



JSF: Laying Out Pages with (Struts) Tiles

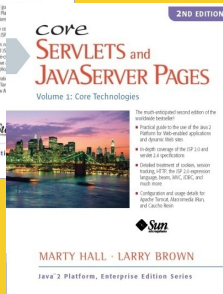
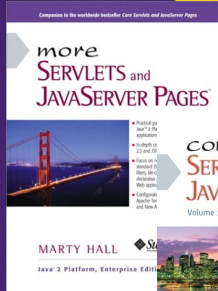
Originals of Slides and Source Code for Examples:

<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**

Agenda

- **Tiles motivations**
- **Setting up JSF (any implementation) for simple Tiles**
- **Tiles basics**
 - Sketch out desired layout
 - Make template file that represents layout
 - Create JSP pages that define layout pieces
 - Create JSP pages that populate layout
- **Setting up MyFaces for Tiles definitions**
- **Using Tiles definitions file**

5

Tiles Motivations

- **Reuse (not rewrite) repeated sections of pages**
- **Simplify the creation of similar pages**
- **Increase flexibility and ease of maintenance compared to `<jsp:include .../>`.**
- **Migrate widely used facility from Struts**

6



Tiles Basics

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.

Setting up JSF for Simple Tiles

- **Make sure struts.jar is in WEB-INF/lib**
 - It is in MyFaces download, but not jsf-blank
 - You can also take struts.jar directly from an existing Struts application
 - Soon, you can use tiles.jar instead
 - Tiles is being moved out of Struts into separate project
- **Portability**
 - Simple tiles work in all JSF implementations just by including JAR file
 - However, definitions files and special view handlers require implementation-specific setup
 - See later sub-section for use in MyFaces

Tiles Basics

- 1. Sketch out desired layout**
 - Do this on paper
- 2. Make template file that represents layout**
 - Use `tiles:insert` wherever layout piece goes
 - Use `tiles:getAsString` wherever changeable text goes
 - To prevent direct access, put template in WEB-INF
- 3. Create JSP pages that define layout pieces**
 - JSP pages containing HTML snippets
 - To prevent direct access, put JSP pages in WEB-INF
- 4. Create JSP pages that populate layout**
 - Use `tiles:insert` to refer to the layout from #1 above
 - Use `tiles:put` to specify layout pieces
 - Unless forwarded to, JSP *not* in WEB-INF

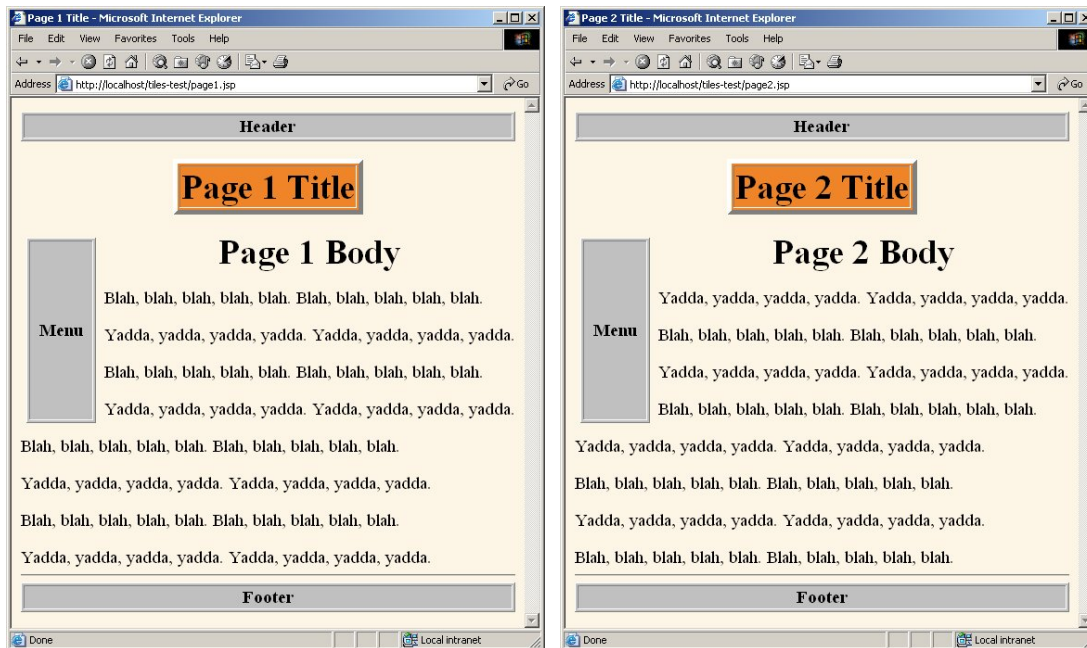
9

Step 1: Sketch Out Desired Layout

- **Many sites have groups of pages that have a consistent general layout**
 - E.g., our example pages will contain five elements
 - Header
 - Title
 - Menu
 - Body
 - Footer
- **Often only one or two parts (e.g., the body) changes from page to page**
- **Helpful to sketch layout on paper**

10

Step 1: Sketch Out Desired Layout



11

Step 2: Make Template File That Represents Layout

- Use regular HTML for static parts
- Declare the Struts tiles tag library
 - `<%@ taglib uri="http://struts.apache.org/tags-tiles" prefix="tiles" %>`
- Use `tiles:insert` to stub out sections that will be filled in by template pieces
 - `<tiles:insert attribute="header"/>`
 - The eventual JSP page will associate a real file with the header attribute
- Use `tiles:getAsString` to stub out locations that will be replaced by simple text
 - `<tiles:getAsString name="title"/>`
- Template file should never be accessed directly
 - So, put it in WEB-INF to avoid accidental access

12

Step 2: Make Template File That Represents Layout

- **WEB-INF/tiles/layout.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
      prefix="tiles" %>
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE><tiles:getAsString name="title"/></TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<tiles:insert attribute="header"/>
<P>
<TABLE BORDER=5 ALIGN="CENTER" BGCOLOR="#EF8429">
  <TR><TH><FONT SIZE="+3">
    <tiles:getAsString name="title"/>
  </FONT></TH></TR></TABLE>
<P>
<TABLE WIDTH=75 ALIGN="LEFT" CELLSPACING="5">
<TR><TD><tiles:insert attribute="menu"/></TD></TR>
</TABLE>
<tiles:insert attribute="body"/>
<BR CLEAR="ALL"><HR>
<tiles:insert attribute="footer"/>
</BODY></HTML>
```

13

Step 3: Create JSP Pages That Define Layout Pieces

- **These are regular JSP pages that define portions of an HTML page**
 - Be sure not to repeat tags such as DOCTYPE, HEAD, BODY, etc., that are defined in template file from Step 2
- **These pages can themselves be tiles**
- **Pages can use JSP and JSF tags**
 - Layout pieces can access beans that are in request scope
 - Only top-level page should have f:view
 - Any tile:insert statements that are within a JSF tag must say flush="false"
 - <tiles:insert attribute="whatever" flush="false"/>
- **Layout pieces should never be accessed directly**

14

– So, put them in WEB-INF to avoid accidental access

Step 3: Create JSP Pages That Define Layout Pieces

- **WEB-INF/tiles/header.jsp**

```
<TABLE BORDER=1 WIDTH="100%" BGCOLOR="#C0C0C0">
  <TR><TH>Header</TH></TR>
</TABLE>
```

- **WEB-INF/tiles/body1.jsp**

```
<H1 ALIGN="CENTER">Page 1 Body</H1>
Blah, blah, blah, blah, blah.
Blah, blah, blah, blah, blah.
<P>
Yadda, yadda, yadda, yadda.
Yadda, yadda, yadda, yadda.
```

- **WEB-INF/tiles/footer.jsp**

```
<TABLE BORDER=1 WIDTH="100%" BGCOLOR="#C0C0C0">
  <TR><TH>Footer</TH></TR>
```

15

Step 4: Create JSP Pages That Populate Layout

- **Declare the Struts tiles tag library**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
  prefix="tiles" %>
```

- **Use tiles:insert to refer to the layout defined in Step 2**

```
<tiles:insert page="/WEB-INF/tiles/layout.jsp">
```

...

```
</tiles:insert>
```

- **Use tiles:put to specify the layout pieces that apply to this specific page**

```
<tiles:insert ...>
```

```
<tiles:put name="title" value="Page 1 Title"/>
```

```
<tiles:put name="header"
  value="/WEB-INF/tiles/header.jsp"/>
```

```
</tiles:insert>
```

16

Step 4: Create JSP Pages That Populate Layout

- **page1.jsp (not in WEB-INF)**

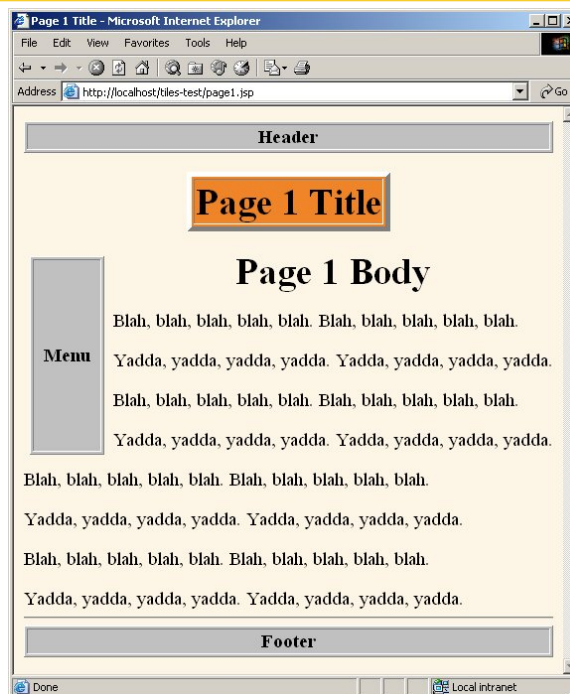
```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/layout.jsp">
  <tiles:put name="title" value="Page 1 Title"/>
  <tiles:put name="header" value="/WEB-INF/tiles/header.jsp"/>
  <tiles:put name="menu" value="/WEB-INF/tiles/menu.jsp"/>
  <tiles:put name="body" value="/WEB-INF/tiles/body1.jsp"/>
  <tiles:put name="footer" value="/WEB-INF/tiles/footer.jsp"/>
</tiles:insert>
```

- **page2.jsp (not in WEB-INF)**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/layout.jsp">
  <tiles:put name="title" value="Page 2 Title"/>
  <tiles:put name="header" value="/WEB-INF/tiles/header.jsp"/>
  <tiles:put name="menu" value="/WEB-INF/tiles/menu.jsp"/>
  <tiles:put name="body" value="/WEB-INF/tiles/body2.jsp"/>
  <tiles:put name="footer" value="/WEB-INF/tiles/footer.jsp"/>
</tiles:insert>
```

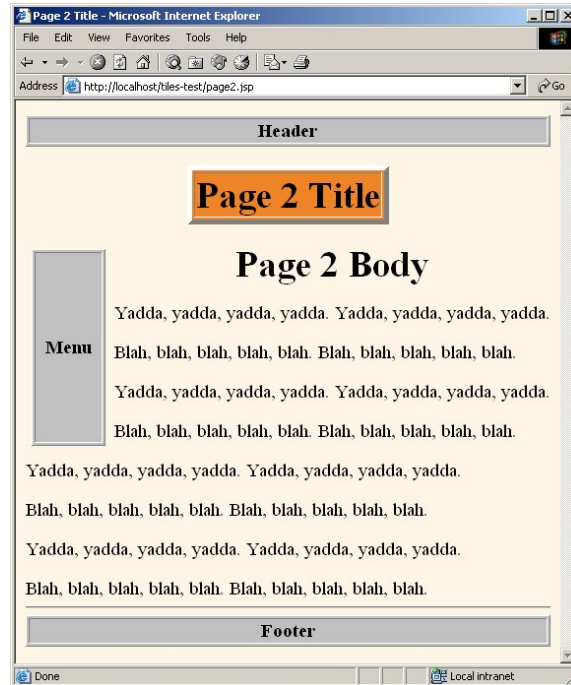
17

Results: page1.jsp



18

Results: page2.jsp



19

© 2012 Marty Hall



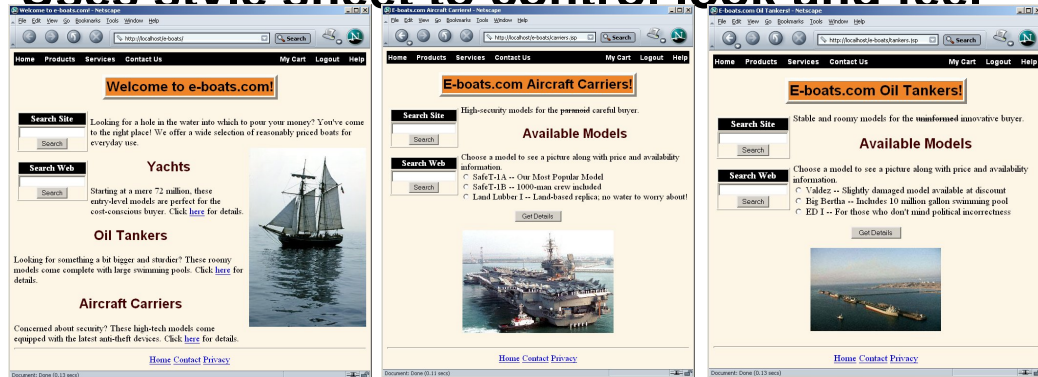
Example: e-boats

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.

The e-boats Application

- Online store to sell boats
- Uses tiles for pages that have similar layouts
 - Only title and body changes from page to page
- Uses style sheet to control look and feel



21

Step 1: Sketch Out Desired Layout

- Header
- Title
- Search menu
- Body (No outline)
- Footer



22

Step 2: Make Template File That Represents Layout

- WEB-INF/tiles/main-layout.jsp

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
      prefix="tiles" %>
<%@ taglib uri="http://struts.apache.org/tags-html"
      prefix="html" %>
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE><tiles:getAsString name="title"/>
</TITLE></HEAD>
<LINK REL="STYLESHEET"
      HREF="css/styles.css"
      TYPE="text/css">
```

23

Step 2: Make Template File That Represents Layout

- WEB-INF/tiles/main-layout.jsp (Continued)

```
<BODY>
<tiles:insert attribute="header"/>
<P>
<TABLE BORDER=5 ALIGN="CENTER">
  <TR><TH CLASS="TITLE"><tiles:getAsString name="title"/>
</TH></TR></TABLE>
<P>
<TABLE WIDTH=75 ALIGN="LEFT" CELSPACING="5">
<TR><TD><tiles:insert attribute="search-menu"/></TD></TR>
</TABLE>
<tiles:insert attribute="body"/>
<BR CLEAR="ALL">
<HR>
<tiles:insert attribute="footer"/>
</BODY></HTML>
```

24

Step 3: Create JSP Pages That Define Layout Pieces

- **WEB-INF/tiles/footer.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-html"
      prefix="html" %>
<CENTER>
<A HREF="index.html">Home</A>
<A HREF="contact.html">Contact</A>
<A HREF="privacy.html">Privacy</A>
</CENTER>
```

27

Step 3: Create JSP Pages That Define Layout Pieces

- **WEB-INF/tiles/body-index.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-html"
      prefix="html" %>
<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.
<IMG SRC="images/yacht.jpg"
      WIDTH=240 HEIGHT=367
      ALIGN="RIGHT" ALT="Base-model yacht">

<H2>Yachts</H2>
Starting at a mere 72 million, these entry-level models are
perfect for the cost-conscious buyer.
Click <A HREF="yachts.jsp">here</A> for details.

<H2>Oil Tankers</H2>
...
<H2>Aircraft Carriers</H2>
...
```

28

Step 3: Create JSP Pages That Define Layout Pieces

- WEB-INF/tiles/body-yachts.jsp
- WEB-INF/tiles/body-tankers.jsp
- WEB-INF/tiles/body-carriers.jsp

29

Step 4: Create JSP Pages That Populate Layout

- index.jsp (not in WEB-INF)

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/main-layout.jsp">
    <tiles:put name="title"
        value="Welcome to e-boats.com!"/>
    <tiles:put name="header"
        value="/WEB-INF/tiles/header.jsp"/>
    <tiles:put name="search-menu"
        value="/WEB-INF/tiles/search-menu.jsp"/>
    <tiles:put name="body"
        value="/WEB-INF/tiles/body-index.jsp"/>
    <tiles:put name="footer"
        value="/WEB-INF/tiles/footer.jsp"/>
</tiles:insert>
```

30

Result: index.jsp



31

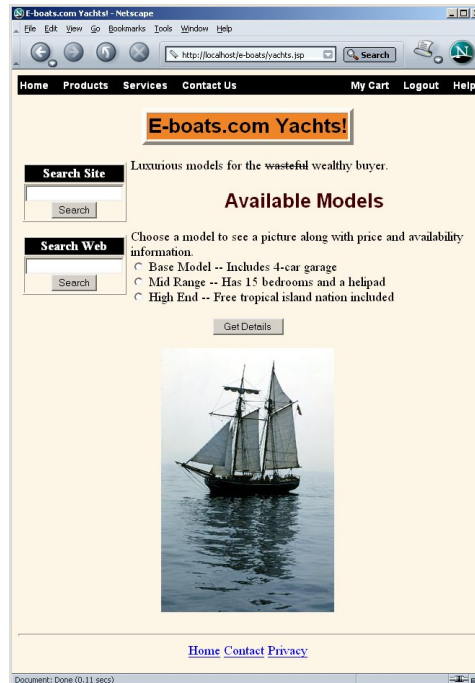
Step 4: Create JSP Pages That Populate Layout

- **yachts.jsp** (not in WEB-INF)

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/main-layout.jsp">
    <tiles:put name="title"
        value="E-boats.com Yachts!"/>
    <tiles:put name="header"
        value="/WEB-INF/tiles/header.jsp"/>
    <tiles:put name="search-menu"
        value="/WEB-INF/tiles/search-menu.jsp"/>
    <tiles:put name="body"
        value="/WEB-INF/tiles/body-yachts.jsp"/>
    <tiles:put name="footer"
        value="/WEB-INF/tiles/footer.jsp"/>
</tiles:insert>
```

32

Result: yachts.jsp



33

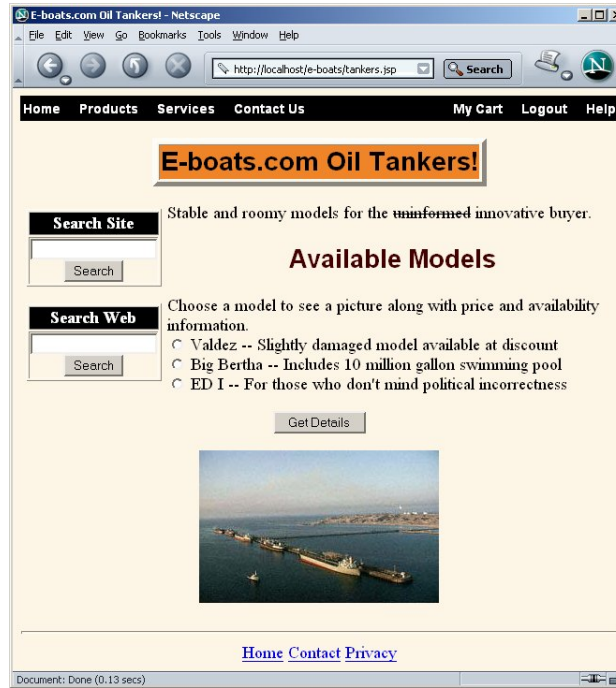
Step 4: Create JSP Pages That Populate Layout

- **tankers.jsp (not in WEB-INF)**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/main-layout.jsp">
  <tiles:put name="title"
    value="E-boats.com Oil Tankers!"/>
  <tiles:put name="header"
    value="/WEB-INF/tiles/header.jsp"/>
  <tiles:put name="search-menu"
    value="/WEB-INF/tiles/search-menu.jsp"/>
  <tiles:put name="body"
    value="/WEB-INF/tiles/body-tankers.jsp"/>
  <tiles:put name="footer"
    value="/WEB-INF/tiles/footer.jsp"/>
</tiles:insert>
```

34

Result: tankers.jsp



35

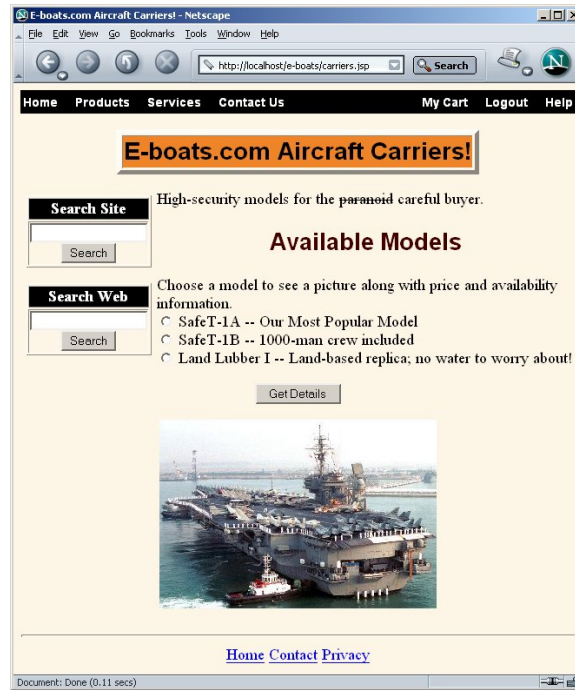
Step 4: Create JSP Pages That Populate Layout

- **carriers.jsp (not in WEB-INF)**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/main-layout.jsp">
  <tiles:put name="title"
    value="E-boats.com Aircraft Carriers!"/>
  <tiles:put name="header"
    value="/WEB-INF/tiles/header.jsp"/>
  <tiles:put name="search-menu"
    value="/WEB-INF/tiles/search-menu.jsp"/>
  <tiles:put name="body"
    value="/WEB-INF/tiles/body-carriers.jsp"/>
  <tiles:put name="footer"
    value="/WEB-INF/tiles/footer.jsp"/>
</tiles:insert>
```

36

Result: carriers.jsp



37

© 2012 Marty Hall



Using Tiles Definitions

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.

Motivation

- **Avoiding repetition**
 - In the previous examples, each final JSP page repeated the definition of all elements
 - Even though only the title and body changed, each final JSP page still had to do tiles:put on the header, search-menu, and footer
- **Centralized changes**
 - The locations of the various template pieces are scattered around multiple JSP pages
 - So, if locations change, multiple files need to be edited
- **Usage of config files**
 - JSF philosophy is to put as much as possible in config files and as little as possible in JSP/Java code

39

Setting Up MyFaces for Tiles Definitions

- **Add to faces-config.xml**

```
<application>
  <view-handler>org.apache.myfaces.tomahawk.
    application.jsp.JspTilesViewHandlerImpl
  </view-handler>
</application>
```

 - Note error in online MyFaces docs
 - Missing / in </view-handler>
- **Add to web.xml**

```
<context-param>
  <param-name>tiles-definitions</param-name>
  <param-value>WEB-INF/tiles-defs.xml</param-value>
</context-param>
```

 - Note minor error in online MyFaces docs
 - They use tiles.xml; standard Struts usage is tiles-defs.xml
- **At least one .faces page must be accessed**

40

Using Definitions: Steps

- 1. Sketch out desired layout**
 - Same as before
- 2. Make template file that represents layout**
 - Same as before
- 3. Create JSP pages that define layout pieces**
 - Same as before
- 4. Define layouts in WEB-INF/tiles-defs.xml**
- 5. Create JSP pages that use layouts**
 - Two line JSP pages
 - Imports tiles tag library
 - Uses tiles:insert with definition name

41

Using Definitions: Key Differences

- **Define layouts in WEB-INF/tiles-defs.xml**
 - Top-level layouts
 - `<definition name="..." path="/WEB-INF/.../some-layout.jsp">`
 - Derived layouts
 - `<definition name="..." extends="top-level-layout-name">`
- **In final JSP page, use definition instead of page for tiles:insert**
 - `<tiles:insert definition="layout-name"/>`
- **Advantages**
 - Derived layouts need only specify parts that differ from parent layout
 - Can make changes in a single file
 - Definitions are in config files, not JSP pages

42

Redoing e-boats with Tiles Definitions

- Same page names
- Same look



43

Step 1: Sketch Out Desired Layout

- Header
- Title
- Search menu
- Body
(No outline)
- Footer



44

Step 2: Make Template File That Represents Layout

- **WEB-INF/tiles/main-layout.jsp**

- No change from previous example

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
      prefix="tiles" %>
<%@ taglib uri="http://struts.apache.org/tags-html"
      prefix="html" %>
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE><tiles:getAsString name="title"/></TITLE></HEAD>
<LINK REL="STYLESHEET"
      HREF="css/styles.css"
      TYPE="text/css">
<BODY>
<tiles:insert attribute="header"/>
<P>
<TABLE BORDER=5 ALIGN="CENTER">
  <TR><TH CLASS="TITLE"><tiles:getAsString name="title"/>
</TH></TR></TABLE>
```

45

Step 3: Create JSP Pages That Define Layout Pieces

- **No changes from previous example**

- WEB-INF/tiles/header.jsp
- WEB-INF/tiles/search-menu.jsp
- WEB-INF/tiles/footer.jsp
- WEB-INF/tiles/body-index.jsp
- WEB-INF/tiles/body-yachts.jsp
- WEB-INF/tiles/body-tankers.jsp
- WEB-INF/tiles/body-carriers.jsp

46

Step 4: Define Layouts in WEB-INF/tiles-defs.xml

- **General format**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE tiles-definitions PUBLIC ...>

<tiles-definitions>
  ...
</tiles-definitions>
```

47

Step 4: Define Layouts in WEB-INF/tiles-defs.xml

- **Top-Level Definition**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE tiles-definitions PUBLIC ...>

<tiles-definitions>
  <definition name=".mainLayout"
    path="/WEB-INF/tiles/main-layout.jsp">
    <put name="title" value=""/>
    <put name="header"
      value="/WEB-INF/tiles/header.jsp"/>
    <put name="search-menu"
      value="/WEB-INF/tiles/search-menu.jsp"/>
    <put name="body" value=""/>
    <put name="footer"
      value="/WEB-INF/tiles/footer.jsp"/>
  </definition>
  ...
</tiles-definitions>
```

48

Step 4: Define Layouts in WEB-INF/tiles-defs.xml

- **Derived Definitions**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE tiles-definitions PUBLIC ...>

<tiles-definitions>
  ...
  <definition name="index.page" extends=".mainLayout">
    <put name="title" value="Welcome to e-boats.com!" />
    <put name="body"
        value="/WEB-INF/tiles/body-index.jsp" />
  </definition>
  <definition name="yachts.page" extends=".mainLayout">
    <put name="title" value="E-boats.com Yachts!" />
    <put name="body"
        value="/WEB-INF/tiles/body-yachts.jsp" />
  </definition>
  ...
</tiles-definitions>
```

49

Step 5: Create JSP Pages That Use Layouts

- **index.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert definition="index.page" />
```

- **yachts.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert definition="yachts.page" />
```

- **tankers.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
<tiles:insert definition="tankers.page" />
```

- **carriers.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
    prefix="tiles" %>
```

50

Result: index.jsp



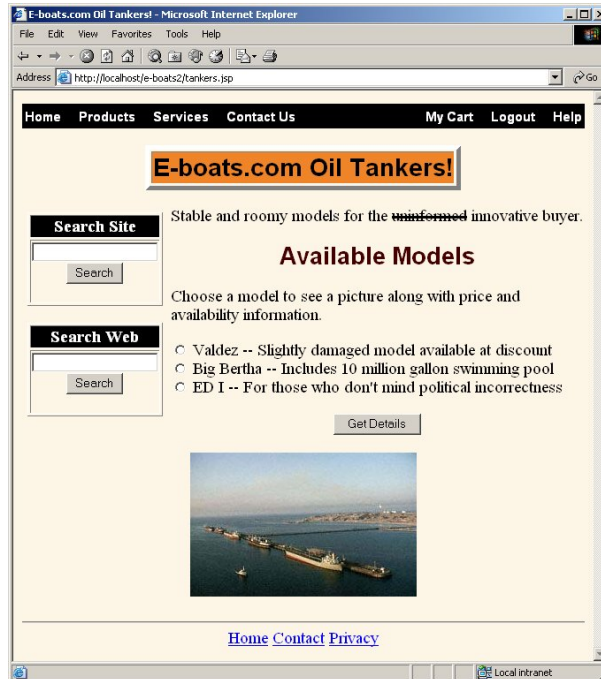
51

Result: yachts.jsp



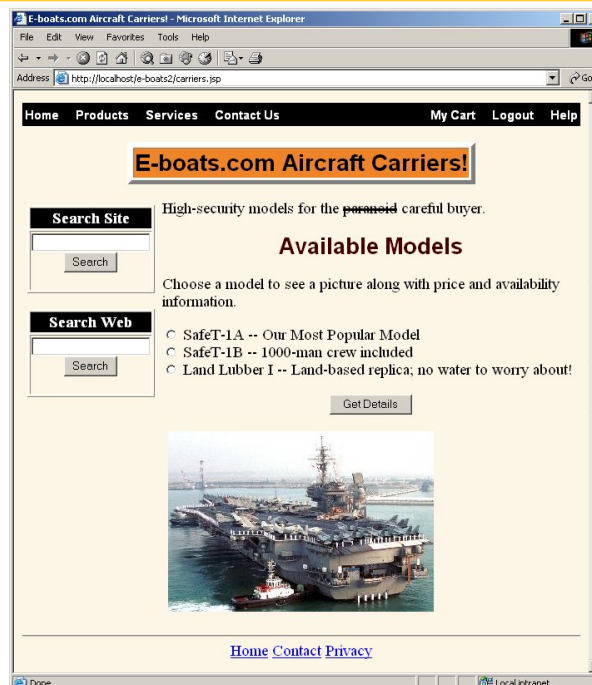
52

Result: tankers.jsp



53

Result: carriers.jsp



54

Using Definition Names as JSF View IDs (Specific to MyFaces)

- **Definitions must be called *blah.tiles***
- **You can list definition names instead of JSP page locations in navigation rules**
 - Change:

```
<navigation-case>  
  <from-outcome>...</from-outcome>  
  <to-view-id>/carriers.jsp</to-view-id>  
</navigation-case>
```
 - To:

```
<navigation-case>  
  <from-outcome>...</from-outcome>  
  <to-view-id>carriers.tiles</to-view-id>  
</navigation-case>
```
- **If JSP pages are only used as results pages, you can omit the 2-line JSP pages**

55

Summary

- **Setup**
 - Basics: put struts.jar in WEB-INF/lib (portable)
 - Definitions: edit faces-config.xml and web.xml (MyFaces only)
- **Basic Usage**
 - Sketch out desired layout
 - Make template file that represents layout
 - Create JSP pages that define layout pieces
 - Create JSP pages that populate layout
- **Usage with Definition File**
 - Sketch out desired layout
 - Make template file that represents layout
 - Create JSP pages that define layout pieces
 - Define layouts in WEB-INF/tiles-defs.xml
 - Create JSP pages that use layouts
 - Can omit and refer to definition names if named .tiles

56



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.