

JBoss Community

LEVERAGE JQUERY PLUGINS To Create JSF Components

Brian Leathem
Senior Software Engineer, Red Hat
JavaOne 2012 - San Francisco
2012-09-02

JSF - STANDARDIZED COMPONENT MODEL

Component-oriented
web framework



COMPONENT PROVIDERS



RichFaces



ICEfaces
ICEfaces



OpenFaces



ORACLE®

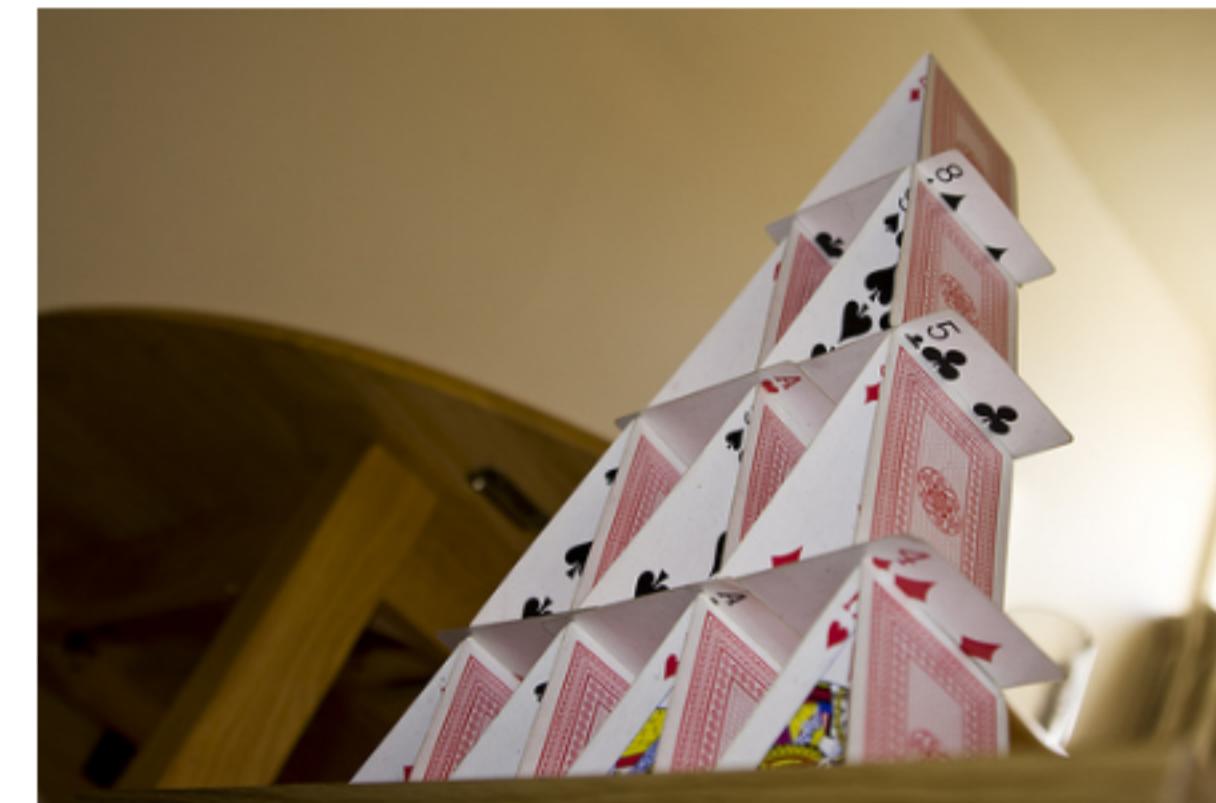
BUILDING JSF COMPONENTS

Server-side code

- Components
- Renderers
- taglib xml
- faces-config.xml

Client-side code

- HTML
- CSS
- javascript



LOTS OF BROWSERS



LOTS OF BROWSERS



+ MOBILE!

THE SOLUTION

Re-use existing javascript "widgets"
There is a huge potential for code re-use

THE SOLUTION

Re-use existing javascript "widgets"
There is a huge potential for code re-use

Google search comparison:

JSF Component:

- ~ 3 Million hits

Javascript Component:

- ~ 416 Million hits

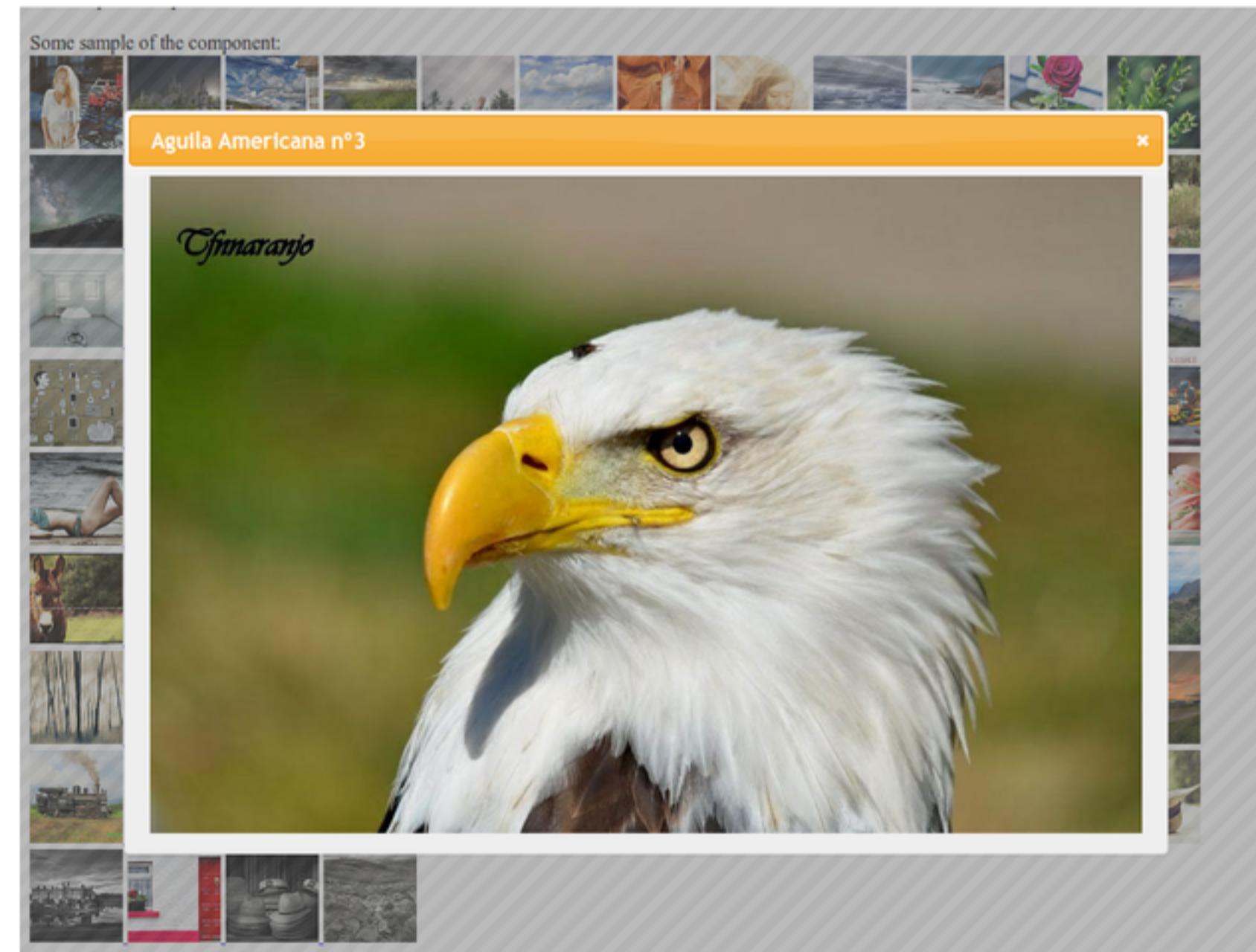
POLYGLOT WEB APPS

Polyglot JVM

Web frameworks in many languages/environments

Consistent L&F throughout a heterogeneous application

IMAGE GALLERY



JQUERY UI

Accordion

Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam. Integer ut neque. Vivamus nisi metus, molestie vel, gravida in, condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante scelerisque vulputate.

Tabs

First	Second	Third
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.		

Dialog

 Open Dialog

Overlay and Shadow Classes

**Nam c
venen.** **conse**
Lorem ipsum dolor sit amet, Nulla nec tortor. Donec id elit quis purus
adipisicing elit, sed do eiusmod tempor

Button

A button element

Choice 1 Choice 2 Choice 3

Autocomplete

100

Slide



Datepicker

Progressbar



Highlight / Error

Hey! Sample ui-state-highlight style.

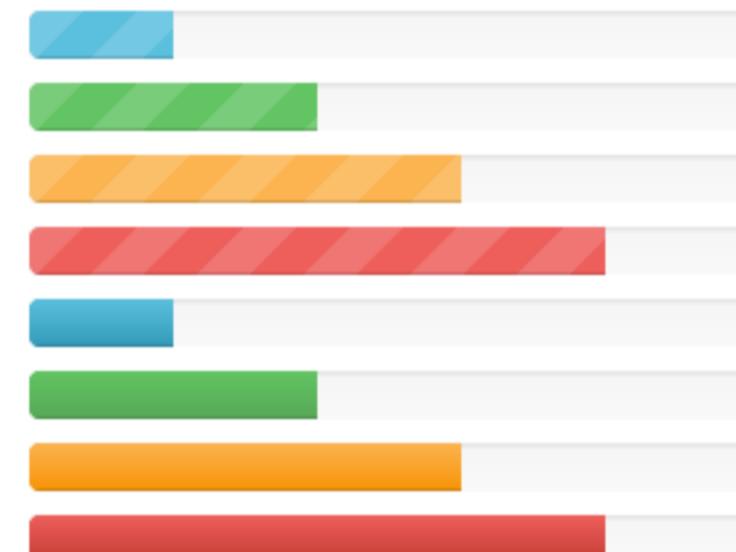
BOOTSTRAP

Default
Success
Warning
Important
Info
Inverse

1
2
4
6
8
10

Inverse
Inverse ▾
Warning ▾

Primary action Action
Primary action Action
Primary action Action
Primary link Link
Primary button Button



Dropup ▾ Right dropup ▾

Large button ▾ Small button ▾ Mini button ▾

Large action ▾ Small action ▾ Mini action ▾

\$.00

LIST HEADER
Home
Library
Applications
ANOTHER LIST HEADER
Profile
Settings
Help

IMAGE GALLERY

- A jQuery UI imagegallery plugin
- An output only component

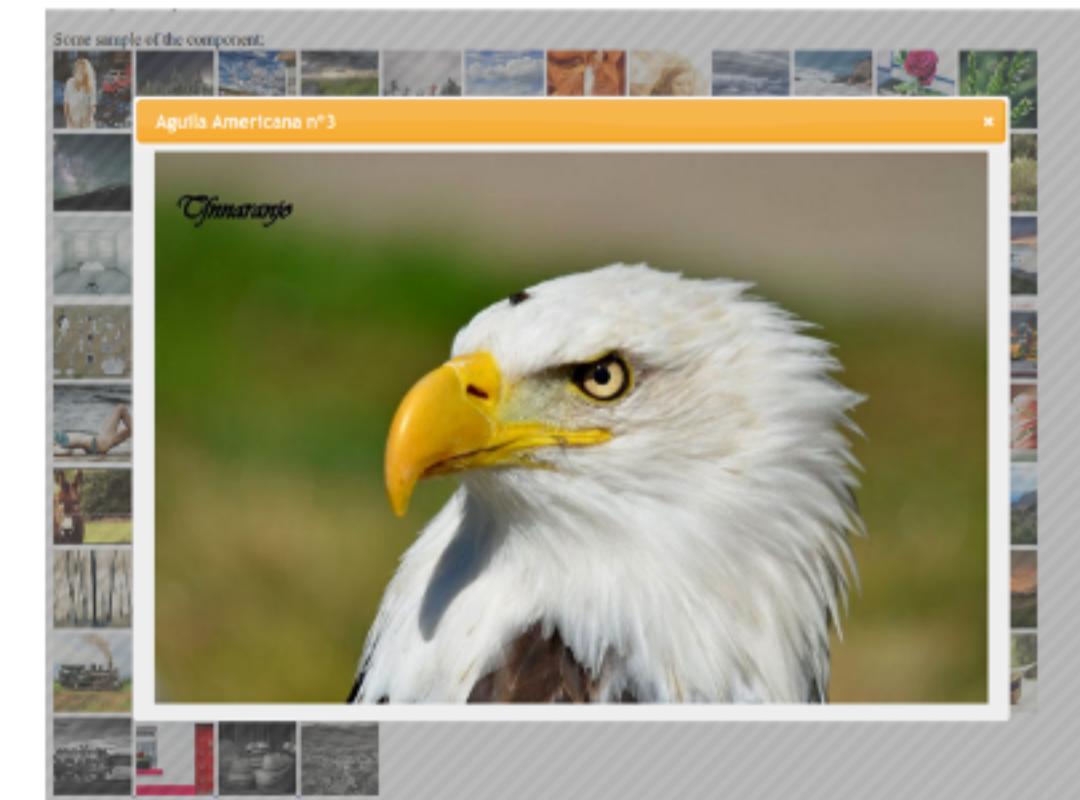


IMAGE GALLERY - COMPONENT

```
<!DOCTYPE html>
<html lang="en-US"
      xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:cc="http://java.sun.com/jsf/composite">

<cc:interface>
    <cc:attribute name="photos" />

    <!-- plugin configuration -->
    <cc:attribute name="selector" />
    [ ... snip ... ]
    <cc:attribute name="dialogClass" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}:div">
        <ui:repeat value="#{cc.attrs.photos}" var="photo">
            <a rel="gallery" href="#{photo.full}" title="#{photo.title}">
                
            </a>
        </ui:repeat>
    </div>

    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/js" name="jquery-ui-1.8.19.custom.min.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="load-image.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="jquery.image-gallery.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputStylesheet library="composite/com.github.blueimp/css" name="jquery.image-gallery.css"/>
    <h:outputScript>
        var pluginOptions = {};
        ! '#{cc.attrs.selector}' || (pluginOptions.selector = '#{cc.attrs.selector}');
        [ ... snip ... ]
        ! '#{cc.attrs.dialogClass}' || (pluginOptions.dialogClass = '#{cc.attrs.dialogClass}');

        jQuery(function() {
            $(document.getElementById('#{cc.clientId}:div')).imagegallery(pluginOptions);
        });
    </h:outputScript>
</cc:implementation>

</html>
```

IMAGE GALLERY - COMPONENT

```
<!DOCTYPE html>
<html lang="en-US"
      xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:cc="http://java.sun.com/jsf/composite">

<cc:interface>
    <cc:attribute name="photos" />

    <!-- plugin configuration -->
    <cc:attribute name="selector" />
    [ ... snip ... ]
    <cc:attribute name="dialogClass" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}:div">
        <ui:repeat value="#{cc.attrs.photos}" var="photo">
            <a rel="gallery" href="#{(photo.full)}" title="#{(photo.title)}">
                
            </a>
        </ui:repeat>
    </div>

    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/js" name="jquery-ui-1.8.19.custom.min.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="load-image.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="jquery.image-gallery.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputStylesheet library="composite/com.github.blueimp/css" name="jquery.image-gallery.css"/>
    <h:outputScript>
        var pluginOptions = {};
        ! '#{cc.attrs.selector}' || (pluginOptions.selector = '#{cc.attrs.selector}');
        [ ... snip ... ]
        ! '#{cc.attrs.dialogClass}' || (pluginOptions.dialogClass = '#{cc.attrs.dialogClass}');

        jQuery(function() {
            $(document.getElementById('#{cc.clientId}:div')).imagegallery(pluginOptions);
        });
    </h:outputScript>
</cc:implementation>
</html>
```

IMAGE GALLERY - COMPONENT

```
<!DOCTYPE html>
<html lang="en-US"
      xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:cc="http://java.sun.com/jsf/composite">

    <cc:interface>
        <cc:attribute name="photos" />

        <!-- plugin configuration -->
        <cc:attribute name="selector" />
        [ ... snip ... ]
        <cc:attribute name="dialogClass" />
    </cc:interface>

    <cc:implementation>
        <div id="#{cc.clientId}:div">
            <ui:repeat value="#{cc.attrs.photos}" var="photo">
                <a rel="gallery" href="#{photo.full}" title="#{photo.title}">
                    
                </a>
            </ui:repeat>
        </div>

        <h:outputScript library="javax.faces" name="jsf.js"/>
        <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
        <h:outputScript library="composite/com.jqueryui/js" name="jquery-ui-1.8.19.custom.min.js"/>
        <h:outputScript library="composite/com.github.blueimp/js" name="load-image.js"/>
        <h:outputScript library="composite/com.github.blueimp/js" name="jquery.image-gallery.js"/>
        <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
        <h:outputStylesheet library="composite/com.github.blueimp/css" name="jquery.image-gallery.css"/>
        <h:outputScript>
            var pluginOptions = {};
            ! '#{cc.attrs.selector}' || (pluginOptions.selector = '#{cc.attrs.selector}');
            [ ... snip ... ]
            ! '#{cc.attrs.dialogClass}' || (pluginOptions.dialogClass = '#{cc.attrs.dialogClass}');

            jQuery(function() {
                $document.getElementById('#{cc.clientId}:div').imagegallery(pluginOptions);
            });
        </h:outputScript>
    </cc:implementation>
</html>
```

IMAGE GALLERY - COMPONENT

```
<!DOCTYPE html>
<html lang="en-US"
      xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:cc="http://java.sun.com/jsf/composite">

<cc:interface>
    <cc:attribute name="photos" />

    <!-- plugin configuration -->
    <cc:attribute name="selector" />
    [ ... snip ... ]
    <cc:attribute name="dialogClass" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}:div">
        <ui:repeat value="#{cc.attrs.photos}" var="photo">
            <a rel="gallery" href="#{photo.full}" title="#{photo.title}">
                
            </a>
        </ui:repeat>
    </div>

    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/js" name="jquery-ui-1.8.19.custom.min.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="load-image.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="jquery.image-gallery.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputStylesheet library="composite/com.github.blueimp/css" name="jquery.image-gallery.css"/>
    <h:outputScript>
        var pluginOptions = {};
        ! '#{cc.attrs.selector}' || (pluginOptions.selector = '#{cc.attrs.selector}');
        [ ... snip ... ]
        ! '#{cc.attrs.dialogClass}' || (pluginOptions.dialogClass = '#{cc.attrs.dialogClass}');

        jQuery(function() {
            $document.getElementById('#{cc.clientId}:div').imagegallery(pluginOptions);
        });
    </h:outputScript>
</cc:implementation>

</html>
```

IMAGE GALLERY - COMPONENT

```
<!DOCTYPE html>
<html lang="en-US"
      xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:cc="http://java.sun.com/jsf/composite">

<cc:interface>
    <cc:attribute name="photos" />

    <!-- plugin configuration -->
    <cc:attribute name="selector" />
    [ ... snip ... ]
    <cc:attribute name="dialogClass" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}:div">
        <ui:repeat value="#{cc.attrs.photos}" var="photo">
            <a rel="gallery" href="#{photo.full}" title="#{photo.title}">
                
            </a>
        </ui:repeat>
    </div>
</cc:implementation>

<h:outputScript library="javax.faces" name="jsf.js"/>
<h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
<h:outputScript library="composite/com.jqueryui/js" name="jquery-ui-1.8.19.custom.min.js"/>
<h:outputScript library="composite/com.github.blueimp/js" name="load-image.js"/>
<h:outputScript library="composite/com.github.blueimp/js" name="jquery.image-gallery.js"/>
<h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
<h:outputStylesheet library="composite/com.github.blueimp/css" name="jquery.image-gallery.css"/>

<h:outputScript>
    var pluginOptions = {};
    ! '#{cc.attrs.selector}' || (pluginOptions.selector = '#{cc.attrs.selector}');
    [ ... snip ... ]
    ! '#{cc.attrs.dialogClass}' || (pluginOptions.dialogClass = '#{cc.attrs.dialogClass}');

    jQuery(function() {
        $(document.getElementById('#{cc.clientId}:div')).imagegallery(pluginOptions);
    });
</h:outputScript>
</cc:implementation>

</html>
```

IMAGE GALLERY - COMPONENT

```
<!DOCTYPE html>
<html lang="en-US"
      xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:cc="http://java.sun.com/jsf/composite">

<cc:interface>
    <cc:attribute name="photos" />

    <!-- plugin configuration -->
    <cc:attribute name="selector" />
    [ ... snip ... ]
    <cc:attribute name="dialogClass" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}:div">
        <ui:repeat value="#{cc.attrs.photos}" var="photo">
            <a rel="gallery" href="#{photo.full}" title="#{photo.title}">
                
            </a>
        </ui:repeat>
    </div>

    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/js" name="jquery-ui-1.8.19.custom.min.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="load-image.js"/>
    <h:outputScript library="composite/com.github.blueimp/js" name="jquery.image-gallery.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>

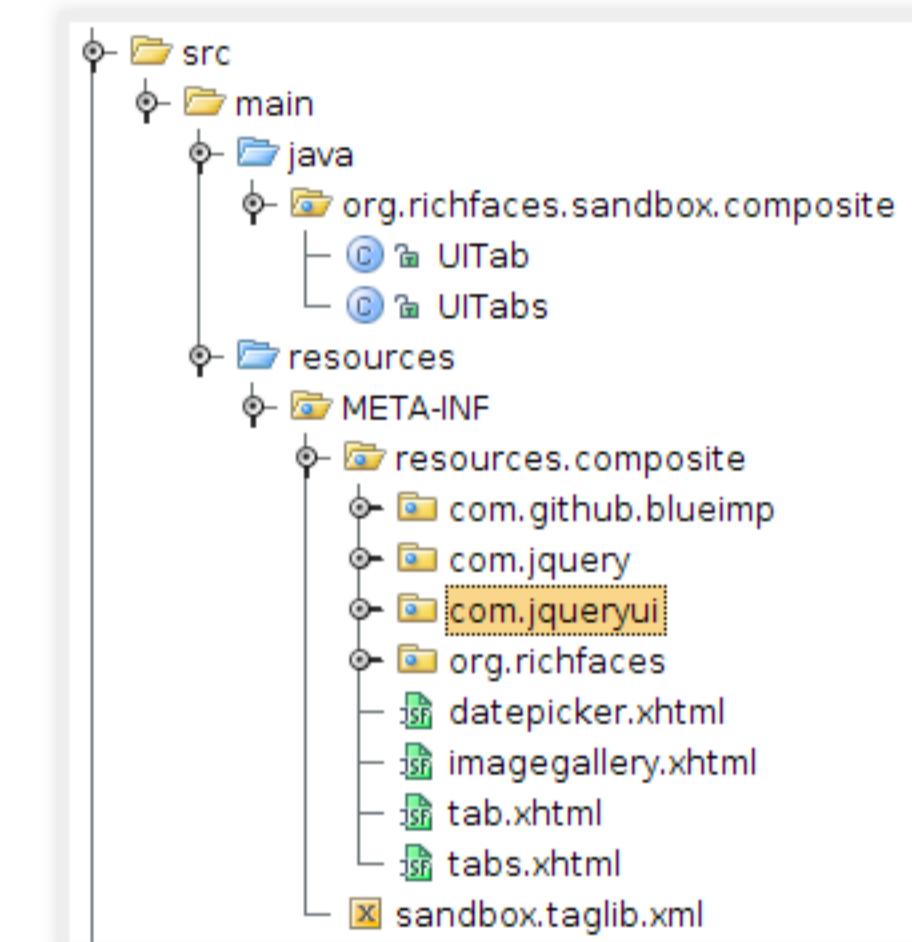
var pluginOptions = {};
! '#{cc.attrs.selector}' || (pluginOptions.selector = '#{cc.attrs.selector}');
[ ... snip ... ]
! '#{cc.attrs.dialogClass}' || (pluginOptions.dialogClass = '#{cc.attrs.dialogClass>');

jQuery(function() {
    $(document.getElementById('#{cc.clientId}:div')).imagegallery(pluginOptions);
});

</h:outputScript>
</cc:implementation>
</html>
```

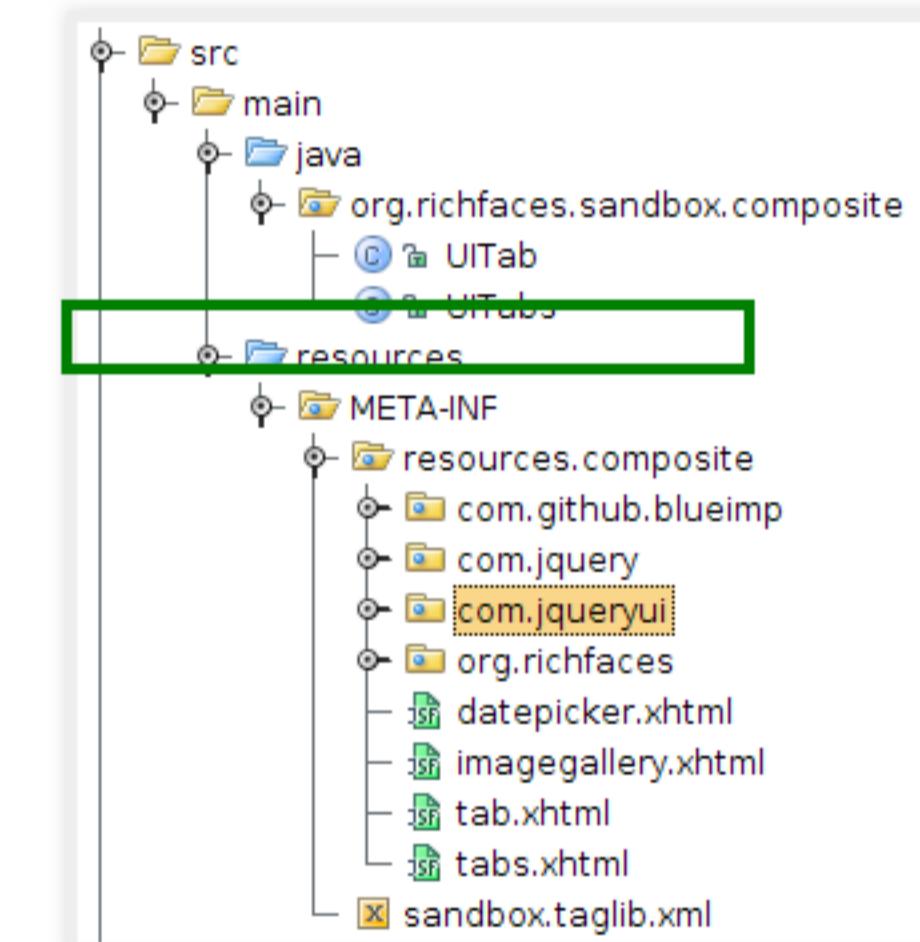
PACKAGING THE CC

Package your composite components in a jar file to share across teams/apps



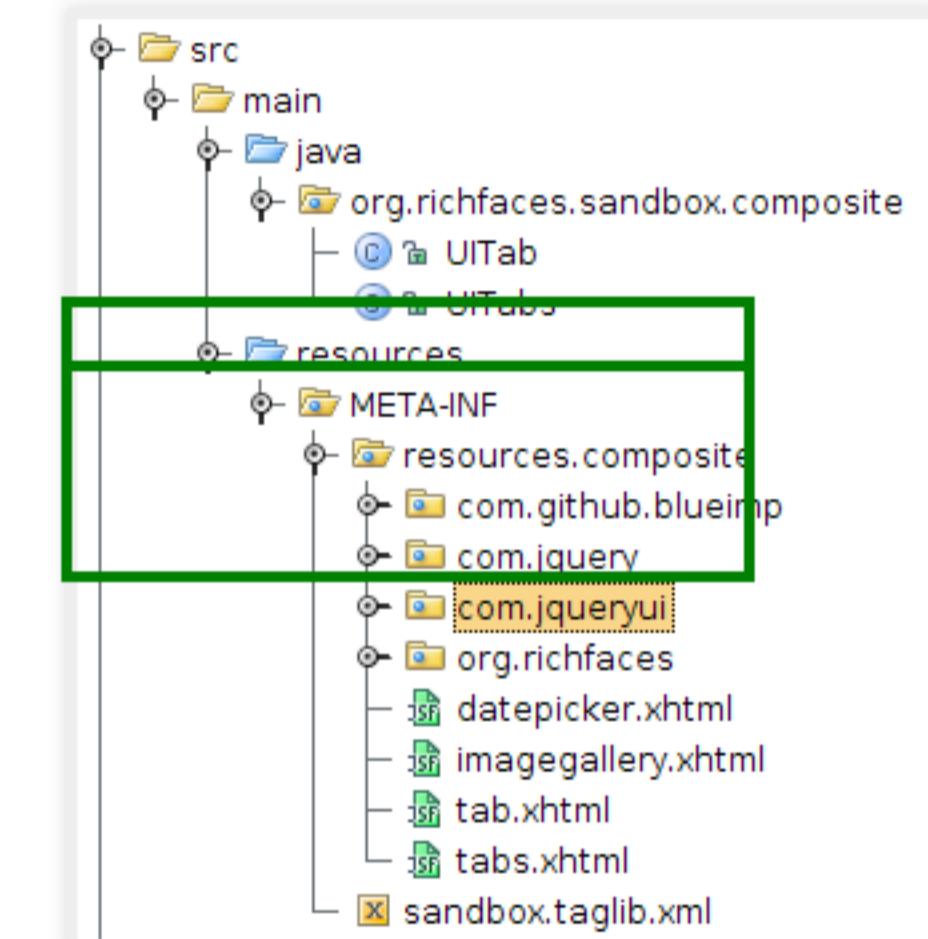
PACKAGING THE CC

Package your composite components in a jar file to share across teams/apps



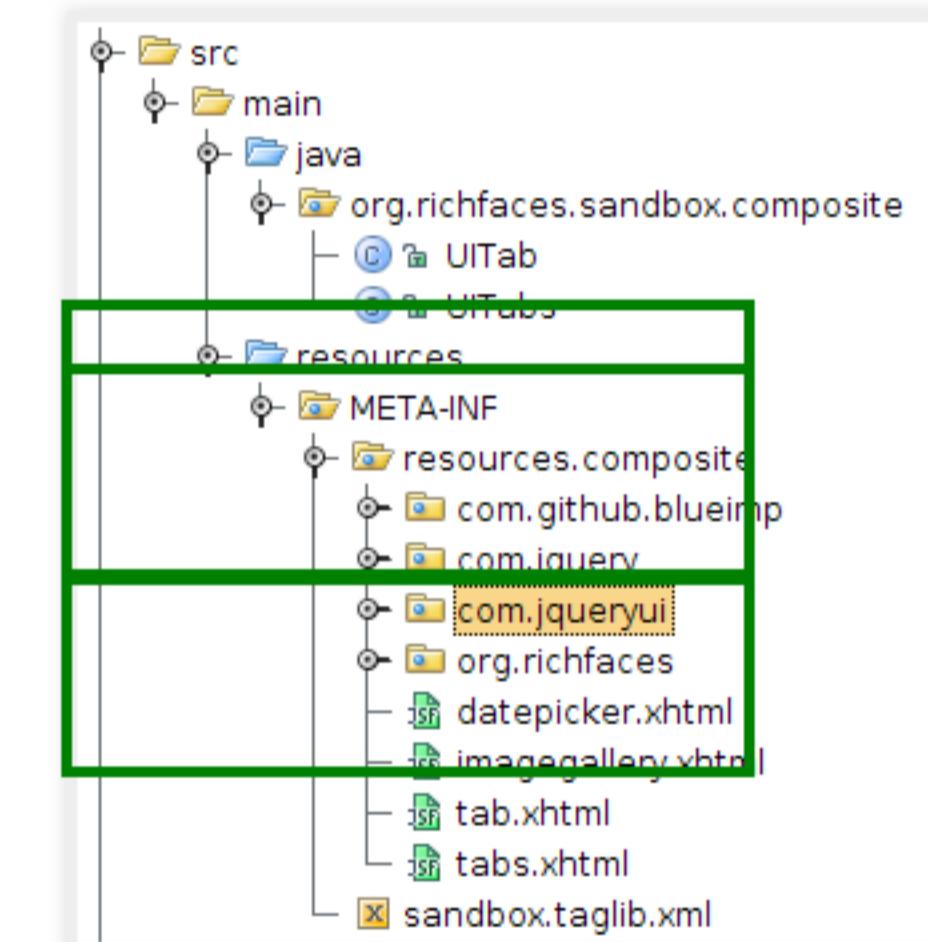
PACKAGING THE CC

Package your composite components in a jar file to share across teams/apps



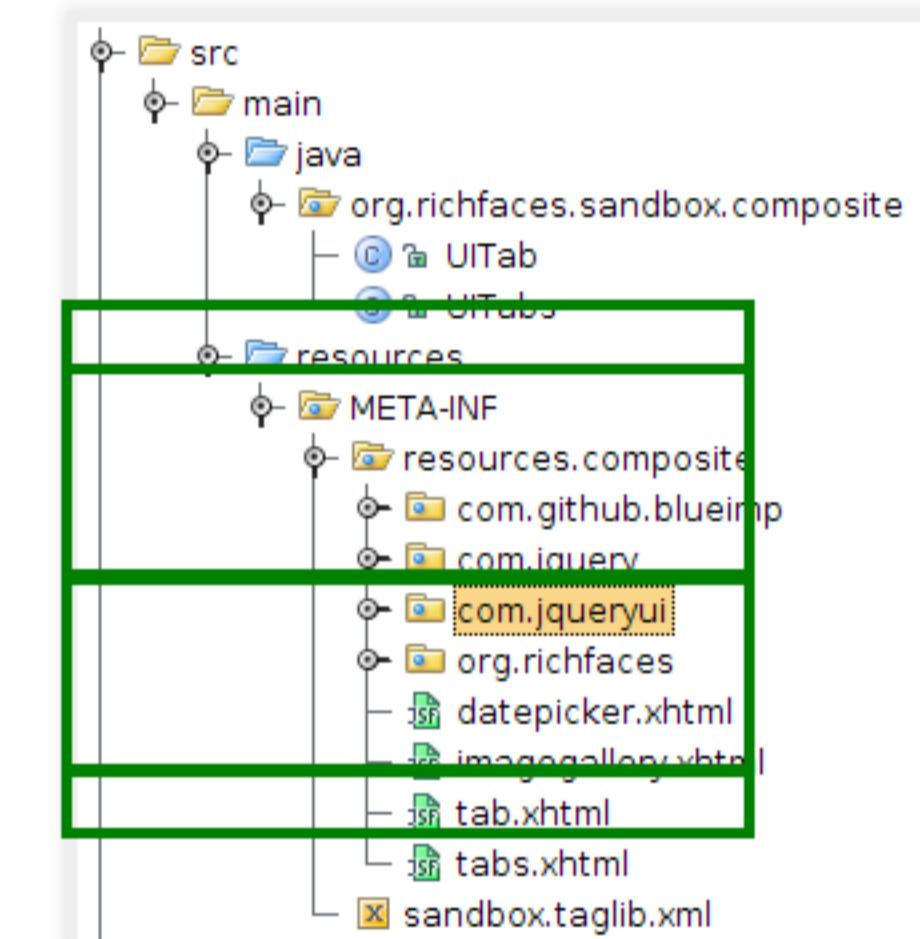
PACKAGING THE CC

Package your composite components in a jar file to share across teams/apps



PACKAGING THE CC

Package your composite components in a jar file to share across teams/apps



USING THE IMAGEGALLERY COMPONENT

```
<!DOCTYPE html>
<html lang="en-US">

<ui:composition template="/resources/template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:s="http://richfaces.org/sandbox/composite">
    <ui:define name="body">
        <h:form id="form">
            Some sample of the component:
            <s:imagegallery photos="#{galleryBean.photos}" />
        </h:form>
    </ui:define>
</ui:composition>
</html>
```

USING THE IMAGEGALLERY COMPONENT

```
<!DOCTYPE html>
<html lang="en-US">

<ui:composition template="/resources/template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:s="http://richfaces.org/sandbox/composite">
    <ui:define name="body">
        <h:form id="form">
            Some sample of the component:
            <s:imagegallery photos="#{galleryBean.photos}" />
        </h:form>
    </ui:define>
</ui:composition>
</html>
```

USING THE IMAGEGALLERY COMPONENT

```
<!DOCTYPE html>
<html lang="en-US">

<ui:composition template="/resources/template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:s="http://richfaces.org/sandbox/composite">
    <ui:define name="body">
        <h:form id="form">
            Some sample of the component:
            <s:imagegallery photos="#{galleryBean.photos}" />
        </h:form>
    </ui:define>
</ui:composition>
</html>
```

GALLERYBEAN.JAVA

```
@ManagedBean  
@RequestScoped  
public class GalleryBean {  
    private List<FlickrPhoto> photos;  
  
    public GalleryBean() {  
        photos = new ArrayList<FlickrPhoto>();  
        photos.add(new FlickrPhoto("title a", "http://flickr.com/a..."));  
        photos.add(new FlickrPhoto("title c", "http://flickr.com/c..."));  
        photos.add(new FlickrPhoto("title d", "http://flickr.com/d..."));  
        ...  
    }  
  
    public List<FlickrPhoto> getPhotos() {  
        return photos;  
    }  
}
```

GALLERYBEAN.JAVA

```
@ManagedBean  
@RequestScoped  
public class GalleryBean {  
    private List<FlickrPhoto> photos;  
  
    public GalleryBean() {  
        photos = new ArrayList<FlickrPhoto>();  
        photos.add(new FlickrPhoto("title a", "http://flickr.com/a..."));  
        photos.add(new FlickrPhoto("title c", "http://flickr.com/c..."));  
        photos.add(new FlickrPhoto("title d", "http://flickr.com/d..."));  
        ...  
    }  
  
    public List<FlickrPhoto> getPhotos() {  
        return photos;  
    }  
}
```

GALLERYBEAN.JAVA

```
@ManagedBean  
@RequestScoped  
public class GalleryBean {  
    private List<FlickrPhoto> photos;  
  
    public GalleryBean() {  
        photos = new ArrayList<FlickrPhoto>();  
        photos.add(new FlickrPhoto("title a", "http://flickr.com/a..."));  
        photos.add(new FlickrPhoto("title c", "http://flickr.com/c..."));  
        photos.add(new FlickrPhoto("title d", "http://flickr.com/d..."));  
        ...  
    }  
  
    public List<FlickrPhoto> getPhotos() {  
        return photos;  
    }  
}
```

JQUERY UI DATEPICKER COMPONENT

- The jQuery UI datepicker plugin
- An input component, bound to a backing bean



DATEPICKER COMPOSITE COMPONENT

```
<cc:interface>
    <cc:attribute name="value" />
    <cc:attribute name="showOn" />
    <cc:attribute name="dateFormat" />
    <cc:attribute name="buttonImageOnly" />
</cc:interface>

<cc:implementation>
    <h:inputText id="input" value="#{cc.attrs.value}" />
    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.datepicker.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputScript>
        var pluginOptions = {
            buttonImage: "${request.contextPath}/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
            buttonImageOnly: "#{true eq cc.attrs.buttonImageOnly}"
        }
        ! '#{cc.attrs.showOn}' || (pluginOptions.showOn = '#{cc.attrs.showOn}');
        ! '#{cc.attrs.dateFormat}' || (pluginOptions.dateFormat = '#{cc.attrs.dateFormat}');

        jQuery(function() {
            $(document.getElementById('#{cc.clientId}:input')).datepicker(pluginOptions);
        });
    </h:outputScript>
</cc:implementation>
```

DATEPICKER COMPOSITE COMPONENT

```
<cc:interface>
    <cc:attribute name="value" />
    <cc:attribute name="showOn" />
    <cc:attribute name="dateFormat" />
    <cc:attribute name="buttonImageOnly" />
</cc:interface>

<cc:implementation>
    <h:inputText id="input" value="#{cc.attrs.value}" />
    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.datepicker.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputScript>
        var pluginOptions = {
            buttonImage: "#{request.contextPath}/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
            buttonImageOnly: "#{true} eq cc.attrs.buttonImageOnly"
        }
        ! '#{cc.attrs.showOn}' || (pluginOptions.showOn = '#{cc.attrs.showOn}');
        ! '#{cc.attrs.dateFormat}' || (pluginOptions.dateFormat = '#{cc.attrs.dateFormat}');

        jQuery(function() {
            $(document.getElementById('#{cc.clientId}:input')).datepicker(pluginOptions);
        });
    </h:outputScript>
</cc:implementation>
```

DATEPICKER COMPOSITE COMPONENT

```
<cc:interface>
    <cc:attribute name="value" />
    <cc:attribute name="showOn" />
    <cc:attribute name="dateFormat" />
    <cc:attribute name="buttonImageOnly" />
</cc:interface>

<cc:implementation>
    <h:inputText id="input" value="#{cc.attrs.value}" />
    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.datepicker.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputScript>
        var pluginOptions = {
            buttonImage: "#{request.contextPath}/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
            buttonImageOnly: "#{true eq cc.attrs.buttonImageOnly}"
        }
        ! '#{cc.attrs.showOn}' || (pluginOptions.showOn = '#{cc.attrs.showOn}');
        ! '#{cc.attrs.dateFormat}' || (pluginOptions.dateFormat = '#{cc.attrs.dateFormat}');

        jQuery(function() {
            $('#{cc.clientId}:input').datepicker(pluginOptions);
        });
    </h:outputScript>
</cc:implementation>
```

DATEPICKER COMPOSITE COMPONENT

```
<cc:interface>
    <cc:attribute name="value" />
    <cc:attribute name="showOn" />
    <cc:attribute name="dateFormat" />
    <cc:attribute name="buttonImageOnly" />
</cc:interface>

<cc:implementation>
    <h:inputText id="input" value="#{cc.attrs.value}" />

    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.datepicker.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>

        <h:outputScript>
            var pluginOptions = {
                buttonImage: "#{request.contextPath}/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
                buttonImageOnly: "#{true} eq cc.attrs.buttonImageOnly"
            }
            ! '#{cc.attrs.showOn}' || (pluginOptions.showOn = '#{cc.attrs.showOn}');
            ! '#{cc.attrs.dateFormat}' || (pluginOptions.dateFormat = '#{cc.attrs.dateFormat}');

            jQuery(function() {
                $(document.getElementById('#{cc.clientId}:input')).datepicker(pluginOptions);
            });
        </h:outputScript>
</cc:implementation>
```

DATEPICKER COMPOSITE COMPONENT

```
<cc:interface>
    <cc:attribute name="value" />
    <cc:attribute name="showOn" />
    <cc:attribute name="dateFormat" />
    <cc:attribute name="buttonImageOnly" />
</cc:interface>

<cc:implementation>
    <h:inputText id="input" value="#{cc.attrs.value}" />
    <h:outputScript library="javax.faces" name="jsf.js"/>
    <h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
    <h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.datepicker.js"/>
    <h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
    <h:outputScript>

var pluginOptions = {
    buttonImage: "#{request.contextPath}/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
    buttonImageOnly: "#{true eq cc.attrs.buttonImageOnly}"
}
! '#{cc.attrs.showOn}' || (pluginOptions.showOn = '#{cc.attrs.showOn}');
! '#{cc.attrs.dateFormat}' || (pluginOptions.dateFormat = '#{cc.attrs.dateFormat}');

jQuery(function() {
    $(document.getElementById('#{cc.clientId}:input')).datepicker(pluginOptions);
});

</h:outputScript>
</cc:implementation>
```

DATEPICKER USAGE

```
<s:datepicker buttonImageOnly="true"  
              dateFormat="yy-mm-dd"  
              showOn="both"/>
```

GENERATED MARKUP

```
<input class="hasDatepicker" id="form:j_idt16:input" name="form:j_idt16:input" type="text">

<script type="text/javascript" src="/composite-demo/javax.faces.resource/jsf.js.jsf?ln=javax.faces"></script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery-1.7.2.min.js.jsf?ln=composite/com.jquery">
</script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery.ui.core.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery.ui.widget.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery.ui.datepicker.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript">
    var pluginOptions = {
        buttonImage:"/composite-demo/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
        buttonImageOnly:"true"
    }
    !'both' || (pluginOptions.showOn = 'both');
    !'yy-mm-dd' || (pluginOptions.dateFormat = 'yy-mm-dd');

    jQuery(function () {
        $(document.getElementById('form:j_idt16:input')).datepicker(pluginOptions);
    });
</script>
```

GENERATED MARKUP

```
<input class="hasDatepicker" id="form:j_idt16:input" name="form:j_idt16:input" type="text">

<script type="text/javascript" src="/composite-demo/javax.faces.resource/jsf.js.jsf?ln=javax.faces"></script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery-1.7.2.min.js.jsf?ln=composite/com.jquery">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.core.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.widget.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.datepicker.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript">
    var pluginOptions = {
        buttonImage:"/composite-demo/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
        buttonImageOnly:"true"
    }
    !'both' || (pluginOptions.showOn = 'both');
    !'yy-mm-dd' || (pluginOptions.dateFormat = 'yy-mm-dd');

    jQuery(function () {
        $(document.getElementById('form:j_idt16:input')).datepicker(pluginOptions);
    });
</script>
```

GENERATED MARKUP

```
<input class="hasDatepicker" id="form:j_idt16:input" name="form:j_idt16:input" type="text">

<script type="text/javascript" src="/composite-demo/javax.faces.resource/jsf.js.jsf?ln=javax.faces"></script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery-1.7.2.min.js.jsf?ln=composite/com.jquery">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.core.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.widget.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.datepicker.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript">
    var pluginOptions = {
        buttonImage:"/composite-demo/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
        buttonImageOnly:"true"
    }
    !'both' || (pluginOptions.showOn = 'both');
    !'yy-mm-dd' || (pluginOptions.dateFormat = 'yy-mm-dd');

    jQuery(function () {
        $(document.getElementById('form:j_idt16:input')).datepicker(pluginOptions);
    });
</script>
```

GENERATED MARKUP

```
<input class="hasDatepicker" id="form:j_idt16:input" name="form:j_idt16:input" type="text">

<script type="text/javascript" src="/composite-demo/javax.faces.resource/jsf.js.jsf?ln=javax.faces"></script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery-1.7.2.min.js.jsf?ln=composite/com.jquery">
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.core.js.jsf?ln=composite/com.jqueryui/development-bu
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.widget.js.jsf?ln=composite/com.jqueryui/development-b
</script>
<script type="text/javascript"
       src="/composite-demo/javax.faces.resource/jquery.ui.datepicker.js.jsf?ln=composite/com.jqueryui/developme
ui">
</script>
<script type="text/javascript">
  var pluginOptions = {
    buttonImage:"/composite-demo/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
    buttonImageOnly:"true"
  }
  !'both' || (pluginOptions.showOn = 'both');
  !'yy-mm-dd' || (pluginOptions.dateFormat = 'yy-mm-dd');

  jQuery(function () {
    $(document.getElementById('form:j_idt16:input')).datepicker(pluginOptions);
  });
</script>
```

GENERATED MARKUP

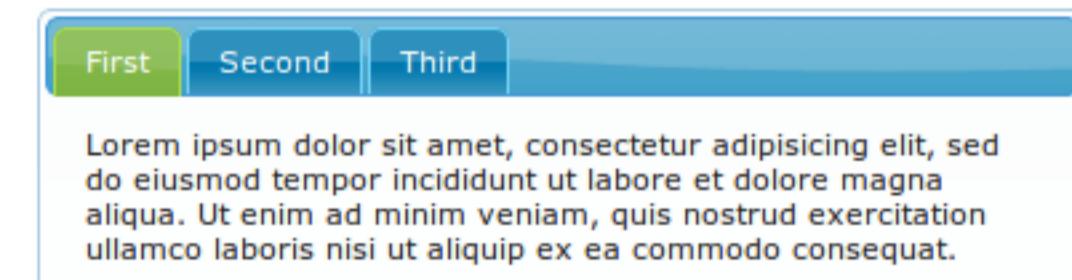
```
<input class="hasDatepicker" id="form:j_idt16:input" name="form:j_idt16:input" type="text">

<script type="text/javascript" src="/composite-demo/javax.faces.resource/jsf.js.jsf?ln=javax.faces"></script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery-1.7.2.min.js.jsf?ln=composite/com.jquery">
</script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery.ui.core.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery.ui.widget.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript"
    src="/composite-demo/javax.faces.resource/jquery.ui.datepicker.js.jsf?ln=composite/com.jqueryui/development-bundle/ui">
</script>
<script type="text/javascript">
var pluginOptions = {
    buttonImage:"/composite-demo/javax.faces.resource/calendar.gif.jsf?ln=composite/org.richfaces",
    buttonImageOnly:"true"
}
! 'both' || (pluginOptions.showOn = 'both');
! 'yy-mm-dd' || (pluginOptions.dateFormat = 'yy-mm-dd');

jQuery(function () {
    $document.getElementById('form:j_idt16:input').datepicker(pluginOptions);
});
</script>
```

JQUERY UI TABS

- The jQuery UI tabs plugin
- A stateful component



TABS COMPOSITE COMPONENT

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITabs">
</cc:interface>

<cc:implementation>
<div id="#{cc.clientId}">
  <h:inputHidden id="selected" value="#{cc.selected}" />
  <ul>
    <ui:repeat var="tab" value="#{cc.tabChildren}"> <!-- #{cc.children} broken see: JAVASERVERFACES-2099 -->
      <li>
        <a href="#{tab.clientId}">#{tab.attributes['title']}
      </li>
    </ui:repeat>
  </ul>
  <cc:insertChildren />
</div>
<h:outputScript library="javax.faces" name="jsf.js"/>
<h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.tabs.js"/>
<h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
<h:outputScript>
  var pluginOptions = { selected: '#{cc.selected}' }

  $(function() {
    var widget = $(document.getElementById('#{cc.clientId}'));
    // initialize the plugin
    widget.tabs(pluginOptions);

    // register a tabsselect change listener
    widget.bind("tabsselect", function(event, ui) {
      var selected_input = '#{cc.clientId}' + ':selected';
      $(document.getElementById(selected_input)).val(ui.index);
      // trigger an ajax update, executing the hiddn input element to update the component state
      var options = { execute: selected_input }
      jsf.ajax.request(selected_input, null, options);
    });
  });
</h:outputScript>
</cc:implementation>
```

TABS COMPOSITE COMPONENT

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITabs">
</cc:interface>
<cc:implementation>

<div id="#{cc.clientId}">
    <h:inputHidden id="selected" value="#{cc.selected}" />
    <ul>
        <ui:repeat var="tab" value="#{cc.tabChildren}"> <!-- #{cc.children} broken see: JAVASERVERFACES-2099
-->
        <li>
            <a href="#">
        </li>
    </ui:repeat>
</ul>
<cc:insertChildren />
</div>

<h:outputScript library="javax.faces" name="jsf.js"/>
<h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.tabs.js"/>
<h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
<h:outputScript>
    var pluginOptions = { selected: '#{cc.selected}' }

    ${function() {
        var widget = ${document.getElementById('#{cc.clientId}')};

        // initialize the plugin
        widget.tabs(pluginOptions);

        // register a tabselect change listener
        widget.bind("tabselect", function(event, ui) {
            var selected_input = '#{cc.clientId}' + ':selected';
            ${document.getElementById(selected_input)}.val(ui.index);
            // trigger an ajax update, executing the hiddn input element to update the component state
            var options = { execute: selected_input }
            jsf.ajax.request(selected_input, null, options);
        });
    }};
</h:outputScript>
</cc:implementation>
```

TABS COMPOSITE COMPONENT

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITabs">
</cc:interface>

<cc:implementation>
<div id="#{cc.clientId}">
<h:inputHidden id="selected" value="#{cc.selected}" />
<ul>
<ui:repeat var="tab" value="#{cc.tabChildren}"><!-- #{cc.children} broken see: JAVASERVERFACES-2099 -->
<li>
<a href="#{tab.clientId}">#{tab.attributes['title']}
</li>
</ui:repeat>
<cc:insertChildren />
</div>

<h:outputScript library="javax.faces" name="jsf.js"/>
<h:outputScript library="composite/com.jquery" name="jquery-1.7.2.min.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
<h:outputScript library="composite/com.jqueryui/development-bundle/ui" name="jquery.ui.tabs.js"/>
<h:outputStylesheet library="composite/com.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>

<h:outputScript>
var pluginOptions = { selected: '#{cc.selected}' };

$(function() {
    var widget = $(document.getElementById('#{cc.clientId}'));
    // initialize the plugin
    widget.tabs(pluginOptions);

    // register a tabselect change listener
    widget.bind("tabsselect", function(event, ui) {
        var selected_input = '#{cc.clientId}' + ':selected';
        $(document.getElementById(selected_input)).val(ui.index);
        // trigger an ajax update, executing the hiddn input element to update the component state
        var options = { execute: selected_input }
        jsf.ajax.request(selected_input, null, options);
    });
});
</h:outputScript>
</cc:implementation>
```

TABS COMPOSITE COMPONENT

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITabs">
</cc:interface>

<cc:implementation>
<div id="#{cc.clientId}">
  <h:inputHidden id="selected" value="#{cc.selected}" />
  <ul>
    <ui:repeat var="tab" value="#{cc.tabChildren}"> <!-- #{cc.children} broken see: JAVASERVERFACES-2099 -->
      <li>
        <a href="#{tab.clientId}">#{tab.attributes['title']}
      </li>
    </ui:repeat>
  <cc:insertChildren />
</div>

var pluginOptions = { selected: '#{cc.selected}' }

$(function() {
  var widget = $(document.getElementById('#{cc.clientId}'));

  // initialize the plugin
  widget.tabs(pluginOptions);

  // register a tabsselect change listener
  widget.bind("tabsselect", function(event, ui) {
    var selected_input = '#{cc.clientId}' + ':selected';
    $(document.getElementById(selected_input)).val(ui.index);
    // trigger an ajax update, executing the hiddn input element to update the component
    state
    var options = { execute: selected_input }
    jsf.ajax.request(selected_input, null, options);
  });
});

</h:outputScript>
</cc:implementation>
```

TABS COMPOSITE COMPONENT

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITabs">
</cc:interface>

<cc:implementation>
<div id="#{cc.clientId}">
<h:inputHidden id="selected" value="#{cc.selected}" />
<ul>
<ui:repeat var="tab" value="#{cc.tabChildren}"> <!-- #{cc.children} broken see: JAVASERVERFACES-2099 -->
<li>
<a href="#{tab.clientId}">#{tab.attributes['title']}
</li>
</ui:repeat>
</ul>
<cc:insertChildren />
</div>
<h:outputScript library="javax.faces" name="jsf.js"/>
<h:outputScript library="composite/con.jquery" name="jquery-1.7.2.min.js"/>
<h:outputScript library="composite/con.jqueryui/development-bundle/ui" name="jquery.ui.core.js"/>
<h:outputScript library="composite/con.jqueryui/development-bundle/ui" name="jquery.ui.widget.js"/>
<h:outputScript library="composite/con.jqueryui/development-bundle/ui" name="jquery.ui.tabs.js"/>
<h:outputStylesheet library="composite/con.jqueryui/css/ui-lightness" name="jquery-ui.custom.css"/>
<h:outputScript>
var pluginOptions = { selected: '#{cc.selected}' }

$(function() {
    var widget = $(document.getElementById('#{cc.clientId}'));
    // initialize the plugin
    widget.tabs(pluginOptions);

    // register a tabselect change listener
    widget.bind("tabsselect", function(event, ui) {
        var selected_input = '#{cc.clientId}' + ':selected';
        $(document.getElementById(selected_input)).val(ui.index);
        // trigger an ajax update, executing the hiddn input element to update the component state
        var options = { execute: selected_input }
        jsf.ajax.request(selected_input, null, options);
    });
});
</h:outputScript>
</cc:implementation>
```

UITABS.JAVA

```
@FacesComponent("org.richfaces.sandbox.composite.UITabs")
public class UITabs extends UINamingContainer {
    enum PropertyKeys {selected}

    public int getSelected() {
        return (Integer) getStateHelper().eval(PropertyKeys.selected, 0);
    }

    public void setSelected(int selected) {
        getStateHelper().put(PropertyKeys.selected, selected);
    }

    public List<UIComponent> getTabChildren() {
        List<UIComponent> children = this.getFacet("javax.faces.component.COMPOSITE_FACET_NAME").getChildren();
        List<UIComponent> tabChildren = new ArrayList<UIComponent>();
        for (UIComponent child : children) {
            if (child instanceof UITab) {
                tabChildren.add(child);
            }
        }
        return tabChildren;
    }
}
```

UITABS.JAVA

```
@FacesComponent("org.richfaces.sandbox.composite.UITabs")
public class UITabs extends UINamingContainer {
    enum PropertyKeys {selected}

    public int getSelected() {
        return (Integer) getStateHelper().eval(PropertyKeys.selected, 0);
    }

    public void setSelected(int selected) {
        getStateHelper().put(PropertyKeys.selected, selected);
    }

    public List<UIComponent> getTabChildren() {
        List<UIComponent> children = this.getFacet("javax.faces.component.COMPOSITE_FACET_NAME").getChildren();
        List<UIComponent> tabChildren = new ArrayList<UIComponent>();
        for (UIComponent child : children) {
            if (child instanceof UITab) {
                tabChildren.add(child);
            }
        }
        return tabChildren;
    }
}
```

UITABS.JAVA

```
@FacesComponent("org.richfaces.sandbox.composite.UITabs")
public class UITabs extends UINamingContainer {
    enum PropertyKeys {selected}

    public int getSelected() {
        return (Integer) getStateHelper().eval(PropertyKeys.selected, 0);
    }

    public void setSelected(int selected) {
        getStateHelper().put(PropertyKeys.selected, selected);
    }

    public List<UIComponent> getTabChildren() {
        List<UIComponent> children = this.getFacet("javax.faces.component.COMPOSITE_FACET_NAME").getChildren();
        List<UIComponent> tabChildren = new ArrayList<UIComponent>();
        for (UIComponent child : children) {
            if (child instanceof UITab) {
                tabChildren.add(child);
            }
        }
        return tabChildren;
    }
}
```

UITABS.JAVA

```
@FacesComponent("org.richfaces.sandbox.composite.UITabs")
public class UITabs extends UINamingContainer {
    enum PropertyKeys {selected}

    public int getSelected() {
        return (Integer) getStateHelper().eval(PropertyKeys.selected, 0);
    }

    public void setSelected(int selected) {
        getStateHelper().put(PropertyKeys.selected, selected);
    }

    public List<UIComponent> getTabChildren() {
        List<UIComponent> children = this.getFacet("javax.faces.component.COMPOSITE_FACET_NAME").getChildren();
        List<UIComponent> tabChildren = new ArrayList<UIComponent>();
        for (UIComponent child : children) {
            if (child instanceof UITab) {
                tabChildren.add(child);
            }
        }
        return tabChildren;
    }
}
```

JQUERY UI TAB

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITab">
    <cc:attribute name="title" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}">
        <cc:insertChildren />
    </div>
</cc:implementation>
```

JQUERY UI TAB

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITab">
    <cc:attribute name="title" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}">
        <cc:insertChildren />
    </div>
</cc:implementation>
```

JQUERY UI TAB

```
<cc:interface componentType="org.richfaces.sandbox.composite.UITab">
    <cc:attribute name="title" />
</cc:interface>

<cc:implementation>
    <div id="#{cc.clientId}">
        <cc:insertChildren />
    </div>
</cc:implementation>
```

TABS USAGE

```
<s:tabs id="tabs">

    <s:tab title="Tab 1">
        Hello <b>Tab</b>!!
    </s:tab>

    <s:tab title="Tab 2">
        With nested components:
        <br/>
        <s:datepicker dateFormat="yy-mm-dd"
                      showOn="both"
                      buttonImageOnly="true"/>
    </s:tab>

</s:tabs>
```

TABS USAGE

```
<s:tabs id="tabs">
  <s:tab title="Tab 1">
    Hello <b>Tab</b>!!
  </s:tab>

  <s:tab title="Tab 2">
    With nested components:
    <br/>
    <s:datepicker dateFormat="yy-mm-dd"
      showOn="both"
      buttonImageOnly="true"/>
  </s:tab>

</s:tabs>
```

TABS USAGE

```
<s:tabs id="tabs">  
    <s:tab title="Tab 1">  
        Hello <b>Tab</b>!!  
    </s:tab>  
  
    <s:tab title="Tab 2">  
        With nested components:  
        <br/>  
        <s:datepicker dateFormat="yy-mm-dd"  
                      showOn="both"  
                      buttonImageOnly="true"/>  
    </s:tab>  
  
</s:tabs>
```

TABS USAGE

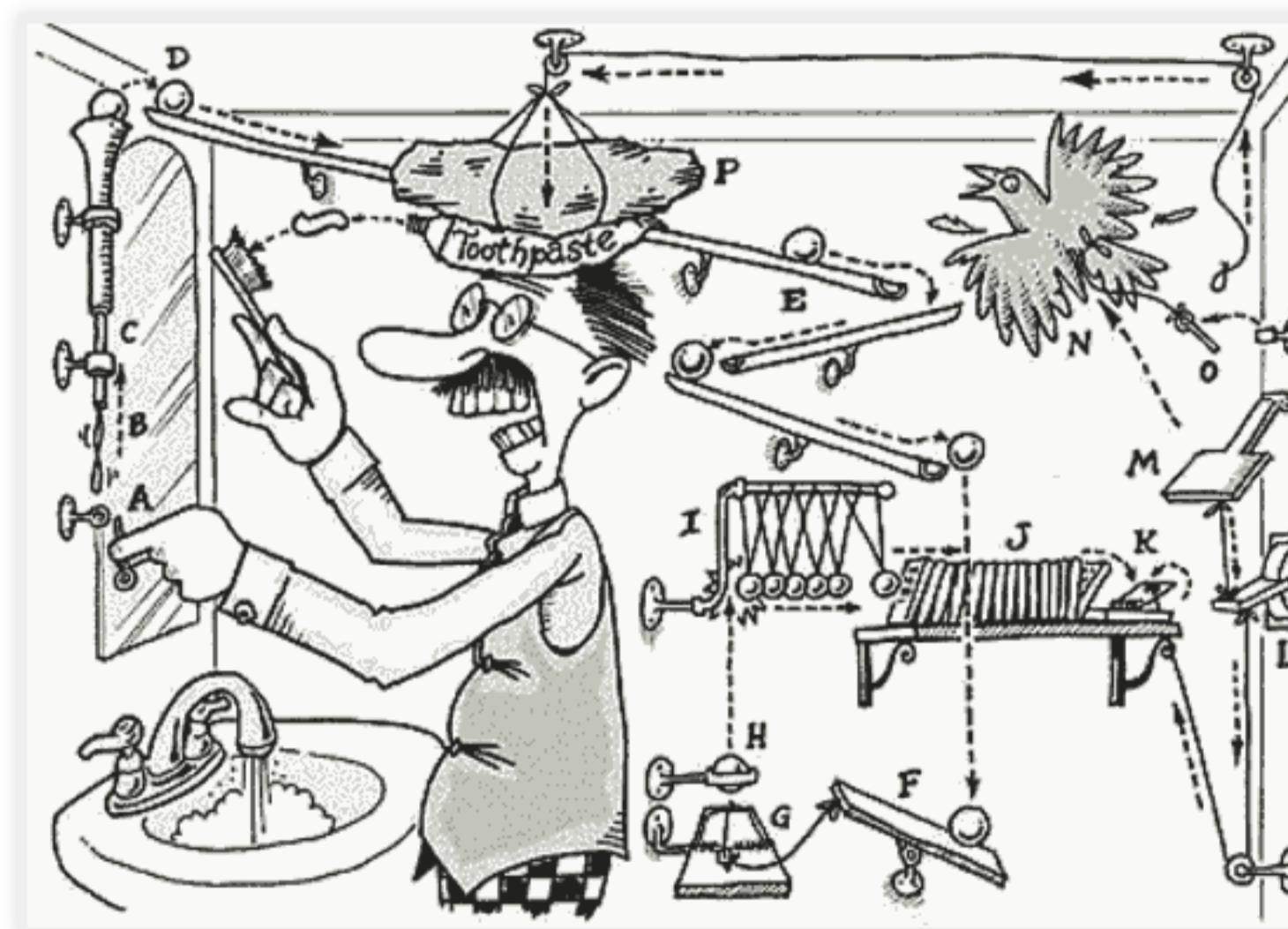
```
<s:tabs id="tabs">

    <s:tab title="Tab 1">
        Hello <b>Tab</b>!!
    </s:tab>

    <s:tab title="Tab 2">
        With nested components:
        <br/>
        <s:datepicker dateFormat="yy-mm-dd"
                      showOn="both"
                      buttonImageOnly="true"/>
    </s:tab>

</s:tabs>
```

JSF 2 CC LIMITATIONS



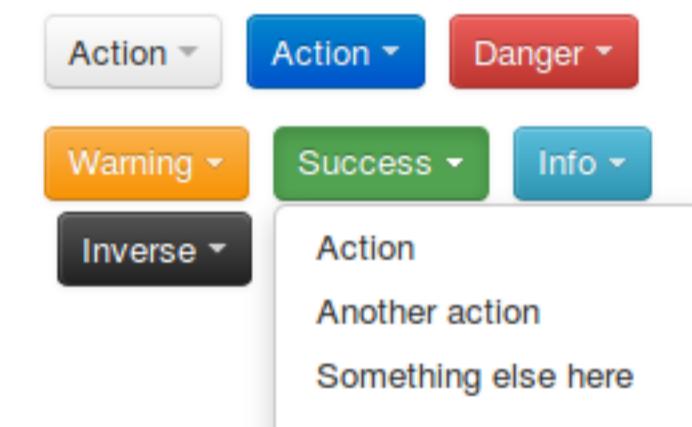
RICHFACES CDK

Use the RichFaces CDK to build "full-fledged" JSF components with the ease and syntax of JSF 2 composite components



BOOTSTRAP DROPODOWN BUTTON

- Group child buttons into a dropdown
- Dropdown renders it's children



DROPDOWN - CDK TEMPLATE

```
<cdk:root xmlns="http://jboss.org/schema/richfaces/cdk/xhtml-el"
    xmlns:cdk="http://jboss.org/schema/richfaces/cdk/core"
    xmlns:c="http://jboss.org/schema/richfaces/cdk/jstl/core"
    xmlns:cc="http://jboss.org/schema/richfaces/cdk/jsf/composite">

    <cc:interface>
        <cdk:class>org.richfaces.bootstrap.renderkit.ButtonDropdownRenderer</cdk:class>
        <cdk:superclass>org.richfaces.bootstrap.renderkit.ButtonDropdownRendererBase</cdk:superclass>
        <cdk:renderer-type>org.richfaces.bootstrap.ButtonDropdownRenderer</cdk:renderer-type>
        <cdk:renders-children>true</cdk:renders-children>
        <cdk:component-base-class>org.richfaces.bootstrap.component.AbstractButtonDropdown</cdk:component-base-class>
    </cc:interface>

    <cc:implementation>
        <div class="btn-group #{component.vertical ne null ? component.vertical.buttonClass : ''}">
            <a data-toggle="dropdown" href="#" cdk:passThrough="on">
                <span class="btn #{component.severity ne null ? component.severity.buttonClass : ''}#{component.scale ne null ? component.scale.buttonStyleClass : ''}#{component.buttonStyle} dropdown-toggle">
                    #{component.title.concat(' ')}
                    <span class="caret"></span>
                </span>
            </a>
            <ul class="dropdown-menu #{component.horizontal ne null ? component.horizontal.buttonClass : ''}">
                <c:forEach var="child" items="#{component.getChildren()}">
                    <li>
                        <cdk:call expression="child.encodeAll(facesContext)" />
                    </li>
                </c:forEach>
            </ul>
        </div>
    </cc:implementation>
</cdk:root>
```

DROPDOWN - CDK TEMPLATE

```
<cdk:root xmlns="http://jboss.org/schema/richfaces/cdk/xhtml-el"
           xmlns:cdk="http://jboss.org/schema/richfaces/cdk/core"
           xmlns:c="http://jboss.org/schema/richfaces/cdk/jstl/core"
           xmlns:cc="http://jboss.org/schema/richfaces/cdk/jsf/composite">

    <cc:interface>
        <cdk:class>org.richfaces.bootstrap.renderkit.ButtonDropdownRenderer</cdk:class>
        <cdk:superclass>org.richfaces.bootstrap.renderkit.ButtonDropdownRendererBase</cdk:superclass>
        <cdk:renderer-type>org.richfaces.bootstrap.ButtonDropdownRenderer</cdk:renderer-type>
        <cdk:renders-children>true</cdk:renders-children>
        <cdk:component-base-class>org.richfaces.bootstrap.component.AbstractButtonDropdown</cdk:component-base-class>
    </cc:interface>

    <cc:implementation>
        <div class="btn-group #{component.vertical ne null ? component.vertical.buttonClass : ''}">
            <a data-toggle="dropdown" href="#" cdk:passThrough="on">
                <span class="btn #{component.severity ne null ? component.severity.buttonClass : ''}>
                    #{component.scale ne null ? component.scale.buttonStyleClass : ''}
                    #{component.buttonStyle} dropdown-toggle">
                    #{component.title.concat(' ')}
                    <span class="caret"></span>
                </span>
            </a>
            <ul class="dropdown-menu #{component.horizontal ne null ? component.horizontal.buttonClass : ''}">
                <c:forEach var="child" items="#{component.getChildren()}">
                    <li>
                        <cdk:call expression="child.encodeAll(facesContext)" />
                    </li>
                </c:forEach>
            </ul>
        </div>
    </cc:implementation>
</cdk:root>
```

DROPDOWN - CDK TEMPLATE

```
<cdk:root xmlns="http://jboss.org/schema/richfaces/cdk/xhtml-el"
    xmlns:cdk="http://jboss.org/schema/richfaces/cdk/core"
    xmlns:c="http://jboss.org/schema/richfaces/cdk/jstl/core"
    xmlns:cc="http://jboss.org/schema/richfaces/cdk/jsf/composite">

<cc:interface>
    <cdk:class>org.richfaces.bootstrap.renderkit.ButtonDropdownRenderer</cdk:class>
    <cdk:superclass>org.richfaces.bootstrap.renderkit.ButtonDropdownRendererBase</cdk:superclass>
    <cdk:renderer-type>org.richfaces.bootstrap.ButtonDropdownRenderer</cdk:renderer-type>
    <cdk:renders-children>true</cdk:renders-children>
    <cdk:component-base-class>org.richfaces.bootstrap.component.AbstractButtonDropdown</cdk:component-base-class>
</cc:interface>

<cc:implementation>
    <div class="btn-group #{component.vertical ne null ? component.vertical.buttonClass : ''}">
        <a data-toggle="dropdown" href="#" cdk:passThrough="on">
            <span class="btn #{component.severity ne null ? component.severity.buttonClass : ''}#{component.scale ne null ? component.scale.buttonStyleClass : ''}#{component.buttonStyle} dropdown-toggle">
                #{component.title.concat(' ')}
                <span class="caret"></span>
            </span>
        </a>
        <ul class="dropdown-menu #{component.horizontal ne null ? component.horizontal.buttonClass : ''}">
            <c:forEach var="child" items="#{component.getChildren()}">
                <li>
                    <cdk:call expression="child.encodeAll(facesContext)" />
                </li>
            </c:forEach>
        </ul>
    </div>
</cc:implementation>
</cdk:root>
```

DROPDOWN - CDK TEMPLATE

```
<cdk:root xmlns="http://jboss.org/schema/richfaces/cdk/xhtml-el"
    xmlns:cdk="http://jboss.org/schema/richfaces/cdk/core"
    xmlns:c="http://jboss.org/schema/richfaces/cdk/jstl/core"
    xmlns:cc="http://jboss.org/schema/richfaces/cdk/jsf/composite">

    <cc:interface>
        <cdk:class>org.richfaces.bootstrap.renderkit.ButtonDropdownRenderer</cdk:class>
        <cdk:superclasses>com.richfaces.bootstrap.renderkit.ButtonDropdownRendererBase</cdk:superclasses>
    </cc:interface>

    <cc:implementation>
        <div class="btn-group #{component.vertical ne null ? component.vertical.buttonClass : ''}">
            <a data-toggle="dropdown" href="#" cdk:passThrough="on*"
                class="btn #{component.severity ne null ? component.severity.buttonClass : ''}
                    #{component.scale ne null ? component.scale.buttonStyleClass : ''}
                    #{component.buttonStyle} dropdown-toggle">
                #{component.title.concat(' ')}
                <span class="caret"></span>
            </a>
            <ul class="dropdown-menu #{component.horizontal ne null ? component.horizontal.buttonClass
                : ''}">
                <c:forEach var="child" items="#{component.getChildren()}">
                    <li>
                        <cdk:call expression="child.encodeAll(facesContext)" />
                    </li>
                </c:forEach>
            </ul>
        </div>
    </cc:implementation>
</cdk:root>
```

DROPDOWN - CDK TEMPLATE

```
<cdk:root xmlns="http://jboss.org/schema/richfaces/cdk/xhtml-el"
    xmlns:cdk="http://jboss.org/schema/richfaces/cdk/core"
    xmlns:c="http://jboss.org/schema/richfaces/cdk/jstl/core"
    xmlns:cc="http://jboss.org/schema/richfaces/cdk/jsf/composite">

    <cc:interface>
        <cdk:class>org.richfaces.bootstrap.renderkit.ButtonDropdownRenderer</cdk:class>
        <cdk:superclass>org.richfaces.bootstrap.renderkit.ButtonDropdownRendererBase</cdk:superclass>
        <cdk:renderer-type>org.richfaces.bootstrap.ButtonDropdownRenderer</cdk:renderer-type>
        <cdk:renders-children>true</cdk:renders-children>
        <cdk:component-base-class>org.richfaces.bootstrap.component.AbstractButtonDropdown</cdk:component-base-class>
    </cc:interface>

    <cc:implementation>
        <div class="btn-group #{component.vertical ne null ? component.vertical.buttonClass : ''}">
            <a data-toggle="dropdown" href="#" cdk:passThrough="on">
                <span class="btn #{component.severity ne null ? component.severity.buttonClass : ''}#{component.scale ne null ? component.scale.buttonStyleClass : ''}#{component.buttonStyle} dropdown-toggle">
                    #{component.title.concat(' ')}
                    <span class="caret"></span>
                </span>
            </a>
            <ul class="dropdown-menu #{component.horizontal ne null ? component.horizontal.buttonClass : ''}">
                <c:forEach var="child" items="#{component.getChildren()}">
                    <li>
                        <cdk:call expression="child.encodeAll(facesContext)" />
                    </li>
                </c:forEach>
            </ul>
        </div>
    </cc:implementation>
</cdk:root>
```

DROPDOWN BUTTON - COMPONENT

```
@JsfComponent(  
    type = AbstractButtonDropdown.COMPONENT_TYPE,  
    family = AbstractButtonDropdown.COMPONENT_FAMILY,  
    renderer = @JsfRenderer(type = ButtonDropdownRendererBase.RENDERER_TYPE),  
    tag = @Tag(name = "buttonDropdown"))  
public abstract class AbstractButtonDropdown extends UIPanel implements EventsMouseProps {  
    public static final String COMPONENT_FAMILY = "org.richfaces.bootstrap.ButtonDropdown";  
    public static final String COMPONENT_TYPE = "org.richfaces.bootstrap.ButtonDropdown";  
  
    @Attribute  
    public abstract String getTitle();  
  
    @Attribute  
    public abstract String getButtonStyle();  
  
    @Attribute  
    public abstract BootstrapSeverity getSeverity();  
  
    @Attribute  
    public abstract BootstrapSize getScale();  
  
    @Attribute  
    public abstract HorizontalPosition getHorizontal();  
  
    @Attribute  
    public abstract VerticalPosition getVertical();  
}
```

DROPDOWN - USAGE

```
<b:buttonDropdown title="Primary" severity="primary">
    <h:link value="heroUnit"
        outcome="/component/heroUnit/index.xhtml" />
    <h:link value="tabbable"
        outcome="/component/tabbable/index.xhtml"/>
    <h:link value="buttonGroup"
        outcome="/component/buttonGroup/index.xhtml"/>
    <h:link value="navbar"
        outcome="/component/navbar/index.xhtml"/>
</b:buttonDropdown>
```

LINKS

RichFaces Sandbox on github

Blog Collapsible panel w/ Composite Components

JSF Composite component API docs:

- composite Tag interface
- composite Tag implementation

Javascript "widgets"

- jQuery UI
- Twitter Bootstrap
- jQuery-Image-Gallery