# Edit Config

* configures the edit behavior of a component.
* includes attributes such as actions available for the component, characteristics of the inplace editor and the listeners related to events on the component.
* This node allows developers to configure how a component is edited, add or remove functionality from the editor and generally tailor the authoring experience for component.
* One of the nice features about the cq:EditConfig is you can use it to get components to show up in the sidekick which do not have a dialog(instead creating an empty dialog).

# editConfig Node Properties

Cq:editConfig has below properties

|  |  |
| --- | --- |
| Properties | Description |
| Cq:dropTargets | Declares that the component accepts draggin and dropping  In Classic - Mulitple drop target is allowed  In Touch - Single drop target is allowed |
| cq:actionConfigs(CLASSIC only) | defines a list of new actions that are appended to the cq:actions list. |
| cq:formParameters | defines additional parameters that are added to the dialog form. |
| cq:inplaceEditing(node type cq:InplaceEditingConfig) | defines an inplace editing configuration for the component. |
| cq:listeners(node type cq:EditListenersConfig) | defines what happens before or after an action occurs on the component. |
| cq:actions(string array) | Defines series of actions to be performed on component (In below table) |

# Cq:actions

|  |  |
| --- | --- |
| **Property Value** | **Description** |
| **text:<some text>** | Displays the static text value <some text> Only visible in classic UI. The touch-enabled UI does not display actions in a contextual menu, so this is not applicable. |
| - | Adds a spacer. Only visible in classic UI. The touch-enabled UI does not display actions in a contextual menu, so this is not applicable. |
| **edit** | Adds a button to edit the component. |
| **delete** | Adds a button to delete the component |
| **insert** | Adds a button to insert a new component before the current one |
| **copymove** | Adds a button to copy and cut the component. |

# Cq:layout (only classic UI)

|  |  |
| --- | --- |
| **Property Value** | **Description** |
| **rollover** | Default value. The component edition is accessible "on mouse over" through clicks and/or context menu. For advanced use, note that the corresponding client side object is: **CQ.wcm.EditRollover**. |
| **editbar** | The component edition is accessible through a toolbar. For advanced use, note that the corresponding client side object is: **CQ.wcm.EditBar**. |
| **auto** | The choice is left to the client side code. |

# cq:emptyText

The **cq:emptyText** property (**String**) defines text that is displayed when no visual content is present. It defaults to: **Drag components or assets here**.

# dialogLayout

The **dialogLayout** property defines how a dialog should open by default.

* A value of **fullscreen** opens the dialog in full screen.
* An empty value or absence of the property defaults to opening the dialog normally.
* Note that the user can always toggle the fullscreen mode within the dialog.

Cq:editConfig has below child nodes

1. Cq:dropTargets
2. Cq:inlineEditing
3. Cq:listeners

# 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** |
| **active** | (**boolean**) True to enable the inplace editing of the component. |
| **configPath** | (**String**) Path of the editor configuration. The configuration can be specified by a configuration node. |
| **editorType** | (**String**) Editor type. The available types are:   * plaintext: to be used for non HTML content. * title: is an enhanced plaintext editor that converts graphical titles into a plaintext before editing begins. Used by the Geometrixx title component. * text: to be used for HTML content (uses the Rich Text Editor). |

Child nodes of particular interest include:

* **cq:editConfig** (**cq:EditConfig**) - this controls visual aspects; for example, it can define the appearance of a bar or widget, or can add customized controls
* **cq:childEditConfig** (**cq:EditConfig**) - this controls the visual aspects for child components that do not have their own definitions

The cq:EditConfig allows for configuring many important features of the component's editing experience including:

* Dialog display behavior
* Edit bar or hover appearance
* Editor items
* Listeners
* Advanced features such as:
  + Drop targets
  + Additional actions
  + InPlace editing

# How a editConfig looks in xml

<jcr:root xmlns:cq="http://www.day.com/jcr/cq/1.0" xmlns:jcr="http://www.jcp.org/jcr/1.0"

cq:actions="[text:My Component,-,edit,delete,insert,copymove]"

cq:dialogMode="floating"

cq:layout="editbar"

jcr:primaryType="cq:EditConfig">

</jcr:root>

# Listeners

<jcr:root xmlns:cq="http://www.day.com/jcr/cq/1.0" xmlns:jcr="http://www.jcp.org/jcr/1.0"

cq:actions="[text:My Component,-,edit,delete,insert,copymove]"

cq:dialogMode="floating"

cq:layout="editbar"

jcr:primaryType="cq:EditConfig">

<cq:listeners

jcr:primaryType="cq:EditListenersConfig"

afteredit="REFRESH\_PAGE"/>

</jcr:root>

Below are the available events for listeners

|  |  |
| --- | --- |
| **Properties** | **Description of each Property** |
| beforedelete | The handler is triggered before the component is removed. |
| beforeedit | The handler is triggered before the component is edited. |
| beforecopy | The handler is triggered before the component is copied. |
| beforeinsert | The handler is triggered before the component is inserted. |
| beforechildinsert | The handler is triggered before the component is inserted.  Only operational for the touch-enabled UI. |
| beforemove | The handler is triggered before the component is moved. |
| afterdelete | The handler is triggered after the component is deleted. |
| afteredit | The handler is triggered after the component is edited. |
| aftercopy | The handler is triggered after the component is copied. |
| afterinsert | The handler is triggered after the component is inserted. |
| afterchildinsert | The handler is triggered after the component is inserted inside another component (containers only). |
| aftermove | The handler is triggered after the component is moved.. |

# Useful Links:

<http://www.sgaemsolutions.com/2019/01/ootbcustom-cqlisteners-in-cqeditconfig.html>

<http://www.sgaemsolutions.com/2019/01/cqdroptagets-in-cqeditconfig-node-in-aem.html>

<http://www.sgaemsolutions.com/2019/02/cqinplaceediting-in-cqeditconfig-node_7.html>

<https://helpx.adobe.com/experience-manager/using/configure-touchui-rte.html>

[https://helpx.adobe.com/in/experience-manager/6-3/sites/developing/using/components-basics.html#EditBehavior](https://helpx.adobe.com/in/experience-manager/6-3/sites/developing/using/components-basics.html%23EditBehavior)