

Composing Components with Facelets

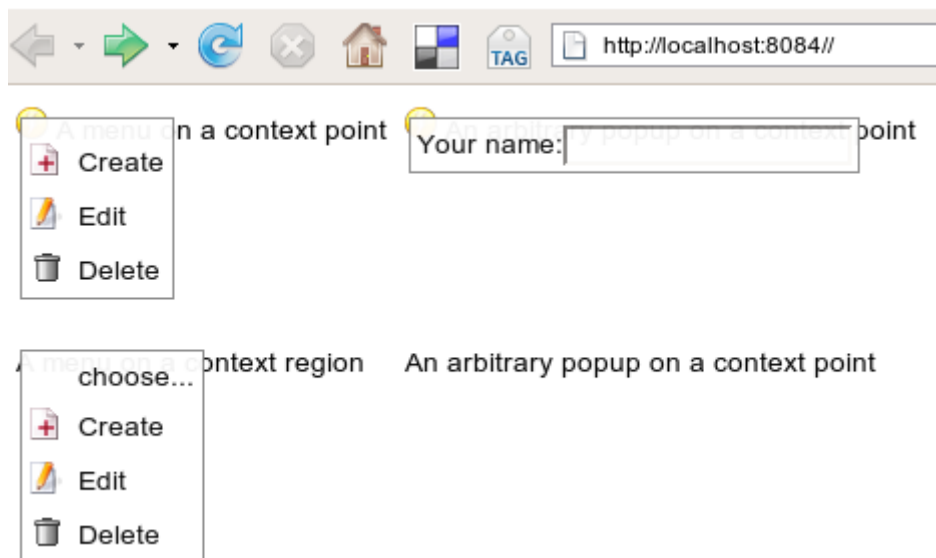
By [Roger Keays](#), 11 December 2006

One of my favourite features of Facelets is how compositions of JSF components can be arranged and reused in various ways. The icing on the cake is, of course, that as well as including and decorating these compositions, you can put them in tag files and use them as if they were native JSF components. Not only this, but your tag files can also be packaged into a jar and reused across all your projects. It's like the instant noodles of JSF components!

Well, that's a lot of terminology to throw around in the first paragraph of a blog (I hope I got it all right) so lets have a look at an example so you can see what I mean.

What we're going to do here is build a quick, but quite flexible popup menu made entirely out of Facelets tag files. Our menu will be activated when you right-click on either a region of text, or an image which indicates the menu is available. Also, we want to leave the possibility open that the context menu might be used to display something other than a menu (a form for example).

Here's a sneak peak of the result, so you can get the idea of what I'm talking about.



Our menu relies mostly on css styles to show and hide it, plus a little bit of javascript to apply those styles. The css is basically just this (although we'll add some other styles to make it look acceptable):

```
.fb-context-box { display: none; position: absolute; }  
.fb-context-box-open { display: block; }
```

To meet all our requirements, we're going to create the following 'components':

Component	Description
contextRegion	Renders an area which, when right-clicked, activates a contextBox or contextMenu.

	The activated component must be a child of this contextRegion.
contextPoint	This operates in a similar way to a contextRegion, except it renders a small menu icon which opens the contextBox or contextMenu on right click.
contextBox	This component renders a div for a generic context popup box. It also renders javascript to deactivate the popup on the mouseout event.
contextMenu	This is just a panelGrid to arrange menu items in a context menu. It wraps itself in a contextBox and would normally appear as part of a contextRegion or contextPoint.
contextMenuItem	Single item for the contextMenu, with an optional icon.

You can have a look at the [attached sample webapp](#) if you want to see the source of those components. It's just a bunch of divs and panelGroups with the odd graphicImage thrown in. To make them available to Facelets, we create a taglib file in WEB-INF which looks something like this:

```
<facelet-taglib>
  <namespace>https://rogerkeays.com/blog/composing_components</namespace>
  <tag>
    <tag-name>contextBox</tag-name>
    <source>tags/contextBox.xhtml</source>
  </tag>
  <tag>
    <tag-name>contextMenu</tag-name>
    <source>tags/contextMenu.xhtml</source>
  </tag>
  ....
</facelet-taglib>
```

Facelets will search the classpath for *.taglib.xml files, but in this case we haven't put ours on the classpath, so we just reference it in web.xml:

```
<context-param>
  <param-name>facelets.LIBRARIES</param-name>
  <param-value>/WEB-INF/contextMenus.taglib.xml</param-value>
</context-param>
```

Voila! That's it. Our components are ready to use. I created the menu's in the screenshot using the following code:

```
<h:panelGrid columns="2">
  <h:panelGroup>
    <m:contextPoint>
```

```

        <m:contextMenu>
            <m:contextMenuItem image="icons/create.png">Create</m:contextMenuItem>
            <m:contextMenuItem image="icons/edit.png">Edit</m:contextMenuItem>
            <m:contextMenuItem image="icons/delete.png">Delete</m:contextMenuItem>
        </m:contextMenu>
    </m:contextPoint>
    <h:outputText value="A menu on a context point"/>
</h:panelGroup>

```

```

<h:panelGroup>
    <m:contextPoint>
        <m:contextBox>
            Your name: <h:inputText size="20"/>
        </m:contextBox>
        <h:outputText value="An arbitrary popup on a context point"/>
    </m:contextPoint>
</h:panelGroup>

```

```

<m:contextRegion>
    <m:contextMenu title="choose...">
        <m:contextMenuItem image="icons/create.png">Create</m:contextMenuItem>
        <m:contextMenuItem image="icons/edit.png">Edit</m:contextMenuItem>
        <m:contextMenuItem image="icons/delete.png">Delete</m:contextMenuItem>
    </m:contextMenu>
    <h:outputText value="A menu on a context region"/>
</m:contextRegion>

```

```

<m:contextRegion>
    <m:contextBox>
        Your name: <h:inputText size="20"/>
    </m:contextBox>
    <h:outputText value="An arbitrary popup on a context point"/>
</m:contextRegion>
</h:panelGrid>

```

Although we could put these components into a jar, it's only going to be of limited use since our static css and image resources won't be accessible. Rumour has it that Facelets 1.2 will have support for serving static resources out of a jar so keep an eye out for that one.

You can [download an operational war of these menus](#) if you want to have a closer look at the code, but please note that the purpose of this blog is simply to demonstrate Facelet's component composition features. These menu components are not production ready! For starters you have to

deal with IE's z-index defects properly and you'd also want to improve the way mouse exits are handled.

About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world.

Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.