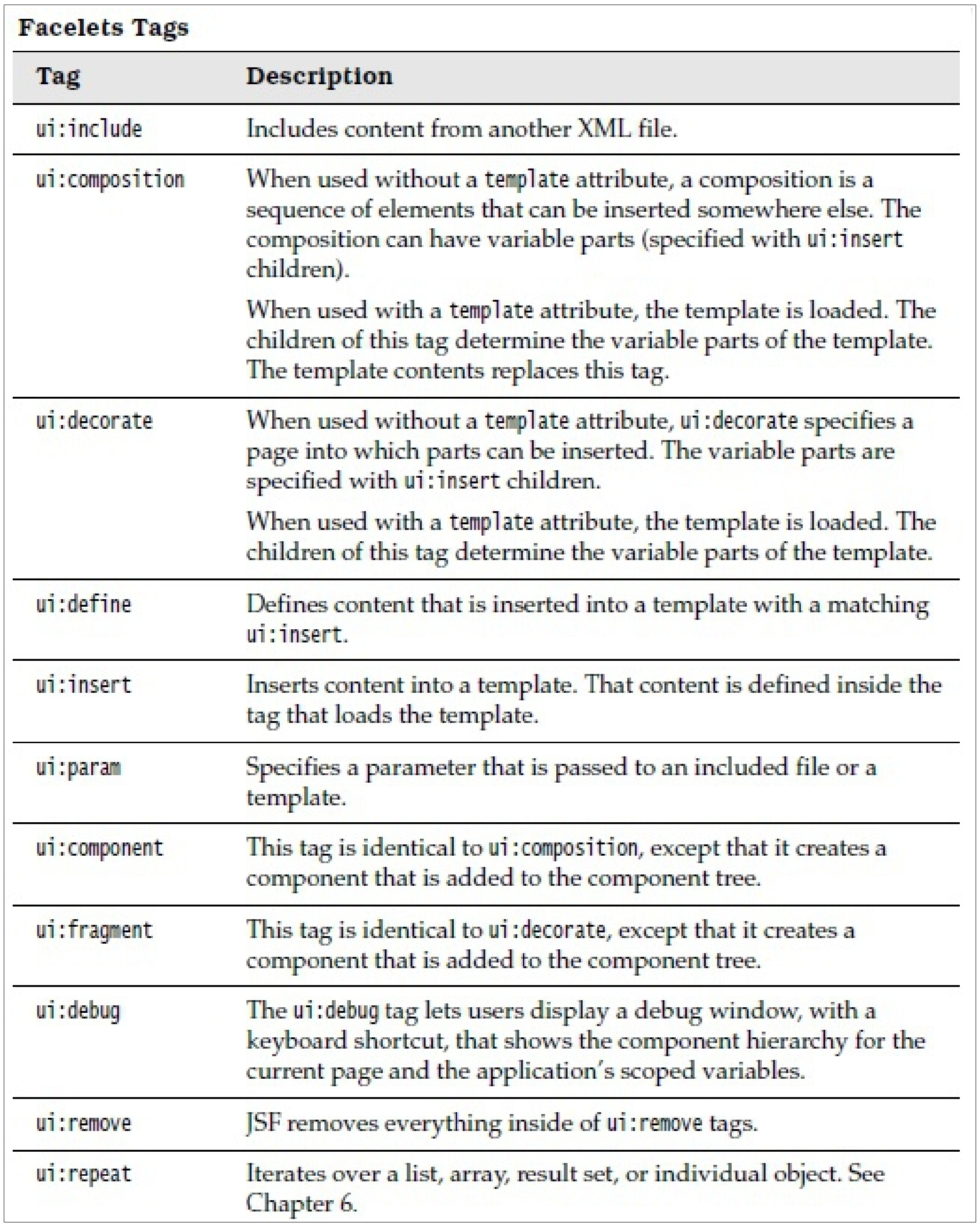
Chapter 5 Facelets

1. Facelets Tags



Facelets Tags

Tag	Description
`ui:include`	Includes content from another XML file.
`ui:composition`	When used without a `template` attribute, a composition is a sequence of elements that can be inserted somewhere else. The composition can have variable parts (specified with `ui:insert` children). When used with a `template` attribute, the template is loaded. The children of this tag determine the variable parts of the template. The template contents replaces this tag.
`ui:decorate`	When used without a `template` attribute, `ui:decorate` specifies a page into which parts can be inserted. The variable parts are specified with `ui:insert` children. When used with a `template` attribute, the template is loaded. The children of this tag determine the variable parts of the template.
`ui:define`	Defines content that is inserted into a template with a matching `ui:insert`.
`ui:insert`	Inserts content into a template. That content is defined inside the tag that loads the template.
`ui:param`	Specifies a parameter that is passed to an included file or a template.
`ui:component`	This tag is identical to `ui:composition`, except that it creates a component that is added to the component tree.
`ui:fragment`	This tag is identical to `ui:decorate`, except that it creates a component that is added to the component tree.
`ui:debug`	The `ui:debug` tag lets users display a debug window, with a keyboard shortcut, that shows the component hierarchy for the current page and the application's scoped variables.
`ui:remove`	JSF removes everything inside of `ui:remove` tags.
`ui:repeat`	Iterates over a list, array, result set, or individual object. See Chapter 6.

2. Templating with Facelets `<ui:insert>`, `<ui:composition>` and `<ui:define>`

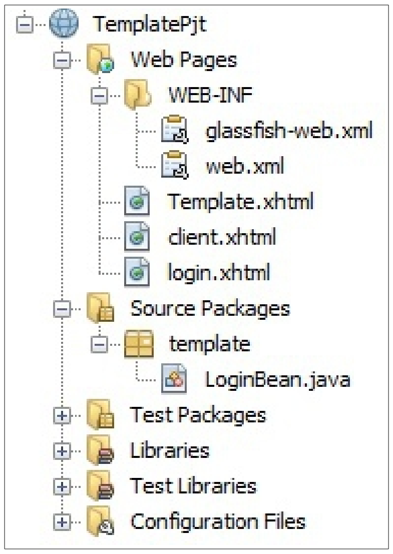
➔ Most of the websites will be have pages with similar layout and styling. They have common header, footer and sidebars etc.

➔ Facelets lets you encapsulate that commonality in a template, so that one can update the site by making changes to template than individual pages.

➔ Template Example

```
<ui:insert>
<ui:composition>
<ui:define>
```

➔ Lets have TemplatePjt directory as shown below



web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd" version="3.0">
  <display-name>TemplateProject</display-name>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <context-param>
    <description>State saving method: 'client' or 'server' (=default). See JSF Specification
2.5.2</description>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
    <param-value>resources.application</param-value>
  </context-param>
  <listener>
    <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
  </listener>
  <welcome-file-list>
  <welcome-file>Login.xhtml</welcome-file>
  </welcome-file-list>
</web-app>
```

login.xhtml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
        xmlns:c="http://java.sun.com/jsf/core"
        xmlns:ui = "http://java.sun.com/jsf/facelets"
        xmlns:h = "http://java.sun.com/jsf/html">

<h:body>
      <h:form>
        Enter name : <h:inputText id="name" value="#{LoginBean.name}"/>
        <h:commandButton id="submit" value="submit" action="client"/>
      </h:form>
</h:body>
</html>
```

template.xhtml

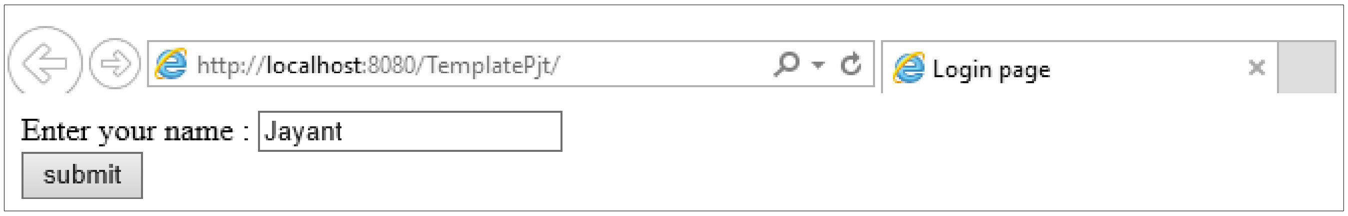
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:c="http://java.sun.com/jsf/core"
xmlns:ui = "http://java.sun.com/jsf/facelets"
xmlns:h = "http://java.sun.com/jsf/html">

<h:body>
      <ui:insert name="message"/>
</h:body>
</html>
```

client.xhtml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:c="http://java.sun.com/jsf/core"
xmlns:ui = "http://java.sun.com/jsf/facelets"
xmlns:h = "http://java.sun.com/jsf/html">

<h:body>
      <h:form>
            <ui:composition template = "/template.xhtml">

            <ui:define name = "message">
                  Hello #{LoginBean.name}
            </ui:define>

             </ui:composition>
      </h:form>
</h:body>
</html>
```

Output

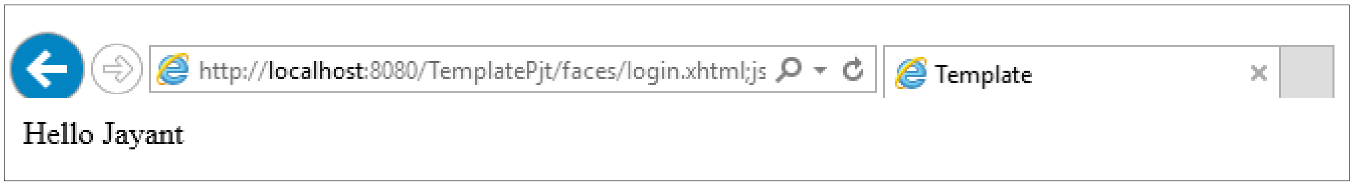
login.xhtml



Enter your name : Jayant

submit

client.xhtml



Hello Jayant

➔ Facelets removes all tags *outside* the ui:composition tag—that is, the doctype declaration, html, head, title, and body tags. This is necessary because the ui:composition is replaced with the template that contains its own set of html, head, title, and body tags.

➔ When the template is loaded, each ui:insert is replaced with the contents of the corresponding ui:define.

3. Decorators `<ui:decorate>`

➔ With decorators first content is designed and then decorated using `<ui:decorate>` tag.

➔ `<ui:decorate>` tag has 'template' attribute and it can be used in the place of `<ui:composition>`.

➔ Difference between `<ui:composition>` and `<ui:decorate>` is, anything outside `<ui:composition>` tag is disregarded but its not true with `<ui:decorate>`. Hence `<ui:decorate>` is beneficial as template-in-template.

4. Parameters `<ui:param>`

➔ In a template one can supply argument in two ways. One is through <ui:define> and other is through <ui:param>.

➔ While `<ui:define>` provides markup that inserted into the template, In constrast `<ui:param>` sets an EL variable for use in template.

➔ Ex

```
<ui:composition template="templates/masterTemplate.xhtml">
      <ui:param name="currentDate" value="#{someBean.currentDate}"/>
</ui:composition>
```

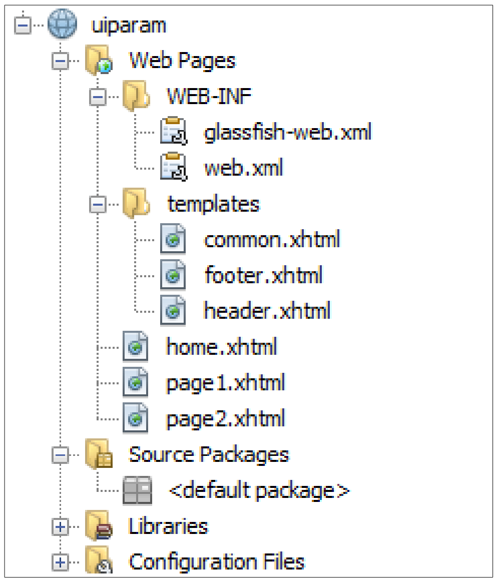
In the corresponding template, you can access the parameter with an EL

expression, like this:

```
...
<body>
Today's date: #{currentDate}"/>
</body>
...
```

➔ Ex

Lets create a 'uiparam' project directory as shown below



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>
```

footer.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>#{defaultFooter}</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>
      <h:body>
      <div style="border-width:2px; border-color:yellow; 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:param name="defaultFooter" value="Default Footer" />
            </ui:include>
         </ui:insert>
      </div>
   </h:body>
</html>
```

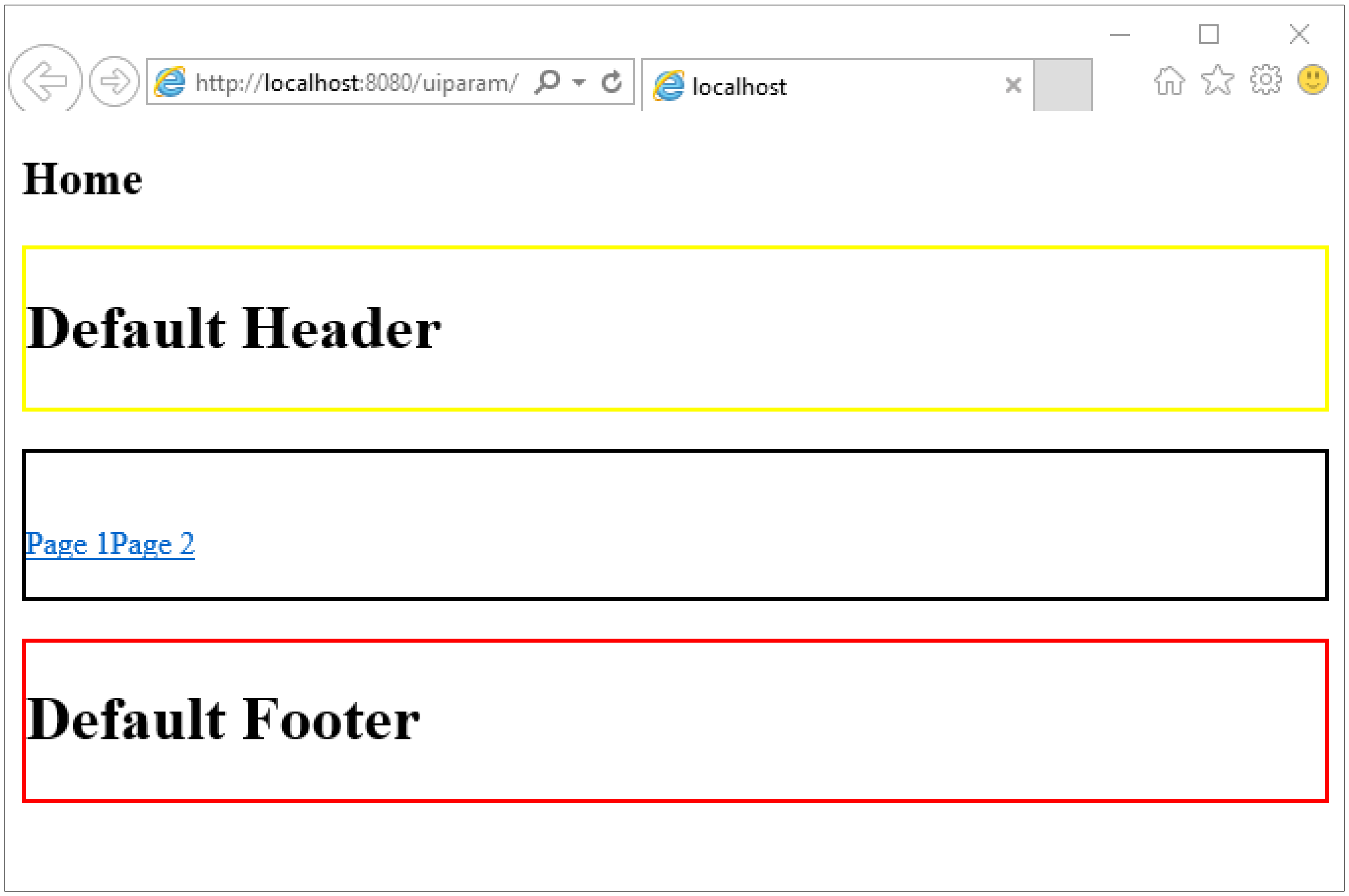
home.xhtml

```
<?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>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <context-param>
        <param-name>javax.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>/faces/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>faces/home.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

<u>Output</u>


http://localhost:8080/uiparam/

localhost

Home

Default Header

Page 1Page 2

Default Footer

Note : Page1 and Page2 view pages has not been implemented and only kept to enable links.

➔ Here we can see how <ui:param> in two ways

I. As child tag with <ui:include> both for header and footer in common.xhtml file.

Ex `<ui:param name="defaultHeader" value="Default Header" />`

ii. As child tag with <ui:composition> for title

`<ui:param name="title" value="Home" />`

5. <u>Components and Fragements `<ui:component>` and `<ui:fragement>`</u>

➔ <ui:component> tag is similar to <ui:composition> except for two things

I. JSF creates a separate component for it and adds it directly to the component tree.

ii. There is no associated template.

➔ <ui:component> tag is used to create component and specify a file name for the component as either the source of a <ui:include> (as done in the following example) or the source of a Facelets tag.

➔ <ui:compostion> is used to define template layout but <ui:component> is encapsulate definition of a facelet component.

➔ It has `id`, `binding` and `rendered` attribute. If '`binding`' attribute has been used then one can programmatically manipulate the component.

➔ Also one can conditionally render the component by setting '`rendered`' attribute to a value expression(which is a boolean value). If its false then component is not rendered in the page.

➔ Any markup occurring outside of the `<ui:component>` tag is not included in the view.

➔ Similarly <ui:fragement> is same as <ui:component> it only includes tags occurring outside itself.

6. <ui:debug> tag

➔ If <ui:debug> tag added to a facelet page then a debug component will be added to the component tree of that page.

➔ If user types hotkey, which is by default CTRL+SHIFT+d, the JSF open a window and displays the state of the component tree and applicatoin's scoped variables.

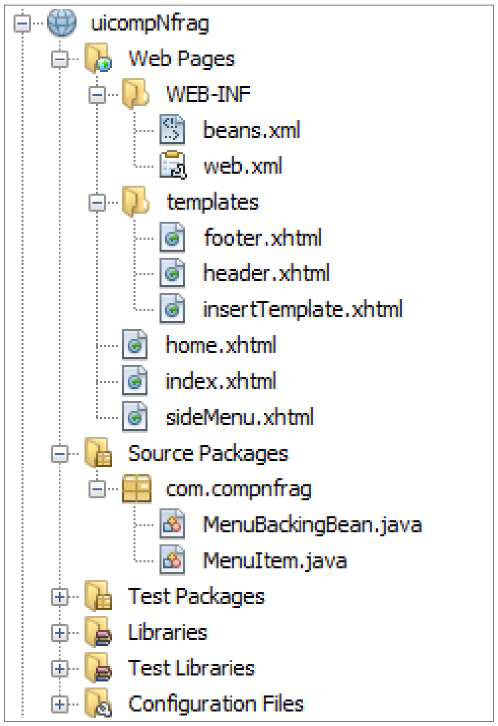
➔ One can change the hotkey using attribute 'hotkey'

Ex <ui:debug hotkey="i"/>

then the hotkey becomes CTRL+SHIFT+i.

➔ Ex uses <ui:component>, <ui:fragement>, <ui:debug>

Lets create project directory for project 'uicompNfrag' as shown below



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>
      <div style="width:100%;line-height:48px;background-color:#336699;color:white; font-
size:30px">
      JSF Facelets Application</div>
   </body>
</html>
```

footer.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">
   <head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>Insert title here</title>
</head>
<body>
<div style="background-color: #336699; width: 100%; color: #FFFFFF">@copyright,RISK
soft</div>
</body>
</html>
```

sideMenu.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"
         xmlns:h="http://java.sun.com/jsf/html"
         xmlns:c="http://xmlns.jcp.org/jsp/jstl/core">
This text will not be rendered.<br/>
<ui:component id="comp">
      <c:forEach var="menu" items="#{menuBackingBean.menus}">
           <a href="#{menu.url}">#{menu.label}</a><br/>
      </c:forEach>
</ui:component>
</html>
```

insertTemplate.xhtml

```
<!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"
xmlns:h="http://java.sun.com/jsf/html">
<head>
<title>Facelets example</title>
</head>
<body >
<div style="border-bottom: grey 2px solid; border-left: grey 2px solid; border-right: grey
2px solid; border-top: grey 2px solid; height: 100%; margin: 4px auto; text-align: center;
width: 100%;">
<ui:insert name="header">
       <ui:include src="/templates/header.xhtml" />
</ui:insert></div>
<table width="100%" >
        <tr>
                <td width="20%">
                <div style="height: 250px; width: 100%; background-color: #e0e0e0; text-
align: center;">
                 <br></br>
                <ui:insert name="sidemenu"/>

                 </div>
                 </td>
                 <td width="85%">
                 <ui:insert name="content">Content displayed from Template </ui:insert>
                 </td>
                 </tr>
</table>
```

```
<div style="border-bottom: grey 2px solid; border-left: grey 2px solid; border-right: grey
2px solid; border-top: grey 2px solid; height: 100%; margin: 4px auto; text-align: center;
width: 100%;">
<ui:insert name="footer">
       <ui:include src="/templates/footer.xhtml" />
</ui:insert></div>
    <ui:debug hotkey="i"/><!--hotkey is placed here -->
</body>
</html>
```

home.xhtml

```
<!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"
         xmlns:h="http://java.sun.com/jsf/html"
         xmlns:f="http://java.sun.com/jsf/core">
<ui:composition template="templates/insertTemplate.xhtml">
<ui:define name="sidemenu">
        <ui:include src="sideMenu.xhtml" />
</ui:define>
<ui:define name="content">
        This is an example of a simple Facelets template.<br/>
        Here Header and Footer appears from template.(inserTemplate.xhtml)<br/>
        Side Menu appears from sideMenu.xhtml<br/>
        This section appears from templateClient.(home.xhtml)
</ui:define>
</ui:composition>
</html>
```

MenuItem.java

```
package com.compnfrag;

public class MenuItem {
    private String url;
    private String label;
    public String getUrl() {
       return url;
    }
    public void setUrl(String url) {
        this.url = url;
    }
    public String getLabel() {
        return label;
    }
    public void setLabel(String label) {
        this.label = label;
    }
    public MenuItem() {
        super();
    }
    public MenuItem(String url, String label) {
        super();
        this.url = url;
        this.label = label;
    }
}
```

MenuBackingBean.java

```
package com.compnfrag;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.Collection;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;

@Named("menuBackingBean")
@SessionScoped
public class MenuBackingBean implements Serializable {
     private Collection<MenuItem> menus;
     public Collection<MenuItem> getMenus() {
         return menus;
     }
     public void setMenus(Collection<MenuItem> menus) {
         this.menus = menus;
     }
     public MenuBackingBean() {
         super();
         menus = new ArrayList<>();
         menus.add(new MenuItem("home.xhtml", "Home"));
         menus.add(new MenuItem("news.xhtml", "News"));
         menus.add(new MenuItem("articles.xhtml", "Articles"));
         menus.add(new MenuItem("about.xhtml", "About Us"));
     }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <context-param>
        <param-name>javax.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>/faces/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>faces/home.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

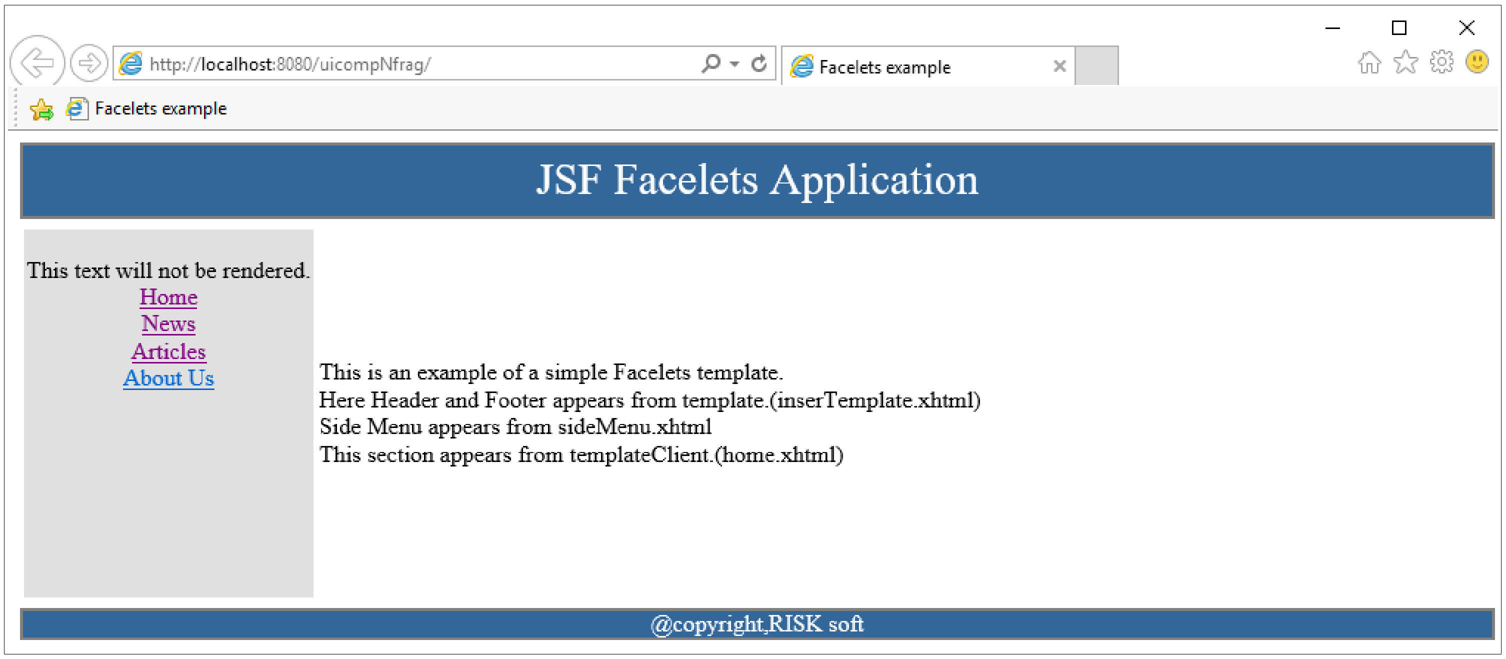
Output

1. Without <ui:component> tag

When <ui:component> tag is commented out inside sideMenu.xhtml as shown below

```
This text will not be rendered.<br/>
<ui:component id="comp">
      <c:forEach var="menu" items="#{menuBackingBean.menus}">
           <a href="#{menu.url}">#{menu.label}</a><br/>
      </c:forEach>
</ui:component>
```

home.xhtml



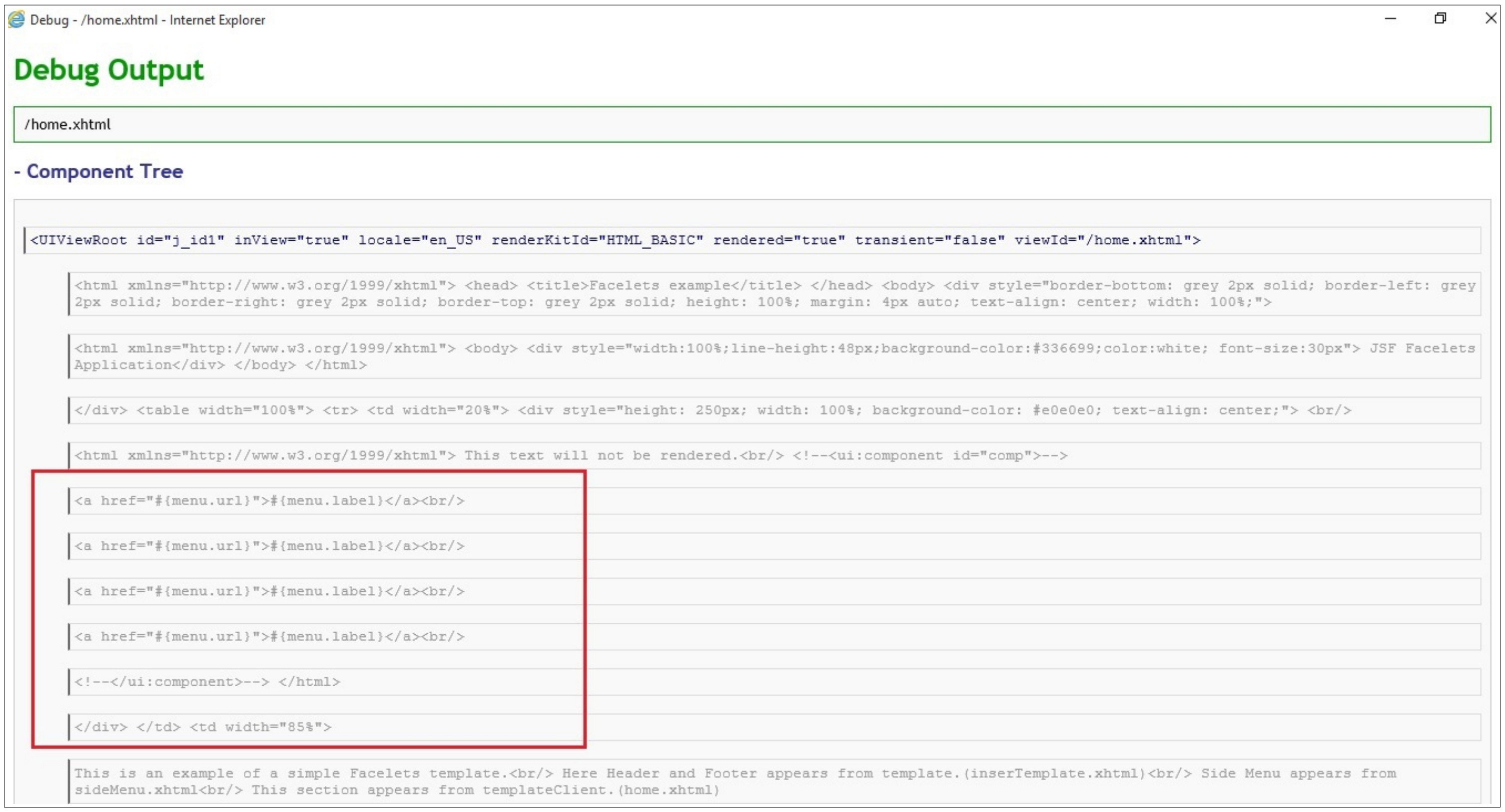
JSF Facelets Application

This text will not be rendered.
Home
News
Articles
About Us

This is an example of a simple Facelets template.
Here Header and Footer appears from template.(inserTemplate.xhtml)
Side Menu appears from sideMenu.xhtml
This section appears from templateClient.(home.xhtml)

@copyright,RISK soft

Debug



Debug Output

/home.xhtml

- Component Tree

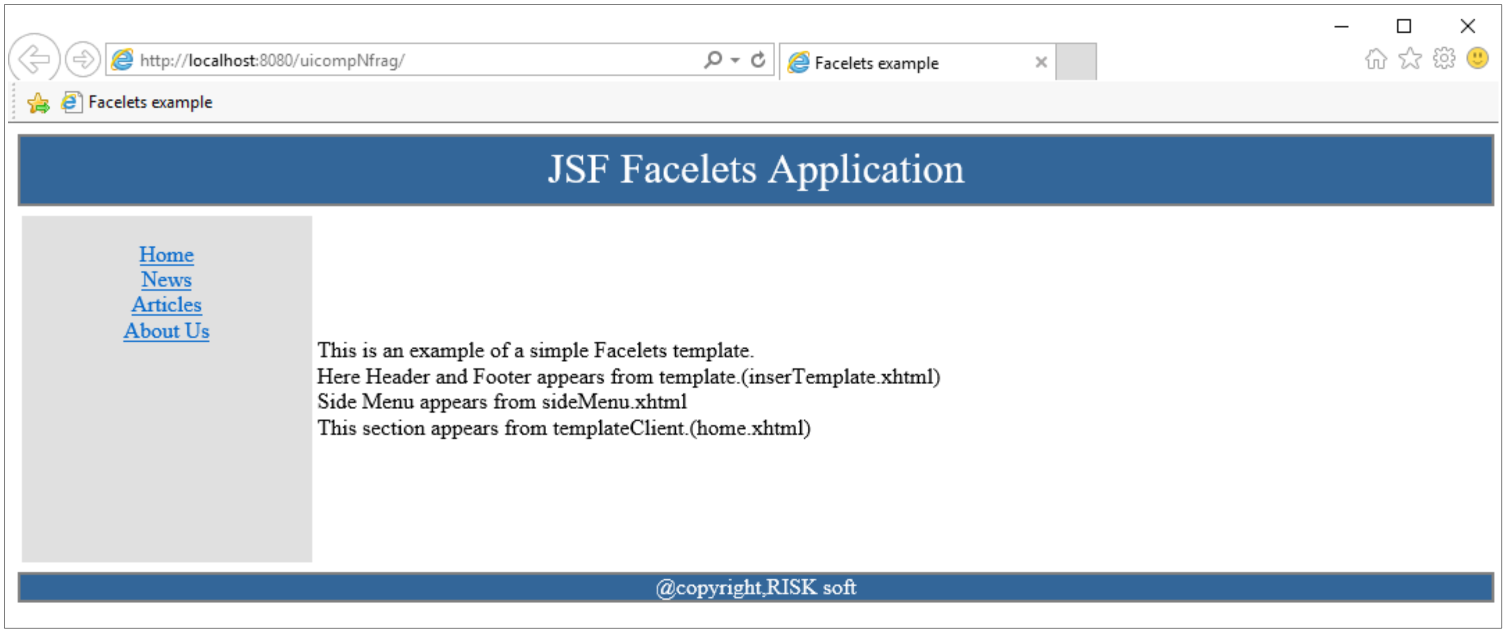
```
<UIViewRoot id="j_id1" inView="true" locale="en_US" renderKitId="HTML_BASIC" rendered="true" transient="false" viewId="/home.xhtml">

    <html xmlns="http://www.w3.org/1999/xhtml"> <head> <title>Facelets example</title> </head> <body> <div style="border-bottom: grey 2px solid; border-left: grey
    2px solid; border-right: grey 2px solid; border-top: grey 2px solid; height: 100%; margin: 4px auto; text-align: center; width: 100%;">

    <html xmlns="http://www.w3.org/1999/xhtml"> <body> <div style="width:100%;line-height:48px;background-color:#336699;color:white; font-size:30px"> JSF Facelets
    Application</div> </body> </html>

    </div> <table width="100%"> <tr> <td width="20%"> <div style="height: 250px; width: 100%; background-color: #e0e0e0; text-align: center;"> <br/>

    <html xmlns="http://www.w3.org/1999/xhtml"> This text will not be rendered.<br/> <!--<ui:component id="comp">-->

    <a href="#{menu.url}">#{menu.label}</a><br/>

    <a href="#{menu.url}">#{menu.label}</a><br/>

    <a href="#{menu.url}">#{menu.label}</a><br/>

    <a href="#{menu.url}">#{menu.label}</a><br/>

    <!--</ui:component>--> </html>

    </div> </td> <td width="85%">

    This is an example of a simple Facelets template.<br/> Here Header and Footer appears from template.(inserTemplate.xhtml)<br/> Side Menu appears from
    sideMenu.xhtml<br/> This section appears from templateClient.(home.xhtml)
```

Note : Above we can see as <ui:component> tag is commented out they content of sideMenu.xhtml has appeared with other contents and not as separate component.

2. With <ui:component> tag

```
This text will not be rendered.<br/>
<ui:component id="comp">
      <c:forEach var="menu" items="#{menuBackingBean.menus}">
           <a href="#{menu.url}">#{menu.label}</a><br/>
      </c:forEach>
</ui:component>
```

home.xhtml



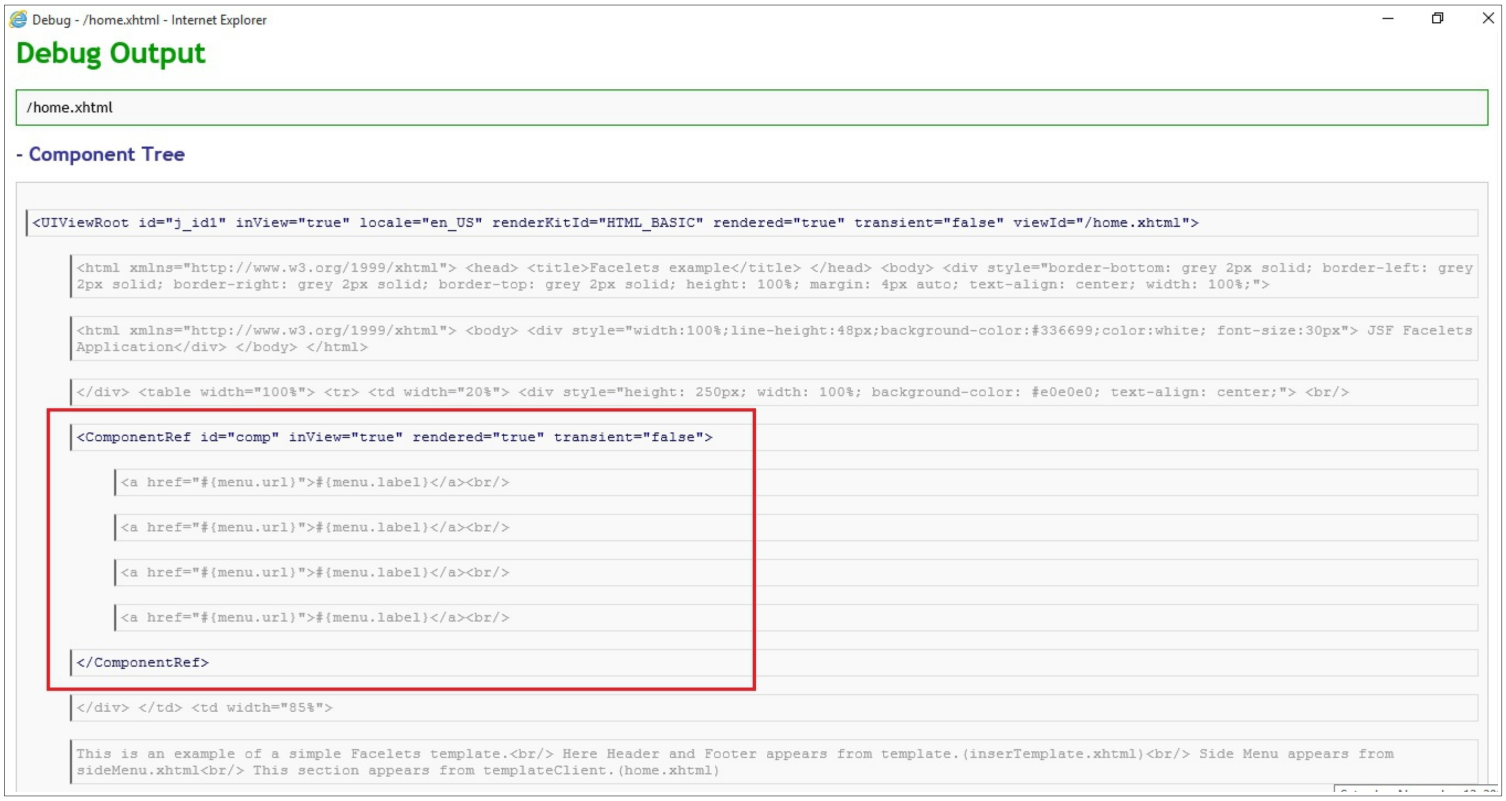
JSF Facelets Application

Home
News
Articles
About Us

This is an example of a simple Facelets template.
Here Header and Footer appears from template.(inserTemplate.xhtml)
Side Menu appears from sideMenu.xhtml
This section appears from templateClient.(home.xhtml)

@copyright,RISK soft

Debug



Debug Output

/home.xhtml

- Component Tree

```
<UIViewRoot id="j_id1" inView="true" locale="en_US" renderKitId="HTML_BASIC" rendered="true" transient="false" viewId="/home.xhtml">

    <html xmlns="http://www.w3.org/1999/xhtml"> <head> <title>Facelets example</title> </head> <body> <div style="border-bottom: grey 2px solid; border-left: grey
    2px solid; border-right: grey 2px solid; border-top: grey 2px solid; height: 100%; margin: 4px auto; text-align: center; width: 100%;">

    <html xmlns="http://www.w3.org/1999/xhtml"> <body> <div style="width:100%;line-height:48px;background-color:#336699;color:white; font-size:30px"> JSF Facelets
    Application</div> </body> </html>

    </div> <table width="100%"> <tr> <td width="20%"> <div style="height: 250px; width: 100%; background-color: #e0e0e0; text-align: center;"> <br/>

    <ComponentRef id="comp" inView="true" rendered="true" transient="false">

        <a href="#{menu.url}">#{menu.label}</a><br/>

        <a href="#{menu.url}">#{menu.label}</a><br/>

        <a href="#{menu.url}">#{menu.label}</a><br/>

        <a href="#{menu.url}">#{menu.label}</a><br/>

    </ComponentRef>

    </div> </td> <td width="85%">

    This is an example of a simple Facelets template.<br/> Here Header and Footer appears from template.(inserTemplate.xhtml)<br/> Side Menu appears from
    sideMenu.xhtml<br/> This section appears from templateClient.(home.xhtml)
```

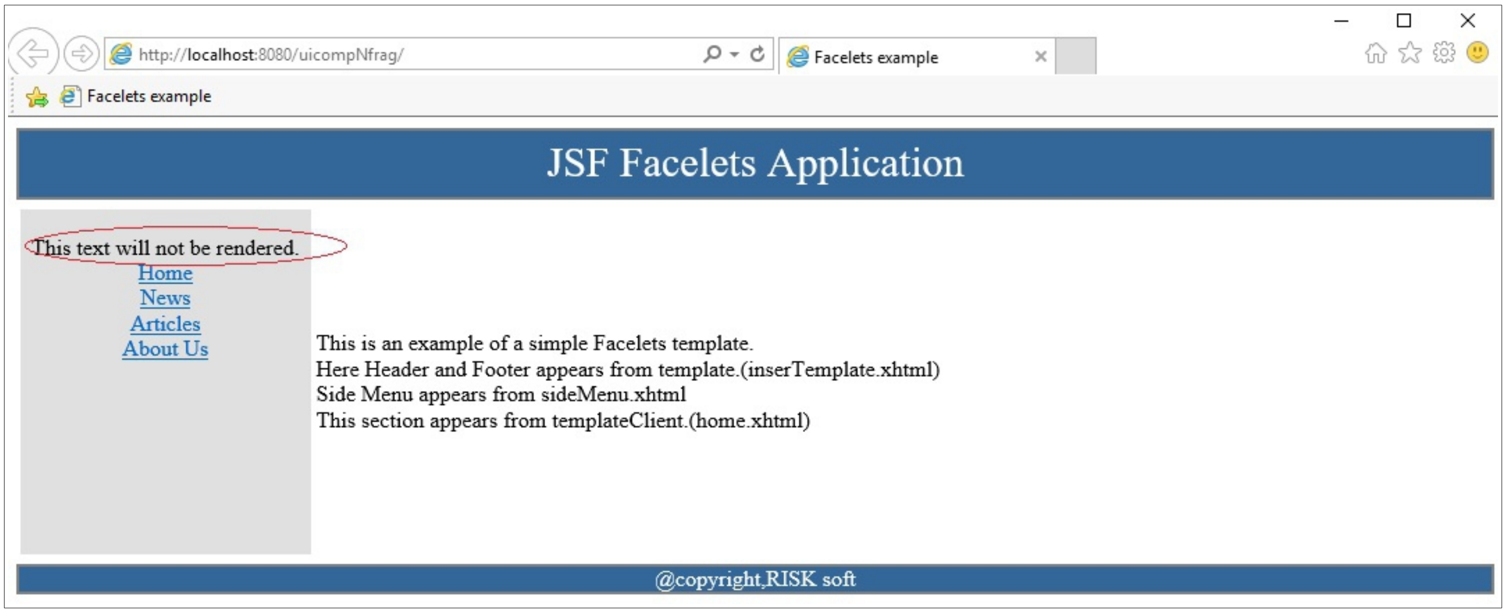
Note: Here we can note that <ui:component> is seen as a separate component in debug window.

3. With <ui:fragement>

Lets replace <ui:component> in sideMenu.xhtml with <ui:fragement> tag as shown below

```
<ui:fragement id="comp">
      <c:forEach var="menu" items="#{menuBackingBean.menus}">
           <a href="#{menu.url}">#{menu.label}</a><br/>
      </c:forEach>
</ui:fragement>
```

home.xhtml(same as without component)



JSF Facelets Application

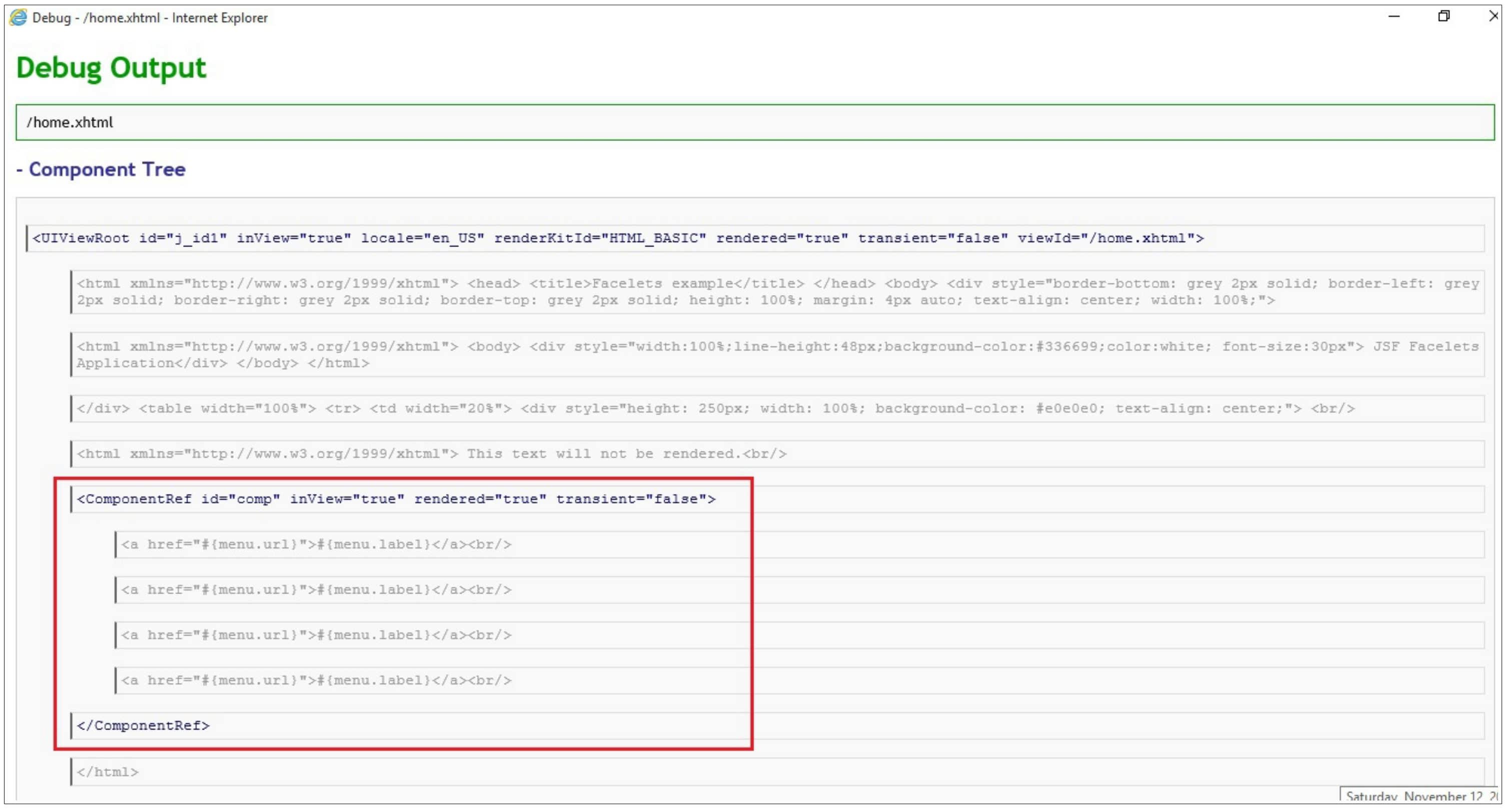
This text will not be rendered.
Home
News
Articles
About Us

This is an example of a simple Facelets template.
Here Header and Footer appears from template.(inserTemplate.xhtml)
Side Menu appears from sideMenu.xhtml
This section appears from templateClient.(home.xhtml)

@copyright,RISK soft

Note: Here contents outside <ui:fragement> is also got displayed.

Debug(Same as with Component)



Debug Output

/home.xhtml

- Component Tree

```
<UIViewRoot id="j_id1" inView="true" locale="en_US" renderKitId="HTML_BASIC" rendered="true" transient="false" viewId="/home.xhtml">

    <html xmlns="http://www.w3.org/1999/xhtml"> <head> <title>Facelets example</title> </head> <body> <div style="border-bottom: grey 2px solid; border-left: grey 2px solid; border-right: grey 2px solid; border-top: grey 2px solid; height: 100%; margin: 4px auto; text-align: center; width: 100%;">

    <html xmlns="http://www.w3.org/1999/xhtml"> <body> <div style="width:100%;line-height:48px;background-color:#336699;color:white; font-size:30px"> JSF Facelets Application</div> </body> </html>

    </div> <table width="100%"> <tr> <td width="20%"> <div style="height: 250px; width: 100%; background-color: #e0e0e0; text-align: center;"> <br/>

    <html xmlns="http://www.w3.org/1999/xhtml"> This text will not be rendered.<br/>

    <ComponentRef id="comp" inView="true" rendered="true" transient="false">

        <a href="#{menu.url}">#{menu.label}</a><br/>

        <a href="#{menu.url}">#{menu.label}</a><br/>

        <a href="#{menu.url}">#{menu.label}</a><br/>

        <a href="#{menu.url}">#{menu.label}</a><br/>

    </ComponentRef>

    </html>
```

Saturday, November 12, 2

7. **<ui:remove> tag**

➔ <ui:remove> removes content from a page i.e. dont render content.

➔ XML comment <!-- --> will not be useful as if it contains a 'value expression' then that will be rendered.

Ex

```
<!-- <h:commandButton id="loginButton"
value="#{msgs.loginButtonText}"
action="planetarium"/> -->
```

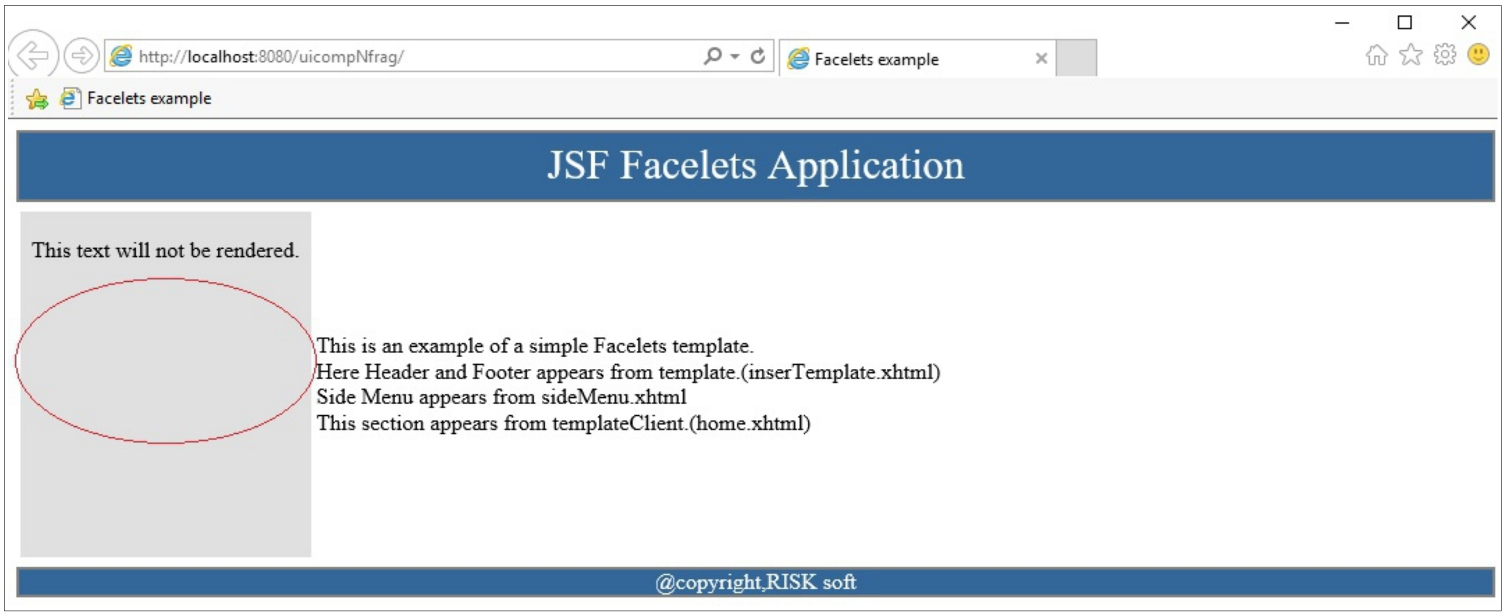
will be rendered as

```
<!-- <h:commandButton id="loginButton"
value="Log In"
action="planetarium"/> -->
```

and if loginButtonText method will throw an exception then it will be a problem. Hence <ui:remove> is the best solution.

➔ If in above example we use <ui:remove> inside sideMenu.xhtml as shown below

```
This text will not be rendered.<br/>
<ui:remove>
      <c:forEach var="menu" items="#{menuBackingBean.menus}">
           <a href="#{menu.url}">#{menu.label}</a><br/>
      </c:forEach>
</ui:remove>
```



JSF Facelets Application

This text will not be rendered.

This is an example of a simple Facelets template.
Here Header and Footer appears from template.(inserTemplate.xhtml)
Side Menu appears from sideMenu.xhtml
This section appears from templateClient.(home.xhtml)

@copyright,RISK soft

8. **Handling Whitespace**

➔ Be default in JSF white spaces are trimmed around components. For example

```
<h:outputText value="#{msgs.name}"/>
<h:inputText value="#{user.name}"/>
```

Here above two will be separated by white space. The facelets wont turn that white space into the text component.

➔ But same is not true for two links in a row

```
<h:commandLink value="Previous" .../> <h:commandLink value="Next" .../>
```

This will turn into PreviousNext. Hence to provide white space one use value expression with a space #{' '}.