

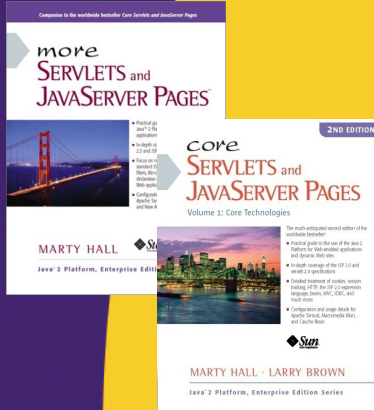


Page Templating with Facelets

Originals of Slides and Source Code for Examples:
<http://www.coreservlets.com/JSF-Tutorial/>

This somewhat old tutorial covers JSF 1, and is left online for those maintaining existing projects. All **new** projects should use JSF 2, which is both simpler and more powerful. See <http://www.coreservlets.com/JSF-Tutorial/jsf2/>.

Customized Java EE Training: <http://courses.coreservlets.com/>
Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live training on JSF 1 or 2, please see courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
 - JSF 2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 6 or 7 programming, custom mix of topics
 - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, etc.) or survey several
 - Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, GWT, Hadoop, SOAP-based and RESTful Web Services
- Contact hall@coreservlets.com for details**

Topics in This Section

- **Facelets motivation**
- **Installation and setup**
- **Basic mechanism**
 - Template file
 - Client file
- **Templating with includes**
- **Handling relative URLs**

5

Problems with JSP when Used with JSF

- **Inadequacies of jsp:include**
 - No real templates or named sections (ala Struts Tiles)
 - Can't easily pass data to included pages
 - jsp:param does not work for JSF data
- **You can't mix regular HTML inside f: tags**
 - Often need clumsy f:verbatim (e.g., with data tables)
- **Pages don't use XML syntax**
 - So less IDE help for JSP page development
 - No HTML syntax checking
- **Need h:outputText**
 - You can't do #{blah} or \${blah} directly
 - JSP (pre 2.1) needs <h:outputText value="#{blah}"/>
- **Verbose**
 - Every page needs two @taglib lines and <f:view>...</f:view>

6

Advantages of Facelets

- **Real templating**
 - Template file gives main layout and defines sections
 - Client file uses template and replaces sections
 - Can pass JSF data to included pages
- **Can use xhtml without special syntax**
- **Pages developed in xhtml**
 - XML syntax help in page development
 - xhtml syntax checking for result
- **JSP 2.1 EL evaluator included**
 - Can use `#{blah}` or `${blah}` anywhere in page
- **Succinct**
 - `@taglib` and `f:view` not required

7

Installation and Setup

- **Need 3 JAR files in WEB-INF/lib**
 - jsf-facelets.jar, el-api.jar, el-ri.jar
 - Download from <http://facelets.dev.java.net/>
 - Or grab from the sample app in this tutorial
- **Need context param(s) in web.xml**
 - Required: `javax.faces.DEFAULT_SUFFIX`
 - Optional but helpful
 - `facelets.REFRESH_PERIOD`
 - `facelets.DEVELOPMENT`
- **Need view-handler declaration in faces-config.xml**
 - `com.sun.facelets.FaceletViewHandler`
- **Note**
 - Facelet app from this tutorial has all necessary pieces already
 - Download from <http://www.coreservlets.com/JSF-Tutorial/> and use as starting point for your facelets apps

8

web.xml Settings

```
<web-app ...>
  <context-param>
    <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
    <param-value>.xhtml</param-value>
  </context-param>

  <context-param>
    <param-name>facelets.REFRESH_PERIOD</param-name>
    <param-value>2</param-value>
  </context-param>

  <context-param>
    <param-name>facelets.DEVELOPMENT</param-name>
    <param-value>true</param-value>
  </context-param>
  ...
</web-app>
```

Extension of facelets files.
E.g., foo.xhtml will be accessed as foo.faces.

How often to check for changes
after initial deployment.

Use debug/development output.

9

faces-config.xml Setting

```
<faces-config>
  <application>
    <view-handler>
      com.sun.facelets.FaceletViewHandler
    </view-handler>
  </application>

  ...

</faces-config>
```

10

Basic Templating Mechanism

- **Define template file**

- Common data for all pages goes directly in template file
- Sections that will be changed marked with ui:insert

```
...<body>
<h2><ui:insert name="title">Default Title</ui:insert></h2>
Content shared by all clients
<ui:insert name="body">Default Body</ui:insert>
</body>...
```

- **Define client file(s)**

- Specify template file with ui:composition
- Give content for the sections with ui:define

```
<ui:composition template="/.../template-file.xhtml">
  <ui:define name="title">Title text</ui:define>
  <ui:define name="body">
    Content to go in "body" section of template
  </ui:define>
</ui:composition>
```

11

Template File: Details

- **Use xhtml**

- Minimum: schemas for basic xhtml and facelets ui: prefix

```
<!DOCTYPE ... (standard xhtml) >
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  ...
</html>
```
- If you use f: tags or h: tags, list those schema also

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html">
  ...
</html>
```

12

Template File: Details

- **Insert shared content literally**
 - Any HTML or JSF data that all clients will have should be placed directly in template file
- **Mark replaceable sections with ui:insert**
 - Give a name to the section (client will refer to name)
 - Put default text between <ui:insert...> and </ui:insert>
- **Can output values without h:outputText**
 - Either #{blah} or \${blah} are legal
 - My convention is to use #{blah} within JSF tag attributes and to use \${blah} outside JSF tags
- **Put template files under WEB-INF**
 - So that it is not possible for users to access them directly

13

Template File: Example (WEB-INF/facelets/sample-template.xhtml)

```
<!DOCTYPE ...>
<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=UTF-8" />
    <title>
      <ui:insert name="title">Default Title</ui:insert>
    </title>
    <link rel="stylesheet"
          href="./css/styles.css"
          type="text/css"/>
  </head>
  <body>
    <table border="5" align="center">
      <tr><th class="title">
        <ui:insert name="title">Default Title</ui:insert>
      </th></tr>
    </table>
    <h2>A random number: ${numberBean.randomNumber}</h2>
    <ui:insert name="body">Default Body</ui:insert>
  </body></html>
```

14

Bean Code

```
package coreservlets;

public class NumberBean {
    public double getRandomNumber() {
        return (Math.random());
    }
}
```

15

faces-config.xml

```
...
<faces-config>
    <application>
        <view-handler>
            com.sun.facelets.FaceletViewHandler
        </view-handler>
    </application>

    <managed-bean>
        <managed-bean-name>numberBean</managed-bean-name>
        <managed-bean-class>
            coreservlets.NumberBean
        </managed-bean-class>
        <managed-bean-scope>request</managed-bean-scope>
    </managed-bean>
</faces-config>
```

16

Client File: Details

- **Use xhtml with appropriate schema references**
 - Needs to be legal xhtml file, even though content for output will come mostly from template
 - Text outside of ui:composition ignored for output
- **Use ui:composition to refer to template**
 - `<ui:composition template="/WEB-INF/...template.xhtml">`
- **Use ui:define to supply content for sections**
 - `<ui:define name="section-name-from-template-file">`
Content to be inserted into section.
xhtml tags, JSF tags, facelet tags, and EL expressions
`</ui:define>`
- **File does not go in WEB-INF**
 - Needs to be accessible to user
 - If file is blah.xhtml, it will be accessed as blah.faces

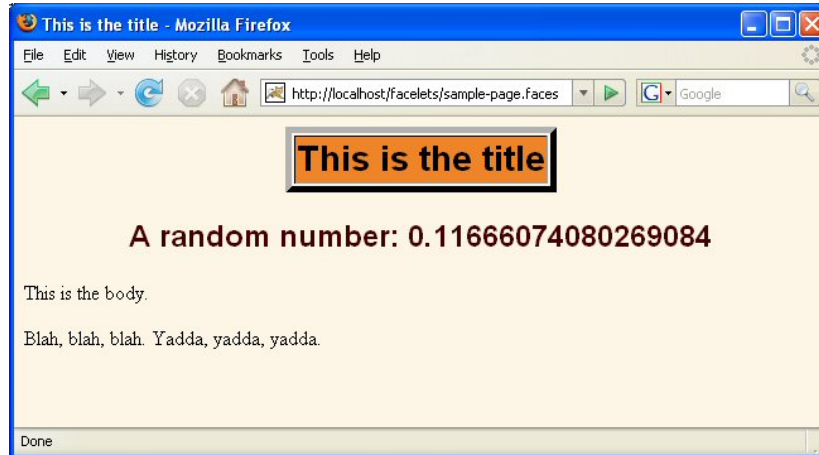
17

Client File: Example (/sample-page.xhtml)

```
<!DOCTYPE ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<body>
<ui:composition
  template="/WEB-INF/facelets/sample-template.xhtml">
  <ui:define name="title">
    This is the title
  </ui:define>
  <ui:define name="body">
    This is the body.
    <p/>
    Blah, blah, blah. Yadda, yadda, yadda.
  </ui:define>
</ui:composition>
</body>
</html>
```

18

Client File: Result



19

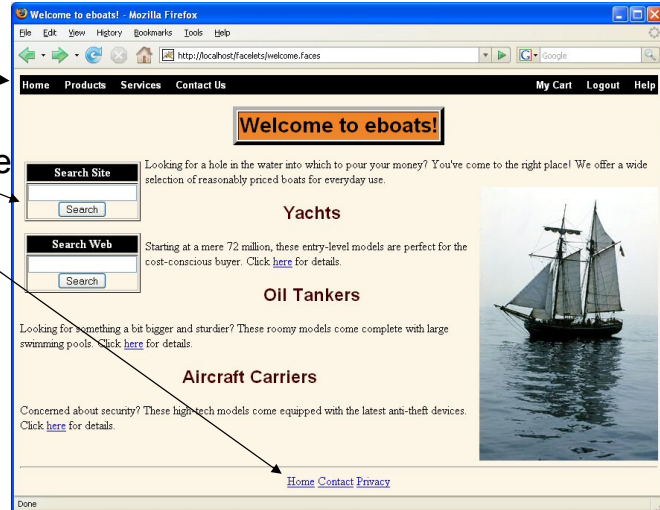
Using ui:include

- **In last example, content that was always shared went in template file**
 - What if content is sometimes shared?
- **Options**
 - Content that is common to all clients
 - Goes in template file
 - Content that is specific to a particular client
 - Goes in client file in the body of ui:define
 - Content that is shared by multiple clients, but might not be shared by all clients
 - Put in separate file
 - Unlike with jsp:include, separate file should be a complete legal xhtml file.
 - But content outside of ui:composition is ignored on inclusion
 - Client file does ui:include in the body of ui:define

20

An Online Boat Store

- **Common to all pages**
 - DOCTYPE; head, title, and body tags; style sheet; border around heading
 - Text goes directly in template
- **Common to several pages (but potentially changeable)**
 - Header
 - Footer
 - Search Menu
 - Text goes in separate files that are loaded via ui:include within ui:define
- **Unique to pages**
 - Title text
 - Body
 - Text goes directly inside ui:define



21

Template File (/WEB-INF/facelets/template.xhtml)

```
<!DOCTYPE ...>
<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=UTF-8" />
    <title><ui:insert name="title">Title</ui:insert></title>
    <link rel="stylesheet"
          href="/css/styles.css"
          type="text/css"/>
  </head>
  <body>
    <ui:insert name="header">Header</ui:insert>
  <p/>
```

22

Template File Continued (/WEB-INF/facelets/template.xhtml)

```
<table border="5" align="center">
  <tr><th class="title">
    <ui:insert name="title">Title</ui:insert>
  </th></tr>
</table>
<p/>
<table width="75" align="left" cellspacing="5">
<tr><td><ui:insert name="menu">Menu</ui:insert></td></tr>
</table>
<p/>
<ui:insert name="body">Body</ui:insert>
<br clear="all"/>
<hr/>
<ui:insert name="footer">Footer</ui:insert>
</body>
</html>
```

23

Header File (/WEB-INF/facelets/header.xhtml)

```
<!DOCTYPE ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <body>
    <ui:composition>
      <table width="100%" class="dark">
        <tr>
          <th align="left">
            <a href="welcome.faces" class="white">Home</a>
            &#160;&#160;&#160;
            <a href="products.faces" class="white">Products</a>
            &#160;&#160;&#160;
            <a href="services.faces" class="white">Services</a>
            &#160;&#160;&#160;
            <a href="contact.faces" class="white">Contact Us</a>
          </th>
          <th align="right">
            <a href="cart.faces" class="white">My Cart</a>
            &#160;&#160;&#160;
            <a href="logout.faces" class="white">Logout</a>
            &#160;&#160;&#160;
            <a href="help.faces" class="white">Help</a>
          </th>
        </tr>
      </table>
    </ui:composition>
  </body></html>
```

Content outside of ui:composition will be ignored during inclusion. But entire file still needs to be well-formed xml (xhtml).

Facelets xml parser has a bug where becomes a regular space (not non-breaking space) during inclusion. So use numeric version instead. (Bug exists as of Facelets 1.1.11)

24

Footer File (/WEB-INF/facelets/footer.xhtml)

```
<!DOCTYPE ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<body>
<ui:composition>
<div align="center">
  <a href="index.faces">Home</a>
  <a href="contact.faces">Contact</a>
  <a href="privacy.faces">Privacy</a>
</div>
</ui:composition>
</body></html>
```

25

Search Menu File (/WEB-INF/facelets/menu.xhtml)

```
<!DOCTYPE ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<body>
<ui:composition>
<div align="center">
<table border="1">
  <tr bgcolor="black"><th>
    <font color="white">Search Site</font>
  </th></tr>
  <tr><th>
    <form action="siteSearch">
      <input type="text" name="query"/><br/>
      <input type="submit" value="Search"/>
    </form>
  </th></tr>
</table>
... (second table)
</ui:composition>
</body>
</html>
```

26

Client File 1

(/welcome.xhtml)

```
<!DOCTYPE ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<body>
<ui:composition template="/WEB-INF/facelets/template.xhtml">

<ui:define name="title">
  Welcome to eboats!
</ui:define>

<ui:define name="header">
  <ui:include src="/WEB-INF/facelets/header.xhtml"/>
</ui:define>

<ui:define name="menu">
  <ui:include src="/WEB-INF/facelets/menu.xhtml"/>
</ui:define>
```

27

Client File 1 Continued

(/welcome.xhtml)

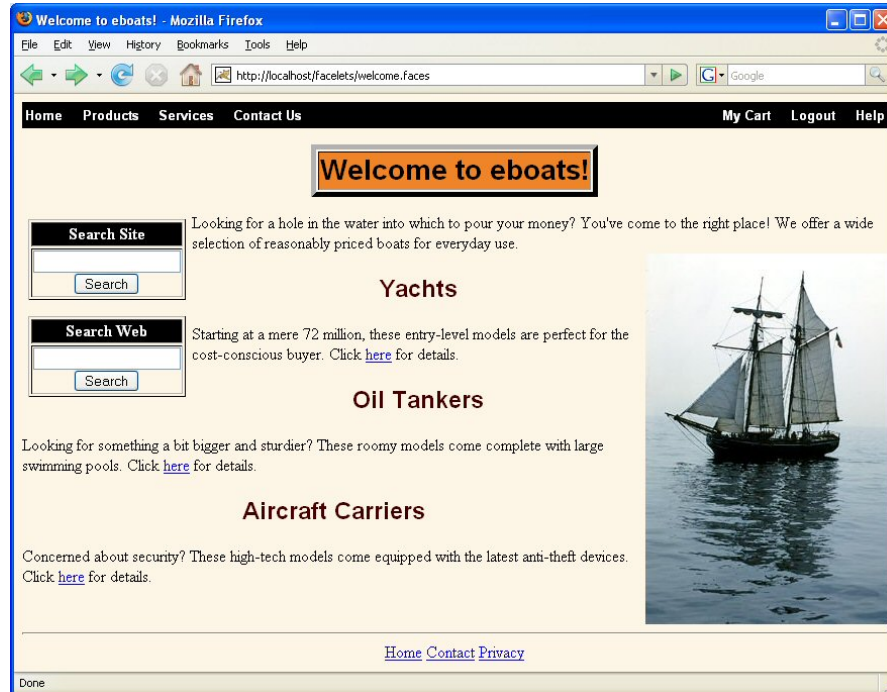
```
<ui:define name="body">
<p/>
Looking for a hole in the water into which
to pour your money?
You've come to the right place! We offer a wide selection of
reasonably priced boats for everyday use.

<h2>Yachts</h2>
... (more body content)
</ui:define>

<ui:define name="footer">
  <ui:include src="/WEB-INF/facelets/footer.xhtml"/>
</ui:define>
</ui:composition>
</body></html>
```

28

Client File 1: Result



29

Client File 2 (/yachts.xhtml)

```
<!DOCTYPE ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<body>
<ui:composition template="/WEB-INF/facelets/template.xhtml">

<ui:define name="title">
  Eboats Yachts!
</ui:define>

<ui:define name="header">
  <ui:include src="/WEB-INF/facelets/header.xhtml"/>
</ui:define>

<ui:define name="menu">
  <ui:include src="/WEB-INF/facelets/menu.xhtml"/>
</ui:define>
```

30

Client File 2 Continued (/yachts.xhtml)

```
<ui:define name="body">

<p/>
Luxurious models for the <s>wasteful</s> wealthy buyer.
<h2>Available Models</h2>
Choose a model to see a picture along with price and
availability information.
... (more body content)
</ui:define>

<ui:define name="footer">
  <ui:include src="/WEB-INF/facelets/footer.xhtml"/>
</ui:define>
</ui:composition>
</body></html>
```

31

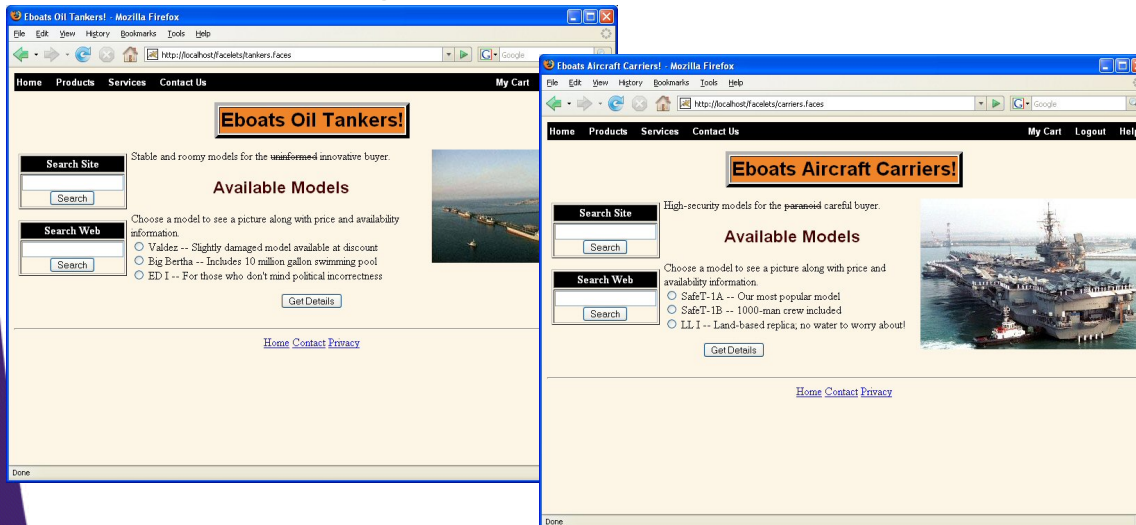
Client File 2: Result



32

Client Files 3 and 4

- Same basic format
- Same header, menu, footer
- Different body and title



33

Handling Relative URLs: Problem

- **How are simple relative URLs resolved?**
 - Relative to the URL the browser sees
 - And URLs starting with a slash would have to be edited if you rename the Web app
- **Can cause problems if templates or included pages use simple relative URLs**
 - If client pages are at different nesting levels in Web app
- **Example**
 - Template or included page
 - ``
 - Client page in dir1 folder
 - URL: `http://hostname/webappname/dir1/pic.jpg`
 - Client page in dir2 folder within dir1 folder
 - URL: `http://hostname/webappname/dir1/dir2/pic.jpg`

34

Handling Relative URLs: Solution

- **Dynamically look up Web app name**
 - And put on the front of URLs after a slash
- **Technique**
 - Use expression language
 - `${facesContext.externalContext.requestContextPath}`
- **Example**
 - ``
- **Applies to**
 - Images
 - Style sheets
 - Regular hypertext links
 - Any tag that uses URLs

35

Example: yacht-template.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">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
<title><ui:insert name="title">Default Title</ui:insert></title>
<link rel="stylesheet"
    href="${facesContext.externalContext.requestContextPath}/css/styles.css"
    type="text/css"/>
</head>
<body>
<div align="center">
<h1><ui:insert name="title">Default Title</ui:insert></h1>
<h2>Here is a Yacht</h2>

</div></body></html>
```

36

Example: yacht-client1.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">
<body>
<ui:composition template="/WEB-INF/facelets/yacht-template.xhtml">
  <ui:define name="title">
    Yacht Page 1
  </ui:define>
</ui:composition>
</body>
</html>
```

37

Result: yacht-client1.xhtml



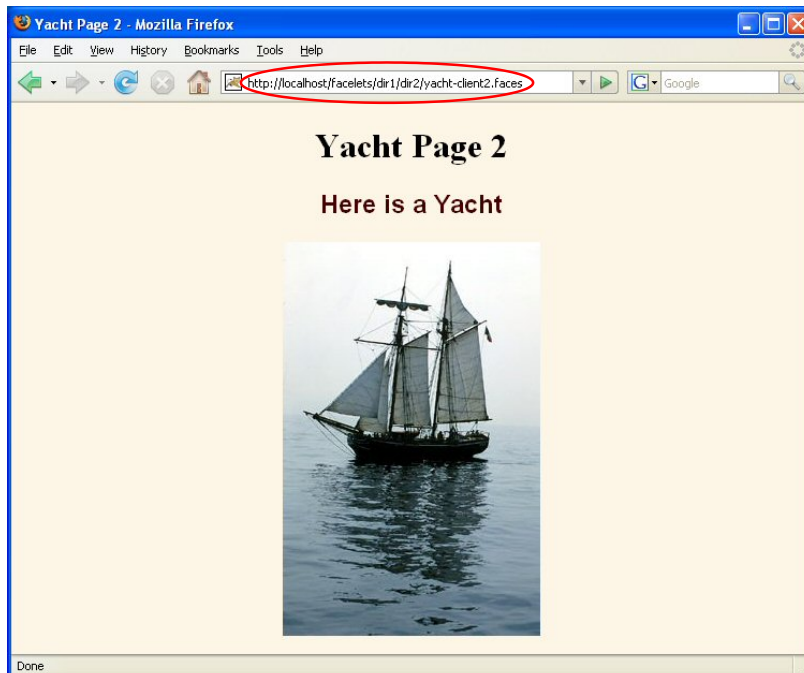
38

Example: yacht-client2.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">
<body>
<ui:composition template="/WEB-INF/facelets/yacht-template.xhtml">
  <ui:define name="title">
    Yacht Page 2
  </ui:define>
</ui:composition>
</body>
</html>
```

39

Result: yacht-client2.xhtml



40

Other Facelet Capabilities (Covered in Later Lecture)

- **Passing data to included files**
 - Use `ui:param`; can refer to JSF variables
- **Using `jsfc` in HTML-oriented tools**
 - `<form jsfc="h:form">` lets Dreamweaver or other tools render it as a form element, but JSF treats it as `h:form`
- **`ui:decorate` instead of `ui:composition`**
 - Content on the outside is not ignored
- **Advanced features**
 - Interacting with underlying components
 - Defining tag libraries
 - Defining tag files

41

Summary

- **Template file**
 - Goes under WEB-INF
 - Inserts content that will be shared by all client pages
 - Names regions with `ui:insert`
- **Client file**
 - Goes in regular Web app folder
 - Uses `ui:composition` to refer to template
 - Uses `ui:define` to supply content for regions
 - Literal text if content is unique to client
 - `ui:include` if content might be shared by multiple clients
- **Relative URLs**
 - Can be simplified with expression language
- **Use of `xhtml`**
 - All templates, clients, and included pages need to be legal XML (usually `xhtml`)
 - Refer to appropriate schema within `<html ...>` tag

42



Questions?

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.