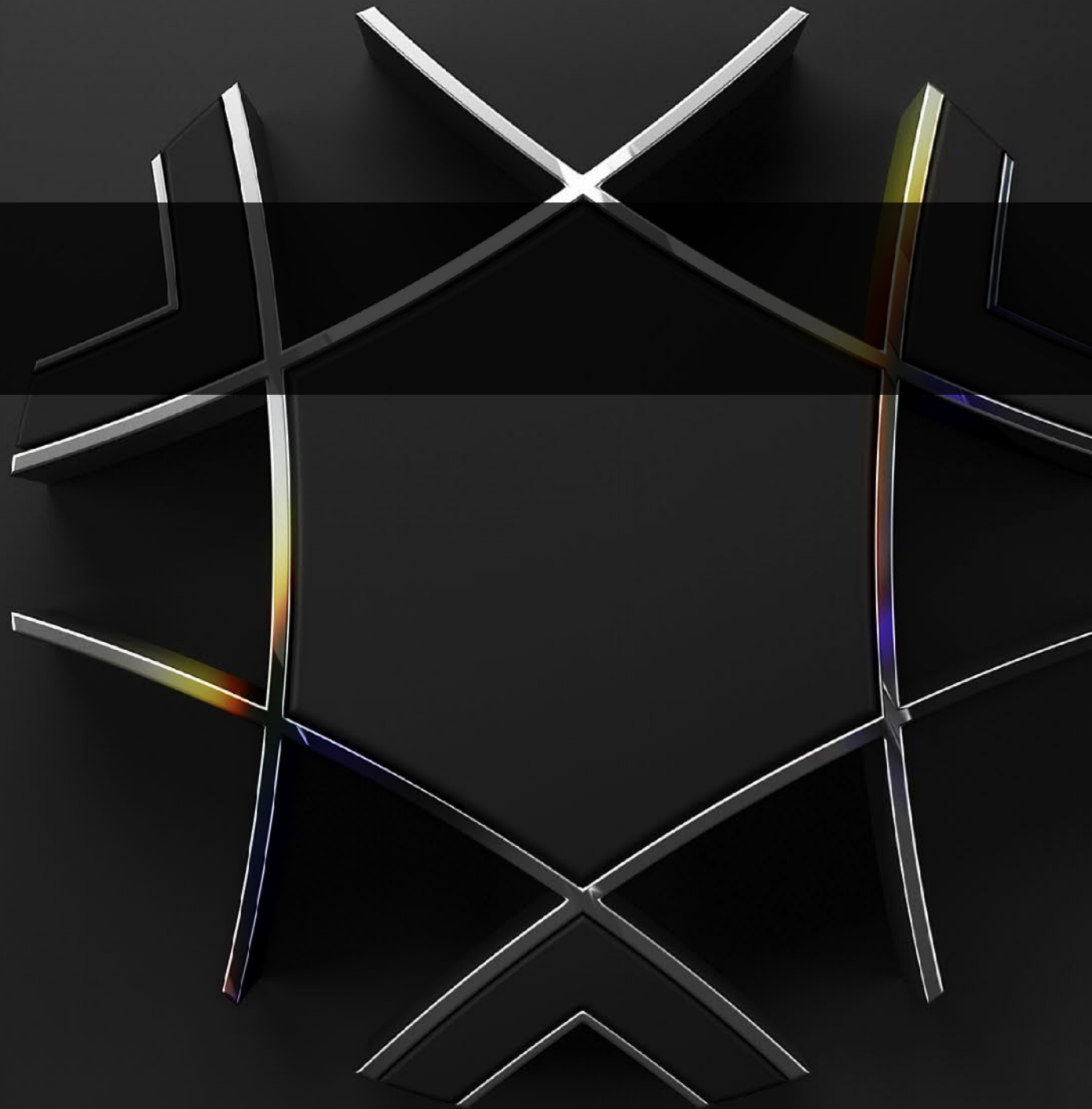
Text: Jacob Wester. Photos: Sofia Sjöberg

Steep mountain sides surround us, like wise trolls from a distant timeline, weathered and worn by long-gone glaciers, green moss now covering the black rock. White sheep forage on steep grass, defying the chilling winds funneled by the deep valley. The subtle hues of the arctic circle are welcoming, comfortable on the eyes. When rare sunrays pierce through the low clouds, the scenery reveals its vibrancy, as the waves reflect a translucent cyan blue before crashing loudly onto white sand. A small but playful groundswell is building, the offshore breeze grooming playful lines down the point, making for welcoming conditions for acclimatizing to cold water and thick neoprene. Knowing it is our last surf before a few days of hard wind, we take full advantage out of every ripple the North Atlantic Ocean sends our way. Tomorrow this place will have taken on a very different, much more hostile appearance.

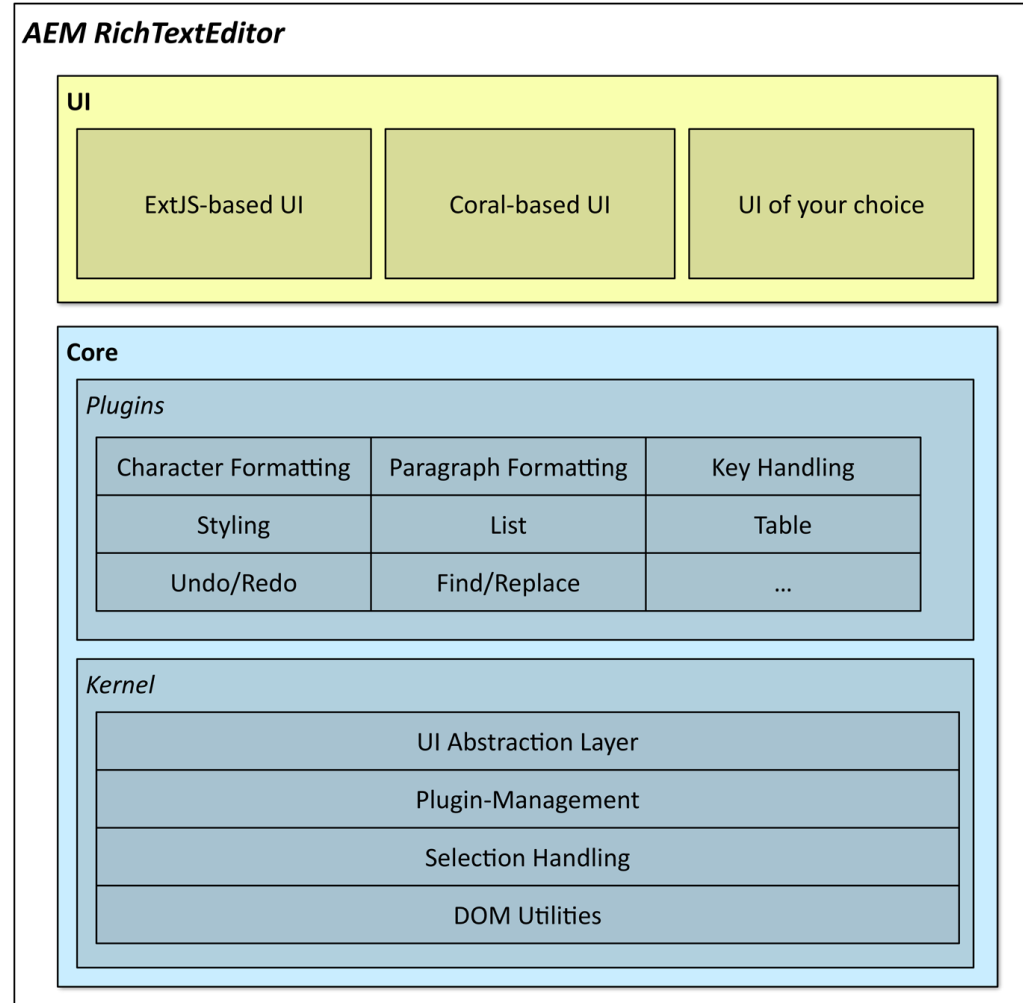
The front is due to arrive any hour now, weather charts calling for anything in between regular bad weather and full arctic storm. In our fishing cabin in the quaint fiord, weather and surf forecast websites are refreshed hourly, for any update or hint on where to find the most manageable conditions. Wetsuits dried and ready, fresh wax coat on the surfboards. Tents are inspected, every seam scrutinized. To properly score in this part of the world, equipment means everything, and any inconsistency can ruin a session, the whole trip even. As our social media devices are put on sleep mode and reading lamps are switched off, the wind grows in intensity, rattling our windows through the night. Every surfer knows the feeling of combined anxiety and excitement, the questions that linger in our minds while falling asleep before a swell. Tomorrow they will be answered; we tell each other before updating the charts just one last time.



Capabilities Demo



Architecture



Architecture

- Classic UI RTE, Touch UI RTE and Content Fragments share the RTE core
- Same plugin configurations work for both UI, offering seamless migration to Touch UI
- Extensible to add new plugins
- Selection Handling wraps native browser functionality for consistent browser UX (even old IE model)
- Dom Handling Utilities work consistently across browsers
- Command-Plugin Architecture allows to enable/disable any single feature or whole of the plugin

CoralUI 2 vs CoralUI 3

- CoralUI3 RTE available since 6.3.
 - Used by default and recommended.
 - All related Clientlibs contain the substring : **rte.coralui3**
- CoralUI2 RTE is also being shipped till 6.5.
 - Can be enabled with simple steps, if required. Not recommended
 - All related Clientlibs contain the substring : **rte.coralui2**
- Seamless migration of configurations and customizations from Coral2 RTE to Coral3 RTE
 - As long as, no private css classes of RTE are used.

RTE clientlibs

Clientlib	Path	Purpose
rte.coralui3	/libs/clientlibs/granite/richtext	RTE Core and Coral UI
cq.authoring.rte.coralui3	/libs/cq/gui/components/authoring/rte/coralui3	AEM Component Authoring for RTE (common to both inline-editing and dialog)
cq.authoring.dialog.rte.coralui3	/libs/cq/gui/components/authoring/dialog/richtext/clientlibs/rte/coralui3	RTE in Authoring Dialog specific code
cq.authoring.editor.core.inlineediting.rte.coralui3	/libs/cq/gui/components/authoring/editors/clientlibs/core/inlineediting/rte/coralui3	Text Inline-editing
cq.rte	/libs/cq/ui/rte	RTE Core and ExtJS UI for UI

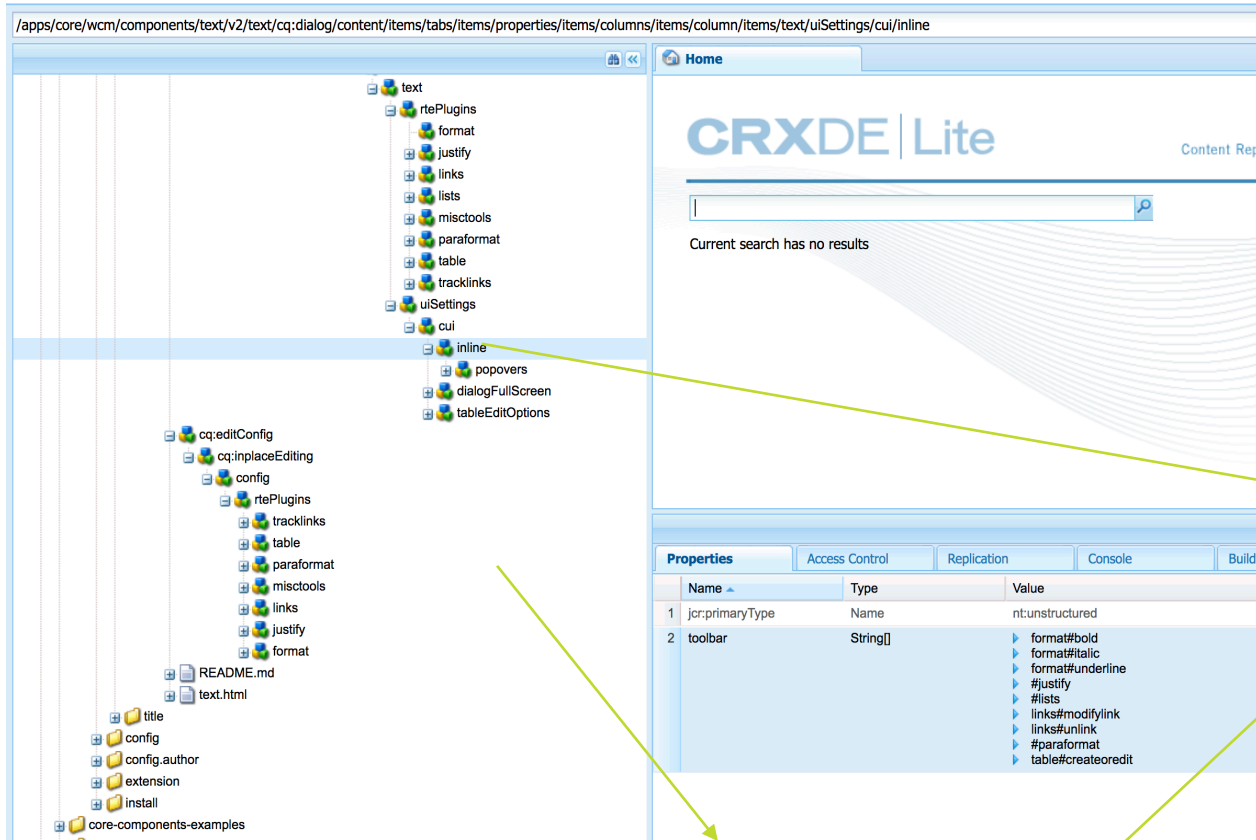
UI Config

To add a button/feature on a particular RTE toolbar–

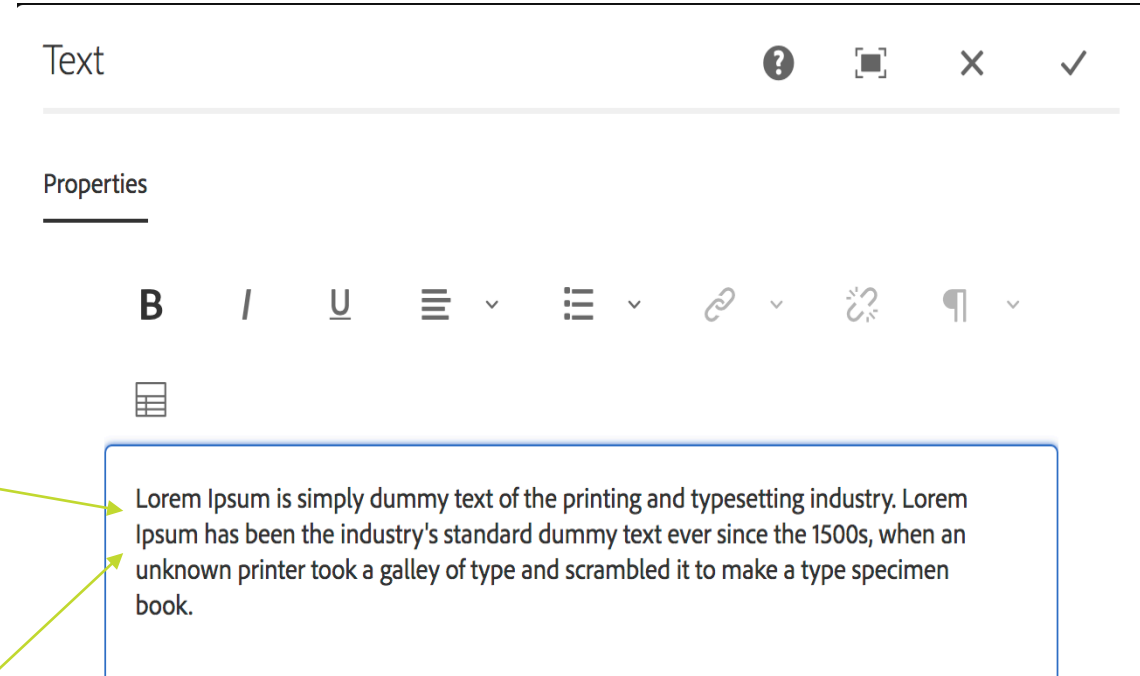
- Enable the Plugin corresponding to that feature for the RTE toolbar
 - Use Design Dialog for Core Text component via Template Editor
 - Edit using crx/de for other component
- Add that button in an ordered list for that toolbar under uiSettings

Note: *Each toolbar can be configured differently to have any buttons on it (via uiSettings)*

UI Config



Format plugin enabled
'format#bold' - first property on inline toolbar for dialog



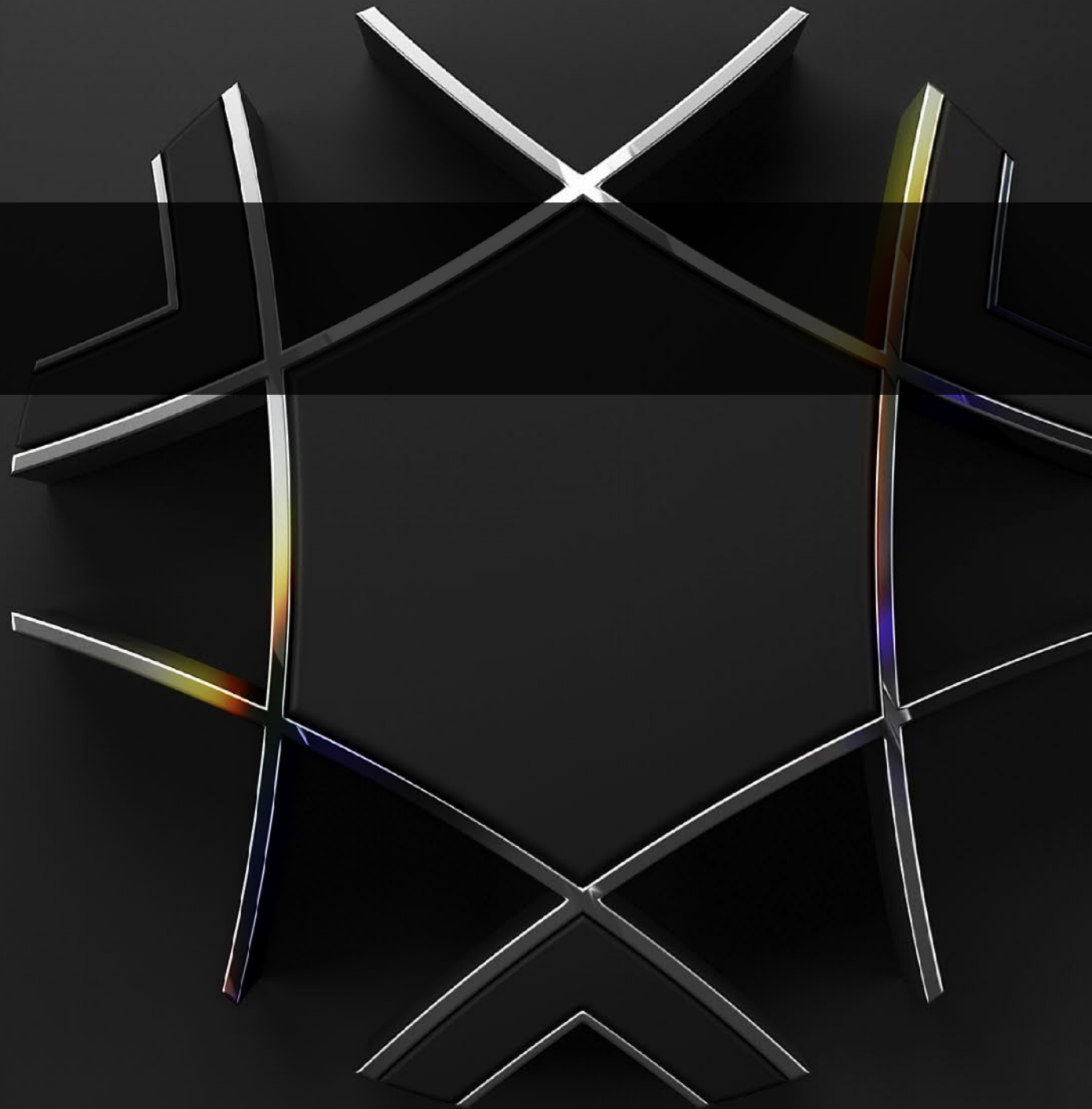
Bold button visible as first button on RTE toolbar

Core Configs

- pasteRules
 - pasteRules.table.ignoreMode
 - pasteRules.list.allow
 - ...
- htmlRules
 - htmlRules.links.protocol
 - htmlRules.doctype.tagBlacklist
 - htmlRules.blockHandling
 - ...



Configurations Demo

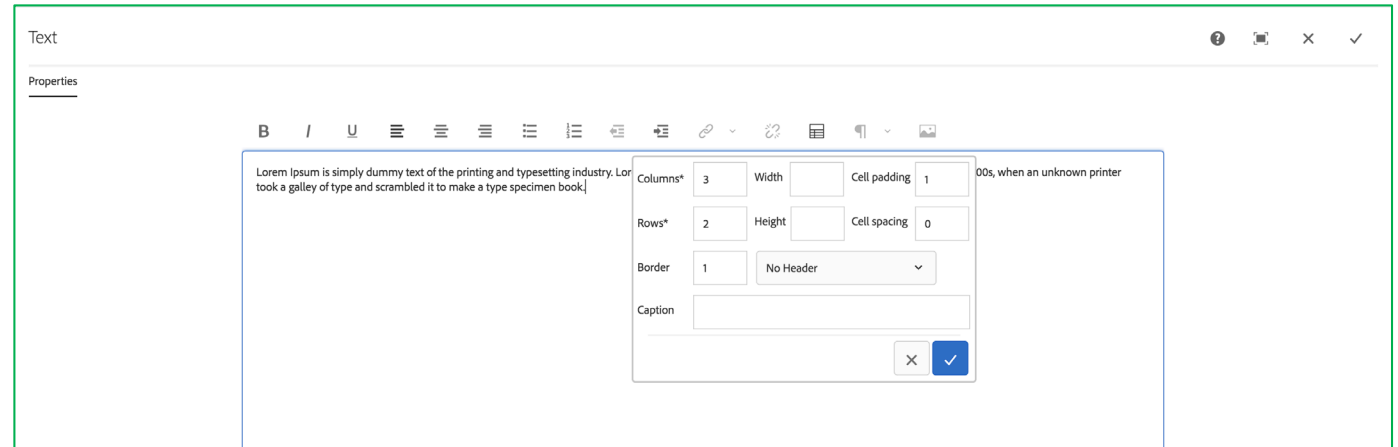
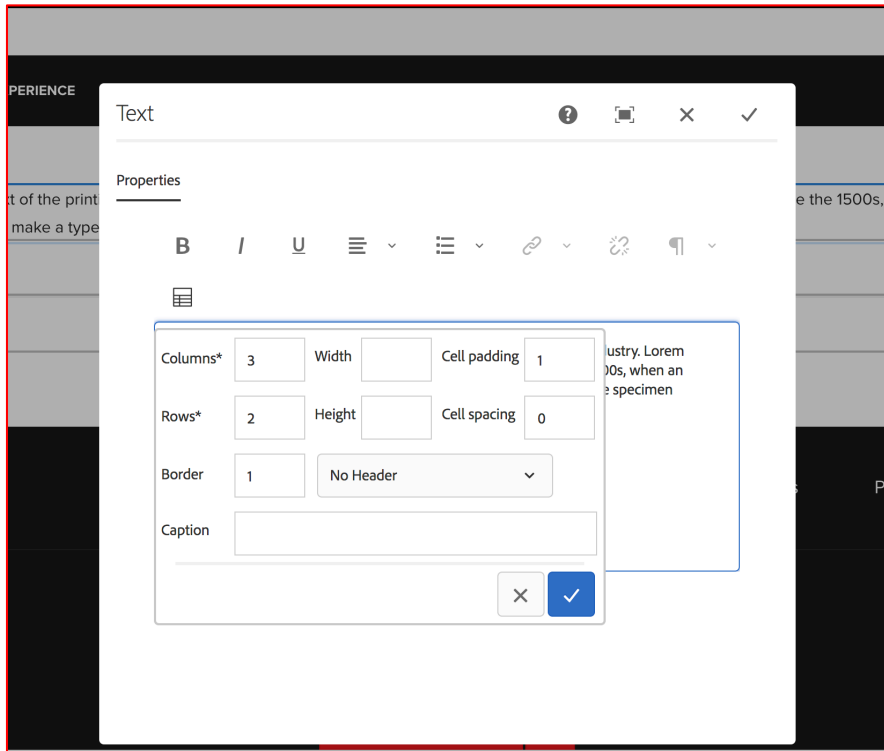


Latest improvements

- 6.3
 - dialogFullscreen toolbar mode introduced
 - Support for “uneditable” tags like comments, iframes etc. added
 - Doc revamp and re-structuring
- 6.4
 - Better UI support for mandatory RTE field inside a dialog
 - Improved support inside multiframe – customStart, required field validation, UI fixes etc.
- 6.5
 - RTE plugin configuration via Design Dialog applies to both Dialog and Inline editing
 - Justify Alignment support added
 - Customize only required toolbars. Customization now applies on top of default settings

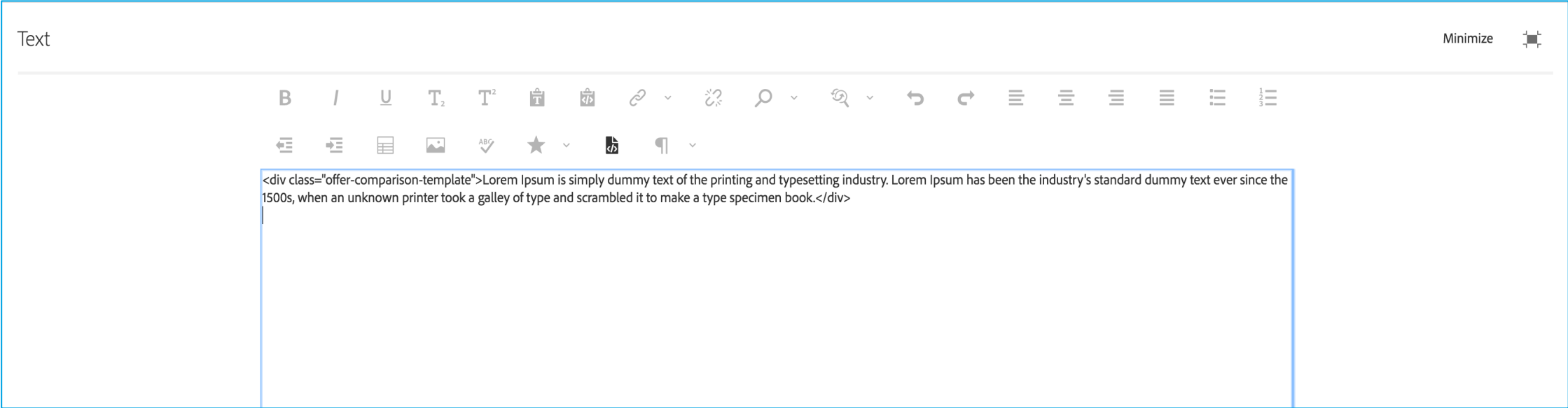
Best Practices

- Use Inline and floating Dialog mode only for simple plugins with No or small popovers. For other plugins, Use Fullscreen mode.



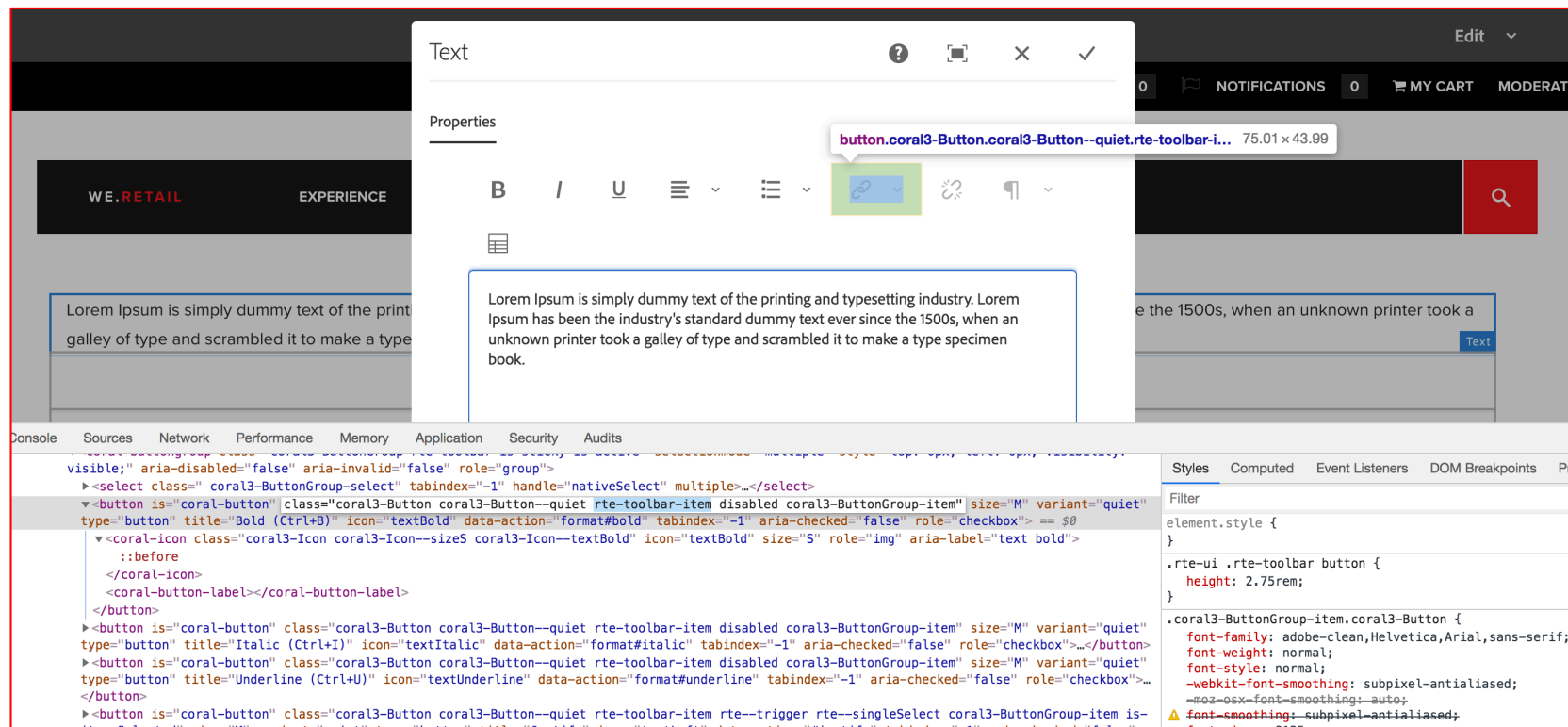
Best Practices

- Avoid directly writing HTML in source-edit mode as much as possible. Use plugins to format your text.



Best Practices

- Don't refer to any private css classes of RTE in your custom plugins' code
 - Private classes start with 'rte-' e.g. **rte-toolbar-item**
- Don't switch to Coral2 RTE. Fix custom plugin code to remove usage of private classes



Best Practices

- Don't embed RTE in page properties console, scaffolding or foundation-forms. Only works in Component Authoring Dialogs
- While pasting formatted text, use Paste-from-Word feature and only use text copied from MS Word.
 - After pasting text, fix formatting using RTE plugins, if needed
 - For other sources, paste it first in MS Word and then copy from there.
- Use latest stable Browser versions

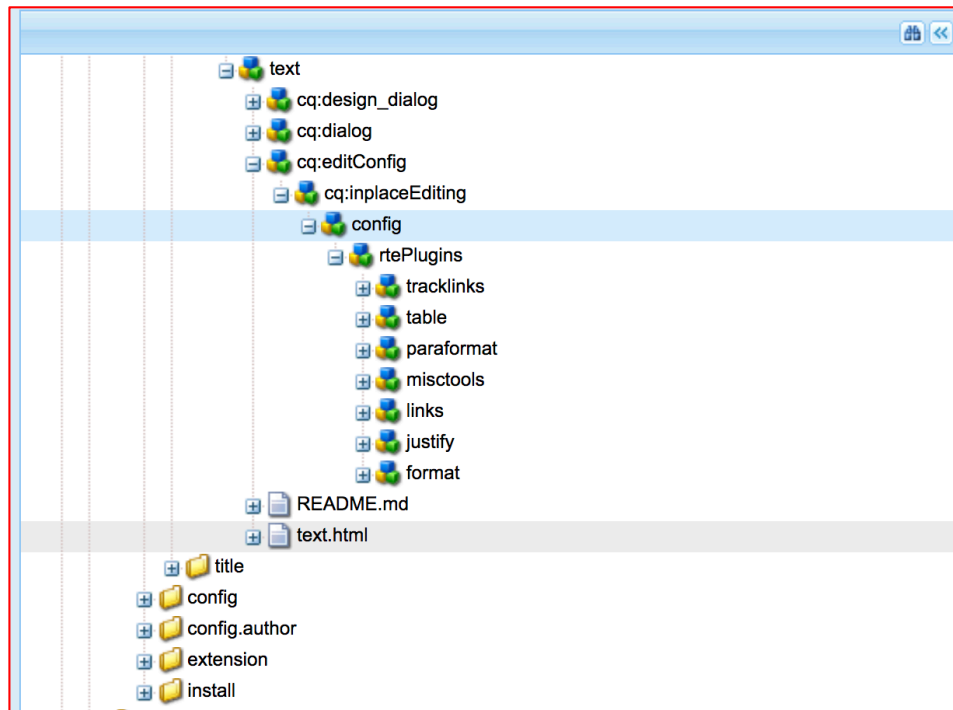
Known Limitations (as of 6.5)

- No Table and Cell styles in Touch UI
- Link Plugin Path Picker doesn't open with the last saved path selected
- Hybrid Devices are not supported

Troubleshooting of Common scenarios

Q: UI Configurations for Plugins are reflected on RTE toolbars for administrators, but not content-authors

A: Ensure that the configuration node for RTE is not named as 'config'. Use 'configPath' property



cq:inplaceEditing

The `cq:inplaceEditing` node (node type `cq:InplaceEditingConfig`) defines an inplace editing configuration for the component. It can have the following properties:

Property Name	Property Value
<code>active</code>	(boolean) True to enable the inplace editing of the component.
<code>configPath</code>	(String) Path of the editor configuration. The configuration can be specified by a configuration node.
<code>editorType</code>	(String) Editor type. The available types are: <ul style="list-style-type: none"><code>plaintext</code>: to be used for non HTML content.<code>title</code>: is an enhanced plaintext editor that converts graphical titles into a plaintext before editing begins. Used by the Geometrix title component.<code>text</code>: to be used for HTML content (uses the Rich Text Editor).

The following configuration enables the inplace editing of the component and defines `plaintext` as the editor type:

```
1 <cq:inplaceEditing
2   jcr:primaryType="cq:InplaceEditingConfig"
3   active="{Boolean}true"
4   editorType="plaintext"/>
```

Troubleshooting of Common scenarios

Q: RTE not working correctly inside Multifield

A:

- Check that your multifield implementation code and node structure are correct.
- Use Coral3 Multifield RTE.
- Don't mix Coral2 and Coral3 RTs in your component dialogs.

PathField and RTE (Rich Text Editor)

This is an example of simple [RTE configuration](#):

```
+ multifield
- sling:resourceType = "granite/ui/components/coral/foundation/form/multifield"
- composite = true
+ field
- sling:resourceType = "granite/ui/components/coral/foundation/container"
- name = "./multidata"
+ items
+ title
- sling:resourceType = "granite/ui/components/coral/foundation/form/textfield"
- name = "title"
+ path
- sling:resourceType = "granite/ui/components/coral/foundation/form/pathfield"
- name = "path"
+ description
- sling:resourceType = "cq/gui/components/authoring/dialog/richtext"
- useFixedInlineToolbar = true
- name = "description"
+ rtePlugins
+ format
- features = ["bold", "italic"]
+ justify
- features = "-"
+ links
- features = ["modifylink", "unlink"]
+ lists
- features = "*"
+ paraformat
- features = "*"
+ tracklinks
- features = "*"
```

Troubleshooting of Common scenarios

Q: The html I added in source-edit mode in RTE is lost, when I save and re-open RTE for editing

A: Verify if you added a XSS prone HTML.

Step1 : Check content node in crx/de if it contains the same html that you saved

Step2 : If yes, Open your page in Preview mode and check Page Source.

If you don't see expected html, it's removed due to XSS Antisamy Rules in AEM, not by RTE.

The XSS Config exists at : `/libs/cq/xssprotection/config.xml`

It is not recommended to modify this config for security concerns.



Troubleshooting Demo



Resources

- Configuring RTE UI :
 - <https://helpx.adobe.com/in/experience-manager/6-4/sites/administering/using/rich-text-editor.html>
- Configurations for OOTB RTE Plugins
 - <https://helpx.adobe.com/in/experience-manager/6-3/sites/administering/using/configure-rich-text-editor-plug-ins.html>
- ExtJS Widget API Doc (provides Core RTE Config JSDoc)
 - <https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/reference-materials/widgets-api/index.html>
- Core Text Component
 - <https://docs.adobe.com/content/help/en/experience-manager-core-components/using/components/text.html>
- Using RTE in Multifield
 - https://helpx.adobe.com/experience-manager/6-5/sites/developing/using/reference-materials/granite-ui/api/jcr_root/libs/granite/ui/components/coral/foundation/form/multifield/index.html



Q&A





Adobe