

JSF - UI:PARAM TAG

http://www.tutorialspoint.com/jsf/jsf_parameters_tag.htm

Copyright © tutorialspoint.com

Using ui:param tag, we can pass parameters to template file or an included file.

In *JSF - template tags* chapter we've learned how to create and use template tags. We defined various section such as header, footer, content and a template combining all the sections.

Now we'll learn

- how to pass parameters to various section of a template
- how to pass parameters to a template

Parameter to section of a template

Create parameter : common.xhtml

Add parameter to ui:include tag. Use **ui:param** tag to define a parameter containing a value to be passed to Header section.

```
<ui:insert name="header" >
  <ui:include src="/templates/header.xhtml" >
    <ui:param name="defaultHeader" value="Default Header" />
  </ui:include>
</ui:insert>
```

Use parameter : header.xhtml

```
<ui:composition>
  <h1>#{defaultHeader}</h1>
</ui:composition>
```

Parameter to template

Create parameter : home.xhtml

Add parameter to ui:composition tag. Use **ui:param** tag to define a parameter containing a value to be passed to template.

```
<ui:composition template="templates/common.xhtml">
  <ui:param name="title" value="Home" />
</ui:composition>
```

Use parameter : common.xhtml

```
<h:body>
  <h2>#{title}</h2>
</h:body>
```

Example Application

Let us create a test JSF application to test the template tags in JSF.

Step	Description
------	-------------

- | | |
|---|---|
| 1 | Create a project with a name <i>helloworld</i> under a package <i>com.tutorialspoint.test</i> as explained in the <i>JSF - Templates Tag</i> chapter. |
|---|---|

- 2 Modify *header.xhtml*, and *common.xhtml* files under *src > main > webapp > templates* folder. Modify them as explained below.
- 3 Modify *home.xhtml* as explained below. Keep rest of the files unchanged.
- 4 Compile and run the application to make sure business logic is working as per the requirements.
- 5 Finally, build the application in the form of war file and deploy it in Apache Tomcat Webserver.
- 6 Launch your web application using appropriate URL as explained below in the last step.

header.xhtml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <body>
    <ui:composition>
      <h1>#{defaultHeader}</h1>
    </ui:composition>
  </body>
</html>
```

common.xhtml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <h:head></h:head>
  <h2>#{title}</h2>
  <div style="border-width:2px; border-color:green; border-style:solid;">
    <ui:insert name="header" >
      <ui:include src="/templates/header.xhtml" >
        <ui:param name="defaultHeader" value="Default Header" />
      </ui:include>
    </ui:insert>
  </div>
  <br/>
  <div style="border-width:2px; border-color:black; border-style:solid;">
    <ui:insert name="content" >
      <ui:include src="/templates/contents.xhtml" />
    </ui:insert>
  </div>
  <br/>
  <div style="border-width:2px; border-color:red; border-style:solid;">
    <ui:insert name="footer" >
      <ui:include src="/templates/footer.xhtml" />
    </ui:insert>
  </div>
</h:body>
</html>
```

home.xhtml

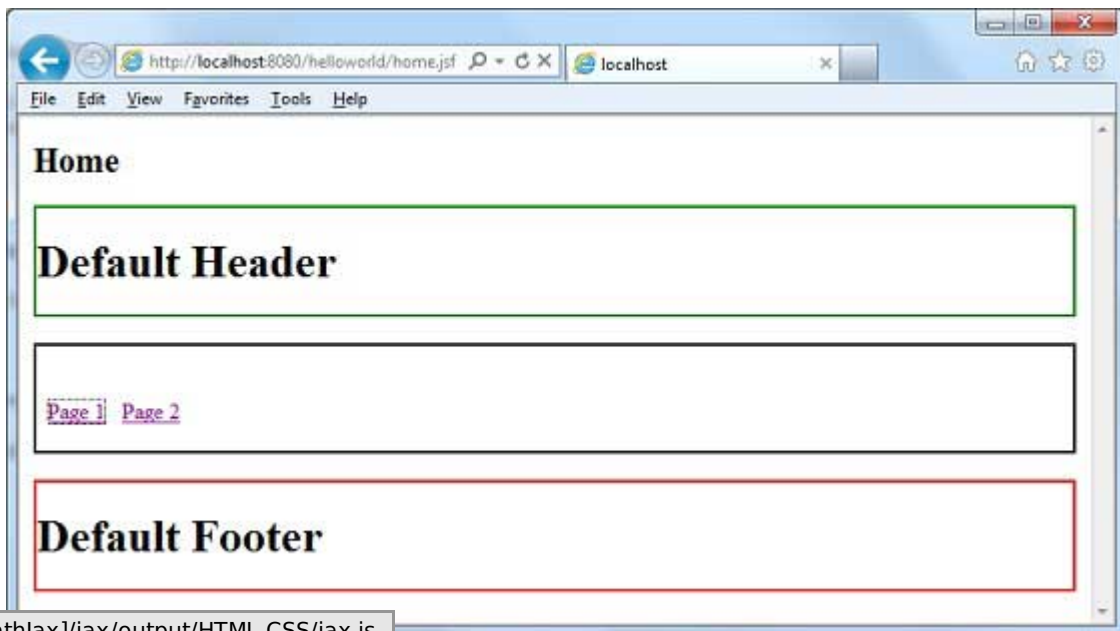
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<h:body>
  <ui:composition template="templates/common.xhtml">
    <ui:param name="title" value="Home" />
    <ui:define name="content">
      <br/><br/>
      <h:link value="Page 1" outcome="page1" />
      <h:link value="Page 2" outcome="page2" />
      <br/><br/>
    </ui:define>
  </ui:composition>
</h:body>
</html>

```

Once you are ready with all the changes done, let us compile and run the application as we did in JSF - First Application chapter. If everything is fine with your application, this will produce following result:



Loading [Mathjax]/jax/output/HTML-CSS/jax.js