



USING ALLOY EDITOR IN CUSTOM APPLICATIONS

Copyright ©2017 Liferay, Inc.

All Rights Reserved.

No material may be reproduced electronically or in print, distributed, copied, sold, resold, or otherwise exploited for any commercial purpose without express written consent of Liferay, Inc.

IMPLEMENTING ALLOY EDITOR INTO YOUR APPLICATION

- ❖ The AlloyEditor can easily be implemented into custom applications using the `<liferay-ui:input-editor />` taglib.
- ❖ Here is an example:

```
<%@ taglib prefix="liferay-ui" uri="http://liferay.com/tld/ui" %>

<div class="alloy-editor-container">
    <liferay-ui:input-editor contents="Default Content"
        cssClass="my-alloy-editor" editorName="alloyeditor"
        name="myAlloyEditor" placeholder="description" showSource="true" />
</div>
```

PASSING JAVASCRIPT FUNCTIONS

- ❖ You can pass JavaScript functions through `onBlurMethod`, `onChangeMethod`, `onFocusMethod`, and `onInitMethod`.

```
<%@ taglib prefix="lui" uri="http://liferay.com/tld/lui" %>
<%@ taglib prefix="liferay-ui" uri="http://liferay.com/tld/ui" %>
<div class="alloy-editor-container">
    <liferay-ui:input-editor contents="Default Content" cssClass="
    my-alloy-editor" editorName="alloyeditor" name="myAlloyEditor"
    onInitMethod="OnDescriptionEditorInit" placeholder="description"
    showSource="true" />
</div>
<lui:script>
    function <portlet:namespace />OnDescriptionEditorInit() {
        <c:if test="<%= !customAbstract %>">
            document.getElementById('<portlet:namespace />myAlloyEditor').
            setAttribute('contenteditable', false);
        </c:if>
    }
</lui:script>
```

ALLOY EDITOR ATTRIBUTES (I)

- ❖ As you can see, Liferay has made it easy to make use of the new AlloyEditor.
- ❖ Below is an overview of the attributes for the `liferay-ui:input-editor` taglib that are used with AlloyEditor:
- ❖ `<liferay-ui:input-editor />`
 - ❖ `autoCreate (java.lang.String)`: A string boolean to determine whether or not to show HTML edit view of editor initially
 - ❖ `contents (java.lang.String)`
 - ❖ `contentsLanguageId (java.lang.String)`: The ID of a language for the input editor's text
 - ❖ `cssClass (java.lang.String)`: A CSS class for styling the component
 - ❖ `data (java.util.Map)`: data that can be used as the editorConfig
 - ❖ `editorName (java.lang.String)`
 - ❖ `name (java.lang.String)`: A name for the input editor. The default value is *editor*.

ALLOY EDITOR ATTRIBUTES (II)

- ❖ `<liferay-ui:input-editor />`
 - ❖ `onBlurMethod (java.lang.String)`: A function to be called when the input editor loses focus
 - ❖ `onChangeMethod (java.lang.String)`: A function to be called on a change in the input editor
 - ❖ `onFocusMethod (java.lang.String)`: A function to be called when the input editor gets focus
 - ❖ `onInitMethod (java.lang.String)`: A function to be called when the input editor initializes
 - ❖ `placeholder (java.lang.String)`: Placeholder text to display in the input editor
 - ❖ `showSource (java.lang.String)`: Whether to enable editing the HTML source code of the content. The default value is *true*.
 - ❖ `skipEditorLoading (boolean)`: Whether to skip loading resources necessary for the CKEditor. The default value is *false*.

Notes: