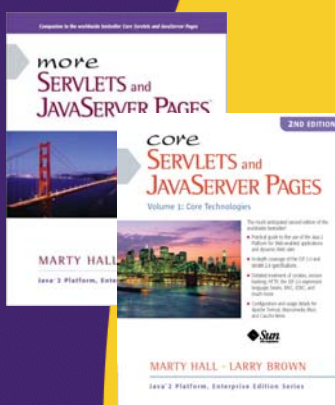# Apache Struts:
Laying Out Pages
with Tiles

Core Servlets & JSP book: www.coreservlets.com
More Servlets & JSP book: www.moreservlets.com
Servlet/JSP/Struts/JSF Training: courses.coreservlets.com

Slides © Marty Hall, http://www.coreservlets.com, books © Sun Microsystems Press

---

For live Struts training, please see JSP/servlet/Struts/JSF training 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.

Slides © Marty Hall, http://www.coreservlets.com, books © Sun Microsystems Press

Agenda

- **Tiles motivations**
- **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
- **Handling relative URLs in Tiles**
  - Use html:rewrite
- **Using Tiles definitions file**

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

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 …/>.**

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

Tiles Basics

Prerequisites

- **Two entries needed in struts-config.xml**

  ```
  <controller processorClass=
          "org.apache.struts.tiles.TilesRequestProcessor"/>
  ...
   <plug-in className="org.apache.struts.tiles.TilesPlugin" >
     <!-- Path to XML definition file -->
     <set-property property="definitions-config"
                     value="/WEB-INF/tiles-defs.xml" />
     <!-- Set Module-awareness to true -->
     <set-property property="moduleAware" value="true" />
   </plug-in>
  ```

- **These entries <u>already</u> exist in the struts-config.xml bundled with struts-blank**
  - No changes or additions required

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

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**

Step 1: Sketch Out Desired Layout
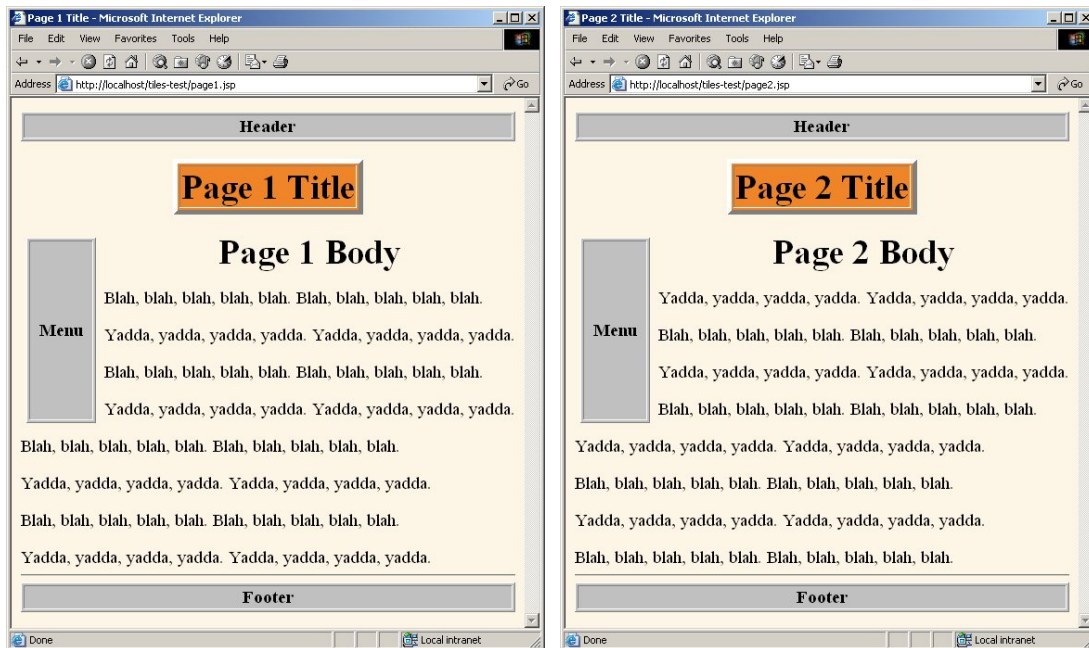
Apache Struts: Laying Out Pages with Tiles

Step 2: Make Template File That Represents Layout

- **Use regular HTML for parts that never change**
- **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

Apache Struts: Laying Out Pages with Tiles

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>
```

Step 3: Create JSP Pages That Define Layout Pieces

- **These are regular JSP pages that define <u>portions</u> 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 the bean: and html: Struts tags**
  - Layout pieces can access beans that are in request scope
- **Layout pieces should never be accessed directly**
  - 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>
</TABLE>
```

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>
  ```
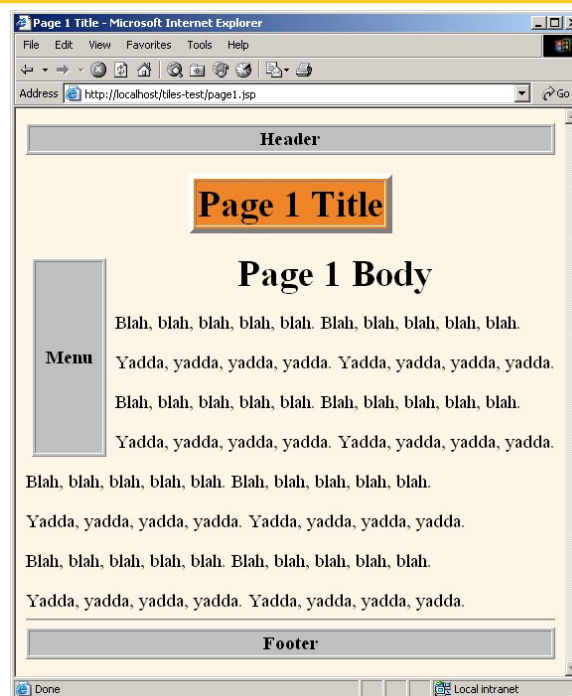
Step 4: Create JSP Pages That Populate Layout

- ## page1.jsp (<u>not</u> 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 (<u>not</u> 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>
```

Results: page1.jsp
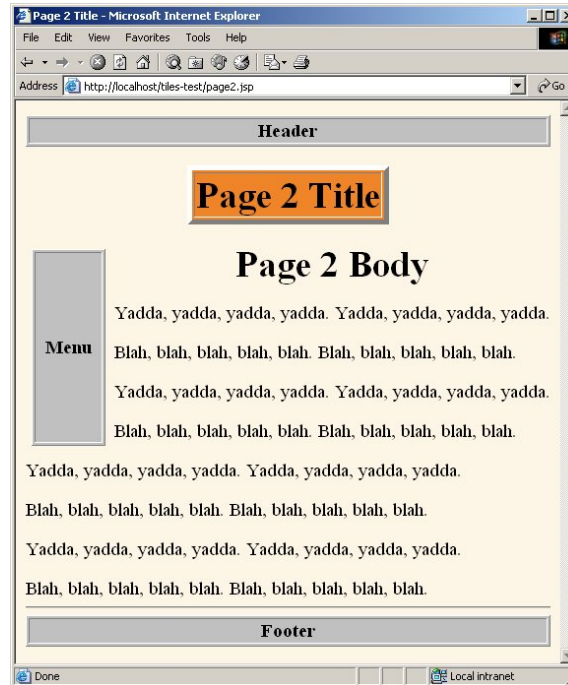
Handling Relative URLs

Relative URLs in Page Snippets

- **Many page elements are referred to with relative URLs**
  - Images
  - Style sheets
  - Simple hypertext links
- **Page elements cannot be referred to with URLs that are relative to location of layout pieces**
  - Pieces are in WEB-INF, so not directly accessible
  - Even if not in WEB-INF, client does not see the URL
- **Could use URL beginning with a slash**
  - But then you would have to list the Web app prefix
- **Could use getContextPath**
  - But then you need scriptlet

Using html:rewrite

- **Lets you use URLs beginning with slashes**
  - You list path relative to Web application home
  - System adds Web application prefix (i.e., context path) to the front
- **You must import the Struts html: library first**
  - Same tag library as used in the section on form handling
- **Examples**
  - From layout page:
  ```
  <LINK REL="STYLESHEET"
        HREF="<html:rewrite page='/css/styles.css'/>"
        TYPE="text/css">
  ```
  - From snippet or layout page:
  ```
  <IMG SRC="<html:rewrite page='/images/pic.jpg'/>"…>
  ```

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**
- **Uses images on multiple pages**
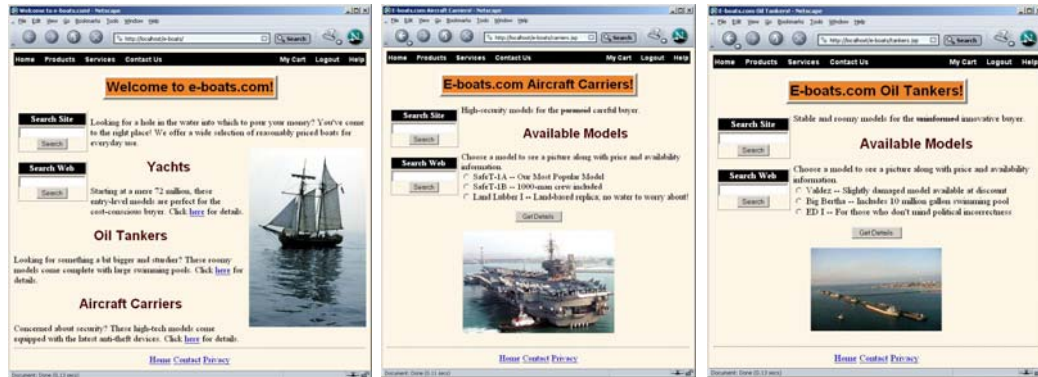
Step 1: Sketch Out Desired Layout

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

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="<html:rewrite page='/css/styles.css'/>"
      TYPE="text/css">
```

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" CELLSPACING="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>
```

Step 3: Create JSP Pages That Define Layout Pieces

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

```
<%@ taglib uri="http://struts.apache.org/tags-html"
           prefix="html" %>
<TABLE WIDTH="100%" CLASS="DARK">
  <TR>
    <TH ALIGN="LEFT">
      <A HREF="<html:rewrite page='/index.html'/>"
         CLASS="WHITE">
      Home</A>   
      <A HREF="<html:rewrite page='/products.html'/>"
         CLASS="WHITE">
      Products</A>   
      <A HREF="<html:rewrite page='/services.html'/>"
         CLASS="WHITE">
      Services</A>   
      <A HREF="<html:rewrite page='/contact.html'/>"
         CLASS="WHITE">
      Contact Us</A>
    </TH> ...
</TABLE>
```

Step 3: Create JSP Pages That Define Layout Pieces

- **WEB-INF/tiles/search-menu.jsp**

```
<%@ taglib uri="http://struts.apache.org/tags-html"
           prefix="html" %>
<CENTER>
<TABLE BORDER=1>
  <TR BGCOLOR="BLACK"><TH>
    <FONT COLOR="WHITE">Search Site</FONT>
  </TH></TR>
  <TR><TH>
    <FORM ACTION=
      "<html:rewrite page='/actions/siteSearch.do'/>">
    <INPUT TYPE="TEXT" NAME="query"><BR>
    <INPUT TYPE="SUBMIT" VALUE="Search">
    </FORM>
  </TH></TR>
</TABLE>
...
</CENTER>
```

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="<html:rewrite page='/index.html'/>">Home</A>
<A HREF="<html:rewrite page='/contact.html'/>">Contact</A>
<A HREF="<html:rewrite page='/privacy.html'/>">Privacy</A>
</CENTER>
```

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="<html:rewrite page='/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="<html:rewrite page='/yachts.jsp'/>">
here</A> for details.

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

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**

Step 4: Create JSP Pages That Populate Layout

- **index.jsp (<u>not</u> in WEB-INF)**

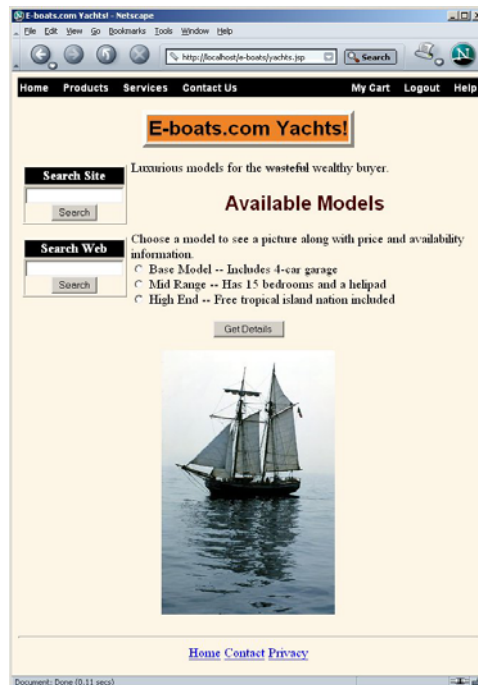```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
           prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/main-layout.jsp">
  <tiles:put name="title"
             value="Welcome to e-boats.com!"/>
  <tiles:put name="header"
             value="/WEB-INF/tiles/header.jsp"/>
  <tiles:put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

# Result: index.jsp



Apache Struts: Laying Out Pages with Tiles                    **www.coreservlets.com**

# Step 4: Create JSP Pages That Populate Layout

- **yachts.jsp (<u>not</u> in WEB-INF)**
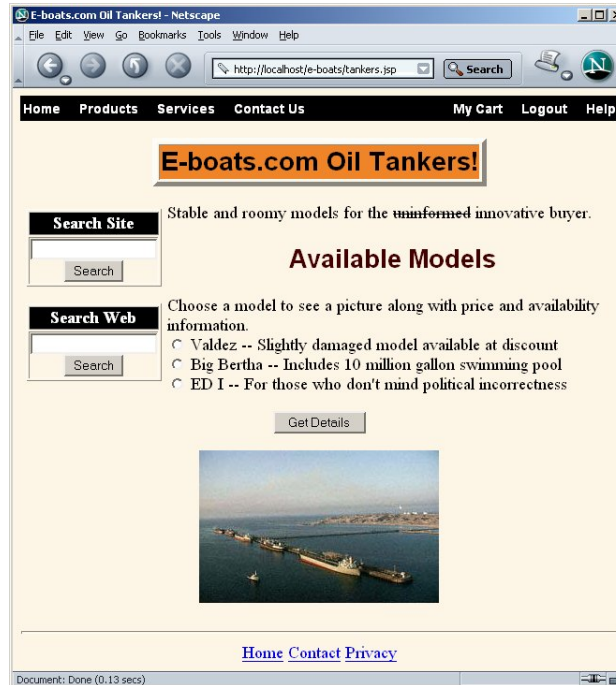
```
<%@ taglib uri="http://struts.apache.org/tags-tiles"
           prefix="tiles" %>
<tiles:insert page="/WEB-INF/tiles/main-layout.jsp">
  <tiles:put name="title"
             value="E-boats.com Yachts!"/>
  <tiles:put name="header"
             value="/WEB-INF/tiles/header.jsp"/>
  <tiles:put name="search-menu"
             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>
```

Apache Struts: Laying Out Pages with Tiles                    **www.coreservlets.com**

# Result: yachts.jsp



Apache Struts: Laying Out Pages with Tiles

---

# Step 4: Create JSP Pages That Populate Layout

- **tankers.jsp (<u>not</u> 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>
```

Apache Struts: Laying Out Pages with Tiles

Result: tankers.jsp

Apache Struts: Laying Out Pages with Tiles

www.coreservlets.com

Step 4: Create JSP Pages That Populate Layout

- **carriers.jsp (<u>not</u> in WEB-INF)**

```jsp
<%@ 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>
```

Apache Struts: Laying Out Pages with Tiles

www.coreservlets.com

Result: carriers.jsp
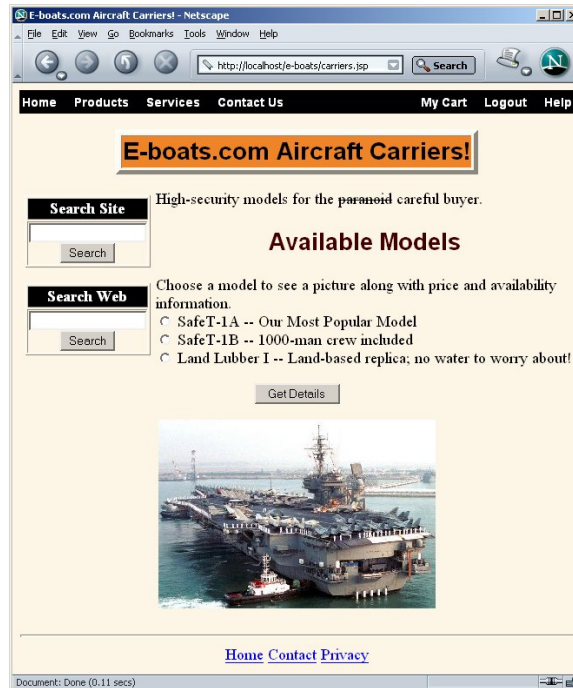
Apache Struts: Laying Out Pages with Tiles

Using Tiles Definitions

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**
  - Struts philosophy is to put as much as possible in config files and as little as possible in JSP/Java code

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

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

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

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

Redoing e-boats with Tiles Definitions

- **Same page names**
- **Same look**

Step 1: Sketch Out Desired Layout
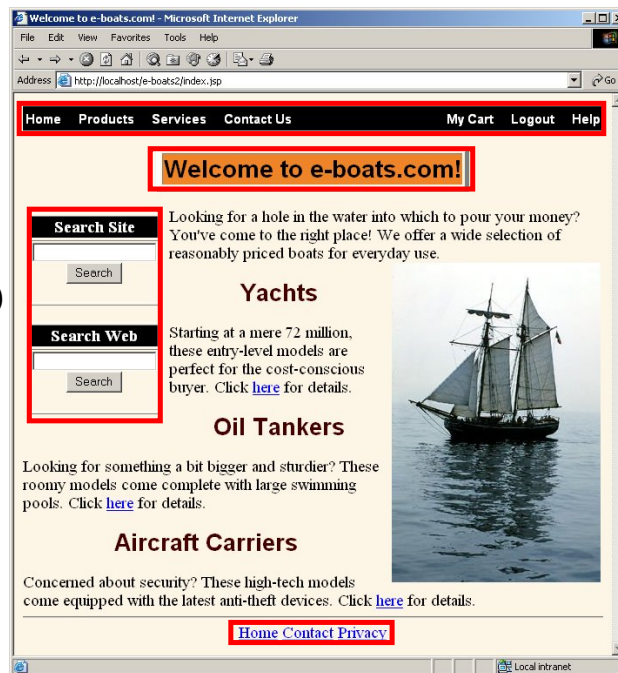
- **Header**
- **Title**
- **Search menu**
- **Body**
   (No outline)
- **Footer**

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="<html:rewrite page='/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>
...
```

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

www.coreservlets.com

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>
```

www.coreservlets.com

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

- **Top-Level Definition**

```xml
<?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>
```

www.coreservlets.com

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

- **Derived Definitions**

```xml
<?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>
```

www.coreservlets.com

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" %>
<tiles:insert definition="carriers.page"/>
```

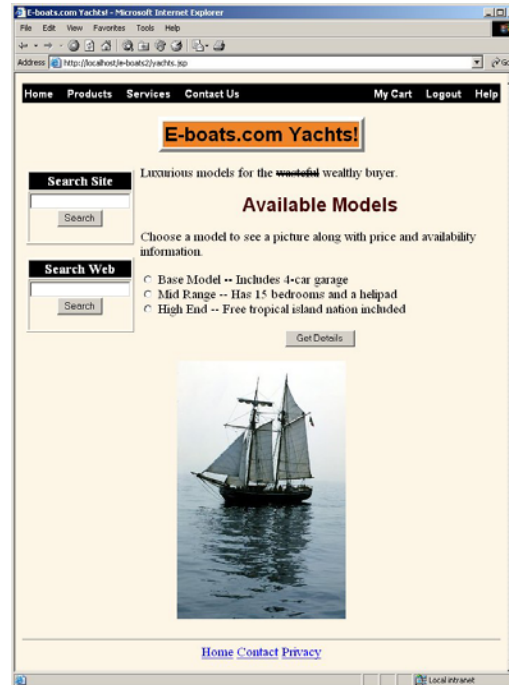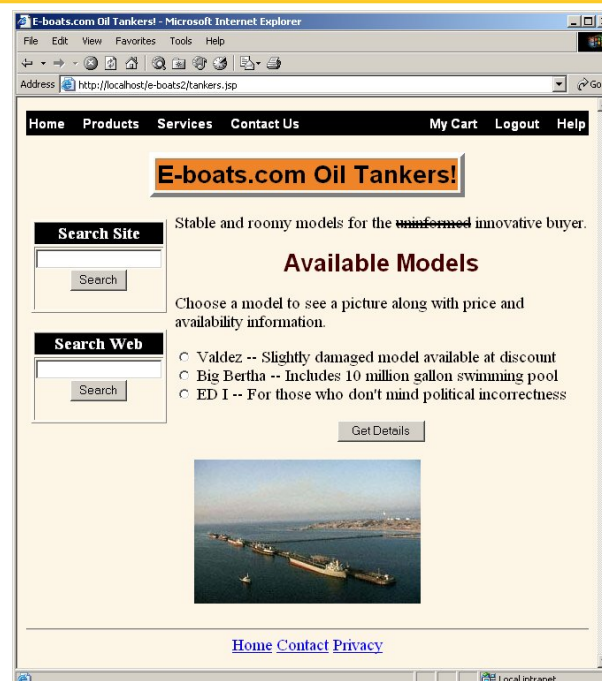# Result: index.jsp

Result: yachts.jsp

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**
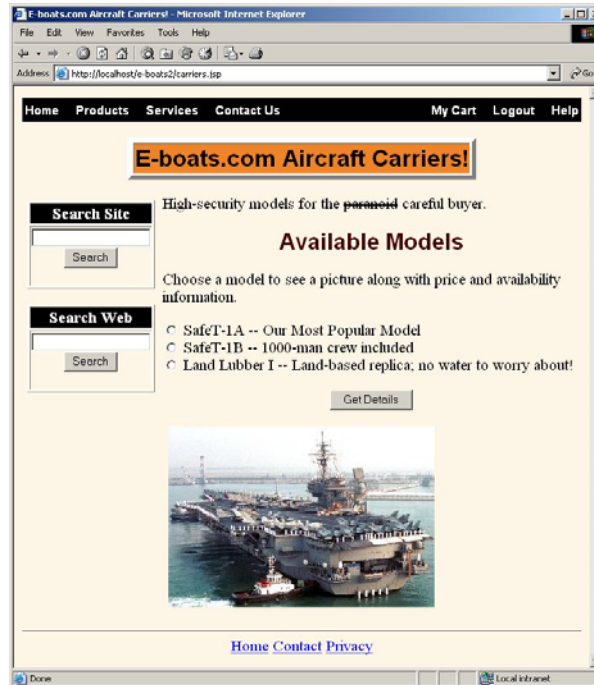
Result: tankers.jsp

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

Result: carriers.jsp

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

Using Definition Names as `forward` Targets

- **You can list definition names instead of JSP page locations in forward declarations**
  - Change:
    ```
    <action path="…" type="…">
      <forward name="…"
               path="/yachts.jsp"/>
    </action>
    ```
  - To:
    ```
    <action path="…" type="…">
      <forward name="…"
               path="yachts.page"/>
    </action>
    ```

- **If JSP pages are only used as forward targets, you can omit the 2-line JSP pages altogether**

Apache Struts: Laying Out Pages with Tiles **www.coreservlets.com**

Summary

- **Basic Usage**
  - Sketch out desired layout
  - Make template file that represents layout
  - Create JSP pages that define layout pieces
    - Use html:rewrite to handle relative URLs
  - 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 this step and refer to definition names in forward declarations

---

Questions?