

Classic UI Dynamic Dialogs

Pavel Nosau



Agenda

Classic UI Dialogs

Event handling on Fields

Sample

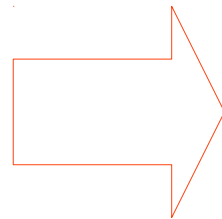
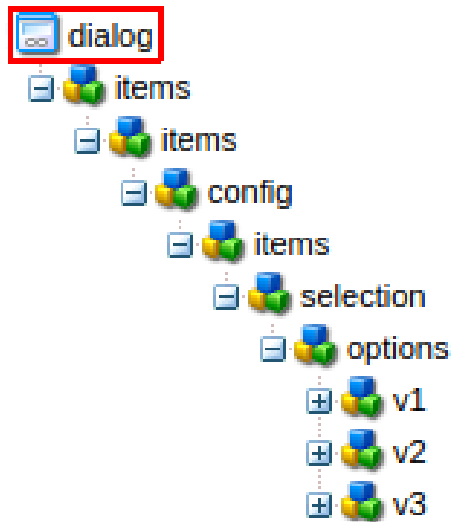
Tips and Tricks



COCAINE INFORMATION
so much information!

Classic UI Dialogs

Classic UI Dialogs



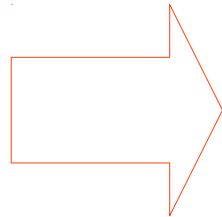
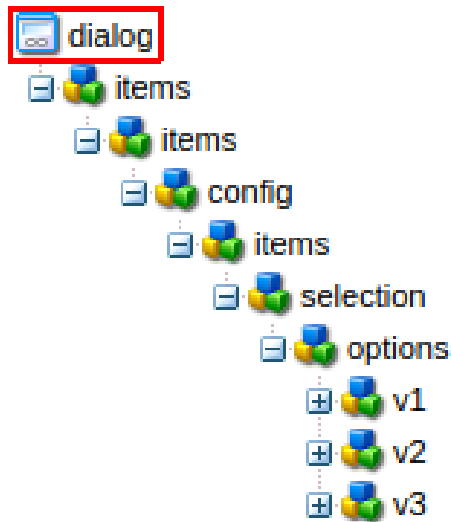
dialog node

ExtJS widgets / xtypes

rendered on client

1	jcr:primaryType	N... cq:Dialog
2	xtype	S... dialog

Classic UI Dialogs

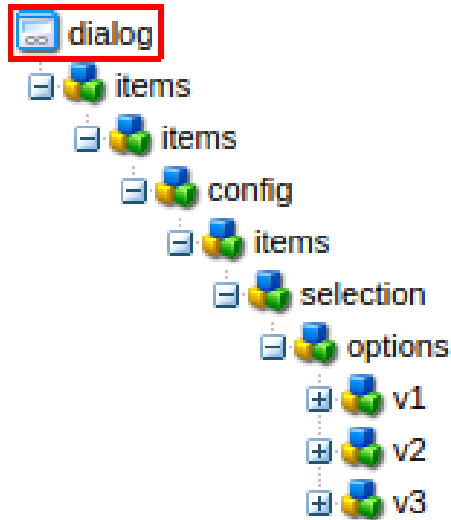


dialog node

ExtJS widgets / xtypes

rendered on client

Classic UI Dialogs



dialog node

ExtJS widgets / xtypes

rendered on client

Classic UI Dialogs

```
<script type="text/javascript">
CQ.WCM.edit({"path":"/content/dynamic-dialogs/test_by/en/homepage/jcr:content/par/dynamictabs", "dialog":"/apps/dynamic-
dialogs/components/dynamicTabs/dialog", "type":"dynamic-dialogs/components/dynamicTabs", "csp":"homepage|page/par|parsys/dynamicTabs|parbase", "editConfig":
{"xtype":"editbar", "inlineEditing":CQ.wcm.EditBase.INLINE_MODE_NEVER, "actions":[{"xtype":"tbtext", "text":" Dynamic
Tabs"}, CQ.wcm.EditBase.EDIT, CQ.wcm.EditBase.DELETE], "disableTargeting":true}});
</script>
```


Classic UI Dialogs

```
<script type="text/javascript">
CQ.WCM.edit({"path":"/content/dynamic-dialogs/test_by/en/homepage/jcr:content/par/dynamictabs", "dialog":"/apps/dynamic-
dialogs/components/dynamicTabs/dialog", "type":"dynamic-dialogs/components/dynamicTabs", "csp":"homepage|page/par|parsys/dynamicTabs|parbase", "editConfig":
{"xtype":"editbar", "inlineEditing":CQ.wcm.EditBase.INLINE_MODE_NEVER, "actions":[{"xtype":"tbtext", "text":" Dynamic
Tabs"}, CQ.wcm.EditBase.EDIT, CQ.wcm.EditBase.DELETE], "disableTargeting":true}});
</script>
```

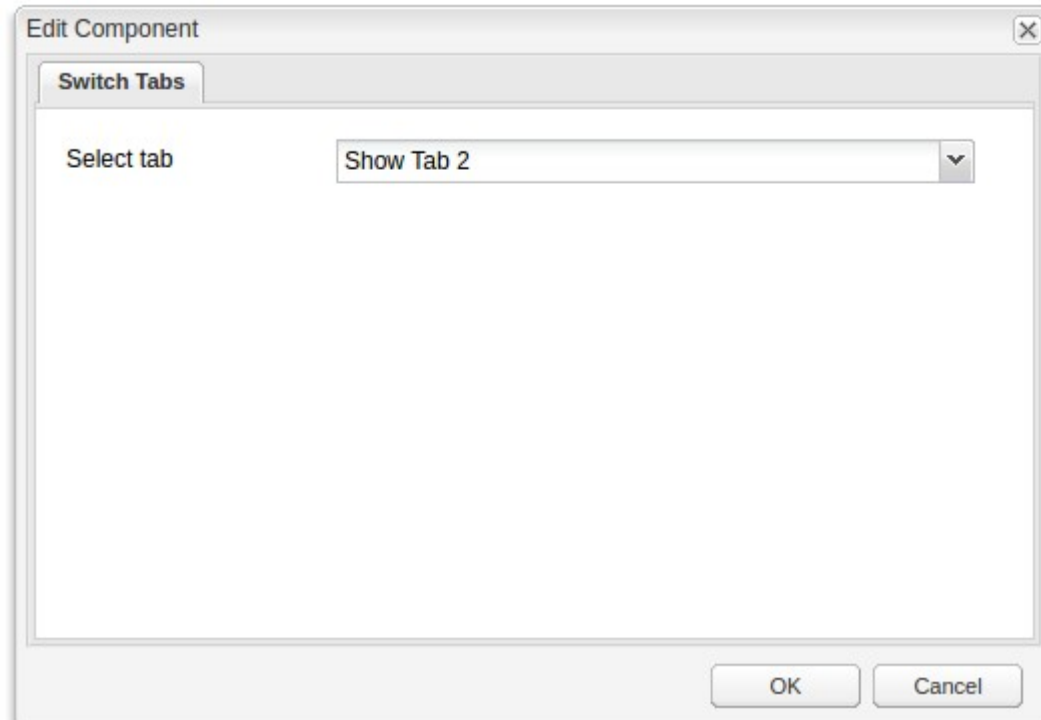
```
<script type="text/javascript">
CQ.WCM.edit({"path":"/content/dynamic-dialogs/test_by/en/homepage/jcr:content/par/dynamictabs", "dialog":"/apps/dynamic-
dialogs/components/dynamicTabs/dialog", "type":"dynamic-dialogs/components/dynamicTabs", "csp":"homepage|page/par|parsys/dynamicTabs|parbase", "editConfig":
{"xtype":"editbar", "inlineEditing":CQ.wcm.EditBase.INLINE_MODE_NEVER, "actions":[{"xtype":"tbtext", "text":" Dynamic
Tabs"}, CQ.wcm.EditBase.EDIT, CQ.wcm.EditBase.DELETE], "disableTargeting":true}});
</script>
```

Classic UI Dialogs

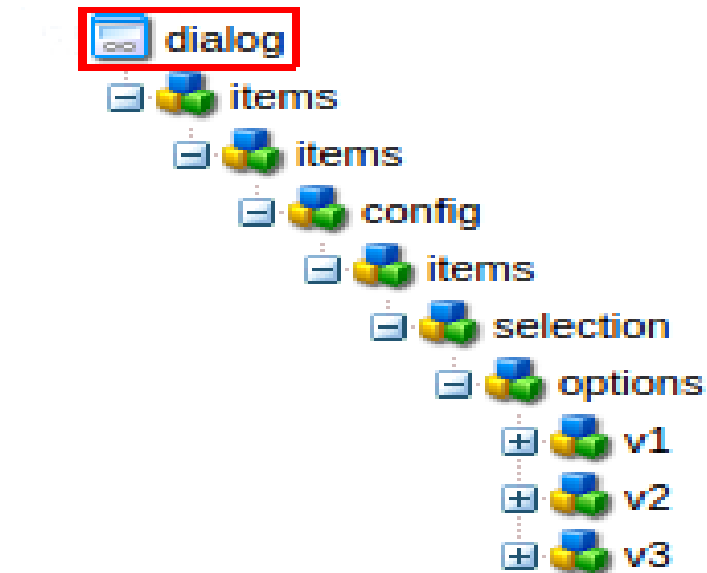
```
<script type="text/javascript">
CQ.WCM.edit({"path":"/content/dynamic-dialogs/test_by/en/homepage/jcr:content/par/dynamicTabs", "dialog":"/apps/dynamic-
dialogs/components/dynamicTabs/dialog", "type":"dynamic-dialogs/components/dynamicTabs", "csp":"homepage|page/par|parsys/dynamicTabs|parbase", "editConfig":
{"xtype":"editbar", "inlineEditing":CQ.wcm.EditBase.INLINE_MODE_NEVER, "actions":[{"xtype":"tbtext", "text":" Dynamic
Tabs"}, CQ.wcm.EditBase.EDIT, CQ.wcm.EditBase.DELETE], "disableTargeting":true}});
</script>
```



ExtJS API + xtypes + config JSON



Classic UI Dialogs



Using xtypes:

<https://docs.adobe.com/docs/en/aem/6-0/develop/components/xtypes.html>

КОНФИГУРИРУЙ

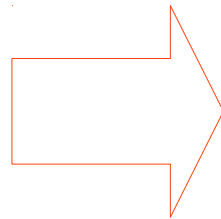
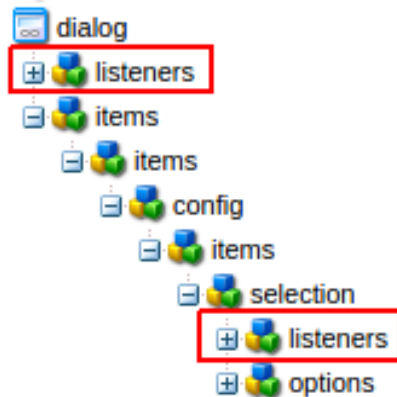


СТРАДАЙ

memegenerator.net

Event handling on Dialog Fields

Event handling on Fields



listeners node

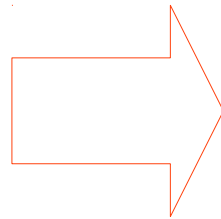
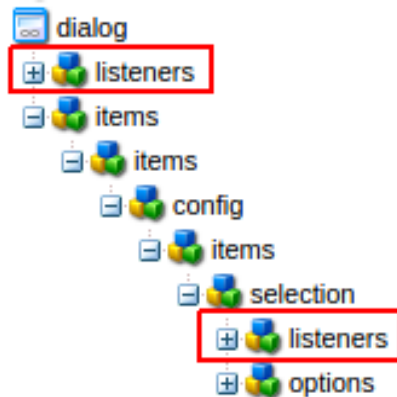
listener - simple js function

each fields has xtype

each xtype has Public Events

1	jcr:primaryType	N... nt:unstructured
2	render	S... function(tab){...}

Event handling on Fields



listeners node

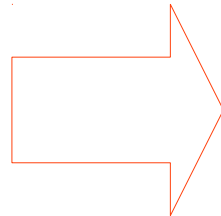
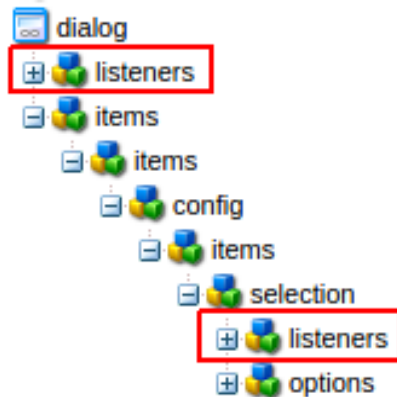
listener - simple js function

each fields has xtype

each xtype has Public Events

1	jcr:primaryType	N... nt:unstructured
2	render	S... function(tab){...}

Event handling on Fields



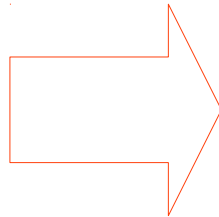
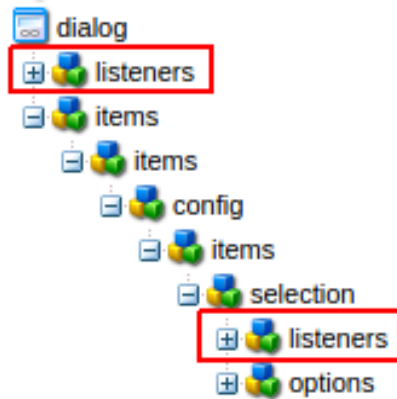
listeners node

listener - simple js function

each fields has xtype

each xtype has Public Events

Event handling on Fields



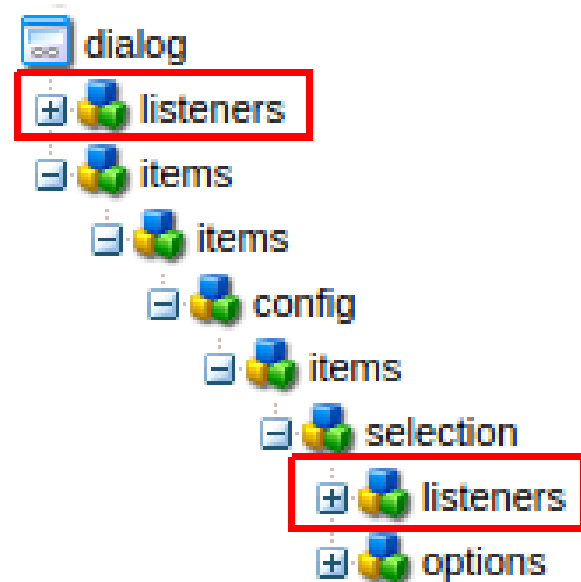
listeners node

listener - simple js function

each fields has xtype

each xtype has Public Events

Event handling on Fields



Using xtype:

<https://docs.adobe.com/docs/en/aem/6-0/develop/components/xtypes.html>

Widget API Documentation:

<https://docs.adobe.com/docs/en/aem/6-0/develop/ref/widgets-api/index.html>

Sample



Sample: dev steps

Step 1: create dialog with all tabs, fields, etc ...

Step 2: choose listeners which will be used

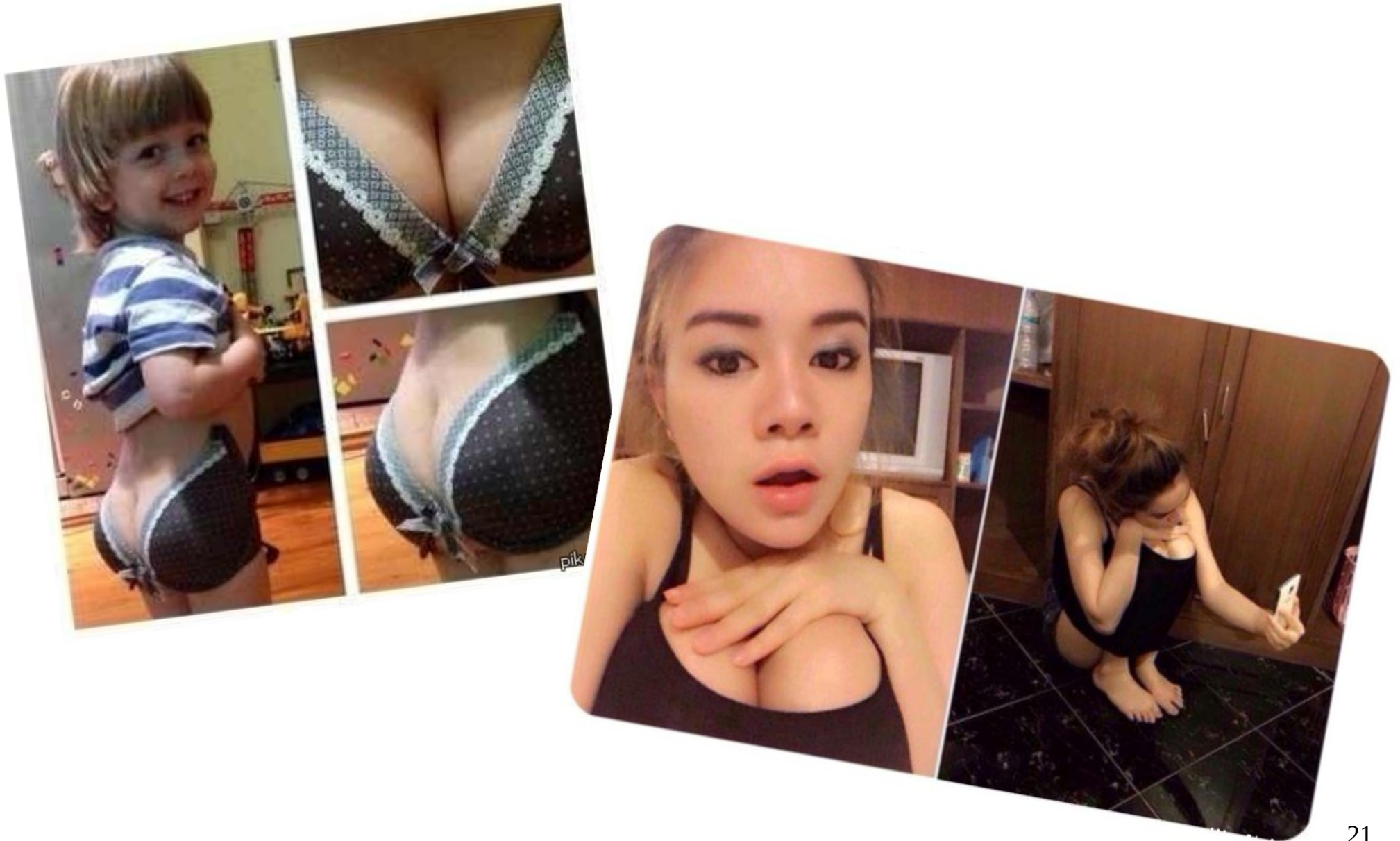
Step 3: create clientLibs, define js module

Step 4: hide all tabs

Step 5: show tabs in depends on chosen select value

Step 6: fix bug with required fields

Tips and Tricks

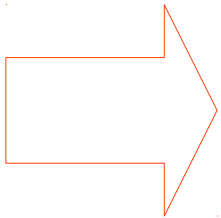


Tips and Tricks

Chose hide/show rather then add/delete

Tips and Tricks

Chose hide/show rather than add/delete



ability to test all dialog components before adding of dynamic behavior

easier to debug

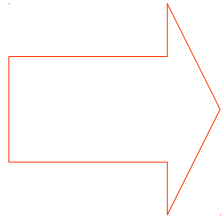
more complex behavior can be implemented

...

Tips and Tricks

Chose hide/show rather than add/delete

ability to test all dialog components before adding of dynamic behavior



easier to debug

more complex behavior can be implemented

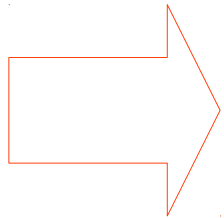
...

Tips and Tricks

Chose hide/show rather than add/delete

ability to test all dialog components before adding of dynamic behavior

easier to debug



more complex behavior can be implemented

...

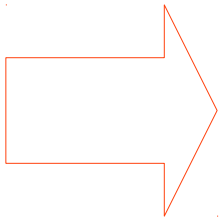


Tips and Tricks

Avoid placing logic under dialog -> listeners

Tips and Tricks

Avoid placing logic under dialog -> listeners



easier to develop

easier to debug

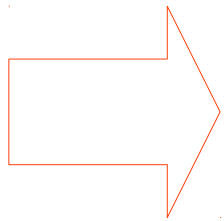
code can be reused

more complex behavior can be implemented

...

Tips and Tricks

Avoid placing logic under dialog -> listeners



easier to develop

easier to debug

code can be reused

more complex behavior can be implemented

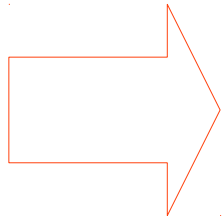
...

Tips and Tricks

Avoid placing logic under dialog -> listeners

easier to develop

easier to debug



code can be reused

more complex behavior can be implemented

...

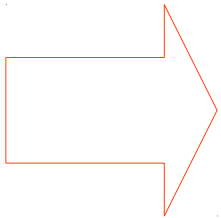
Tips and Tricks

Avoid placing logic under dialog -> listeners

easier to develop

easier to debug

code can be reused




more complex behavior can be implemented

...

Tips and Tricks

Avoid placing logic under dialog -> listeners

```
2 loadcontent S... function(tabPanel, configFieldValue) { var config = 
configFieldValue ? findConfig(configFieldValue) :
null; for (var i = 1; i != Tabs.tabs.length; i++) { if
(config && config.tabs.indexOf(i) >= 0) {
restoreTabFieldsPermissions(tabPanel.getItem(i));
tabPanel.unhideTabStripItem(i); } else {
cleanTabFieldsPermissions(tabPanel.getItem(i));
tabPanel.hideTabStripItem(i); } }
//tabPanel.doLayout(); }
```

VS

Tips and Tricks

Avoid placing logic under dialog -> listeners

```
2 loadcontent S... function(tabPanel, configFieldValue) { var config =  
configFieldValue ? findConfig(configFieldValue) :  
null; for (var i = 1; i != Tabs.tabs.length; i++) { if  
(config && config.tabs.indexOf(i) >= 0) {  
restoreTabFieldsPermissions(tabPanel.getItem(i));  
tabPanel.unhideTabStripItem(i); } else {  
cleanTabFieldsPermissions(tabPanel.getItem(i));  
tabPanel.hideTabStripItem(i); }  
//tabPanel.doLayout(); }
```

VS

loadcontent

S... function(field,rec,path){DD.Tabs.showTabs(field);}

+



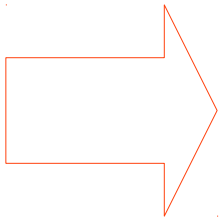
```
/**  
 * Manages the tabs of the specified tab panel.  
 * Tabs which allowed into config will be shown,  
 * the others will be hidden.  
 */  
Tabs.manageTabs = function(tabPanel, configFieldValue) {  
    var config = configFieldValue ? findConfig(configFieldValue) : null;  
  
    for (var i = 1; i != Tabs.tabs.length; i++) {  
        if (config && config.tabs.indexOf(i) >= 0) {  
            restoreTabFieldsPermissions(tabPanel.getItem(i));  
            tabPanel.unhideTabStripItem(i);  
        } else {  
            cleanTabFieldsPermissions(tabPanel.getItem(i));  
            tabPanel.hideTabStripItem(i);  
        }  
    }  
    //tabPanel.doLayout();  
};
```


Tips and Tricks

Keep logic under clientlibs

Tips and Tricks

Keep logic under clientlibs



CQ way to implement js logic

can be switched on/off

clientlibs features can be used

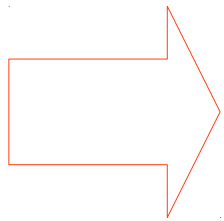
more complex behavior can be implemented

...



Tips and Tricks

Keep logic under clientlibs



cq way to implement js logic

can be switched on/off

clientlibs features can be used

more complex behavior can be implemented

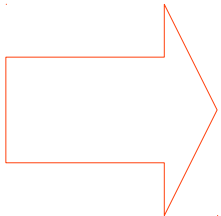
...

Tips and Tricks

Keep logic under clientlibs

cq way to implement js logic

can be switched on/off



clientlibs features can be used

more complex behavior can be implemented

...

Tips and Tricks

Keep logic under clientlibs

cq way to implement js logic

can be switched on/off

clientlibs features can be used

more complex behavior can be implemented

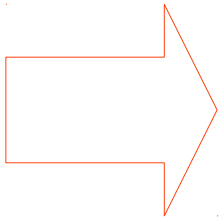
...

Tips and Tricks

If something is not visible it doesn't mean it is not validatable

Tips and Tricks

If something is not visible it doesn't mean it is not validatable

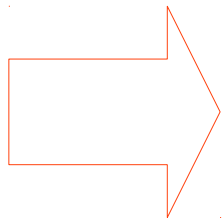


switched on/off validation once show/hide

store validation rules before hide

Tips and Tricks

If something is not visible it doesn't mean it is not validatable



switched on/off validation once show/hide

store validation rules before hide

Thank you!

Questions

