

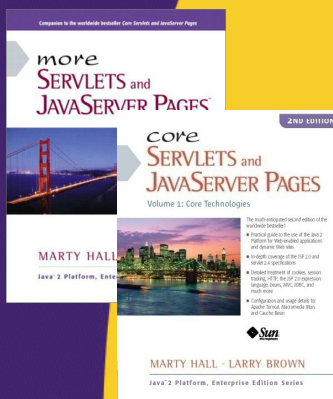


Controlling Web Application Behavior

The Deployment Descriptor: web.xml

Originals of Slides and Source Code for Examples:
<http://courses.coreservlets.com/Course-Materials/msajsp.html>

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live Java EE training, please see training courses at <http://courses.coreservlets.com/>. Servlets, JSP, Struts Classic, Struts 2, JSF 1.x, JSF 2.0, Ajax (with jQuery, Dojo, Prototype, Ext, etc.), GWT, Java 5, Java 6, Spring, Hibernate/JPA, and customized combinations of topics.



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. Contact hall@coreservlets.com for details.

Agenda

- **Location and purpose of web.xml**
- **Custom URLs**
- **Initialization parameters**
 - Servlets
 - JSP Pages
- **Preloading pages**
- **Welcome pages**
- **Error pages**

4

© 2009 Marty Hall



Basics

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

Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Deployment Descriptor: Basics

- **Location**

- Eclipse: WebContent/WEB-INF/web.xml
- Deployed: webAppName/WEB-INF/web.xml
 - *install_dir/conf/web.xml* is Tomcat-specific! Ignore it!

- **When processed**

- Only *required* to be read when server (app) starts
 - Tomcat monitors web.xml and reloads Web app when web.xml changes. Eclipse redeploys app when web.xml changes.

- **Basic format**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation=
           "http://java.sun.com/xml/ns/j2ee web-app_2_4.xsd"
         version="2.4">
```

<!-- "Real" elements go here. All are optional. -->

```
</web-app>
```

You should know how to enter the tags that go in here. But never type the huge header in by hand; let Eclipse create it automatically, or copy from existing example.

6

Eclipse Structure (IDE-specific) vs. Deployment Structure (Standard)

Eclipse

- **Java code**
 - src/subDirMatchingPackage
- **HTML, JSP, Images**
 - WebContent
 - WebContent/randomDir
- **web.xml**
 - WebContent/WEB-INF

Deployed

- **Java code**
 - deployDir/webAppName/WEB-INF/classes/subDirMatchingPackage
- **HTML, JSP, Images**
 - deployDir/webAppName
 - deployDir/webAppName/randomDir
- **web.xml**
 - deployDir/webAppName/WEB-INF
- **Note**
 - On Tomcat, deployDir is tomcat_install_dir/webapps

7

Latest web.xml Version: 2.5

```
<web-app version="2.5"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web=
    "http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
...
</web-app>
```

- Works in Tomcat 6, JBoss 5, Glassfish 3
- Supports the updated (unified) expression language
- Few new features in servlets 2.5 vs. 2.4 or JSP 2.1 vs. JSP 2.0
 - But required for JSF 2.0

8

Most Commonly Used web.xml Version: 2.4

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
...
</web-app>
```

- 2.4 or later required if you use the JSP expression language
- Works in Tomcat 5, Tomcat 6, BEA WebLogic 9.x, Oracle AppServer 10.x, and IBM WebSphere 6.x

9

Older web.xml Version: 2.3

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
    ...
</web-app>
```

- Obsolete: rarely used now
 - Order of entries matters in 2.3 version of web.xml
 - Does not support the JSP expression language
 - Required if using Tomcat 4, BEA WebLogic 8.x, Oracle AppServer 9.x, or IBM WebSphere 5.x

10

The Art of WAR (Files)

- **Idea**
 - When Eclipse deploys to Tomcat, it just builds a folder (described earlier) and puts it in the “webapps” directory
 - But, you can also deploy a single .war file instead of a folder. More convenient when emailing or FTPing
 - All servers must support WAR files
 - WAR files are simply ZIP files
- **Building WAR files**
 - Eclipse can build WAR files automatically
 - R-click project, Export → WAR file
 - You can also do it manually with “jar” or a ZIP utility
- **Deploying WAR files**
 - Location is server specific (“webapps” folder for Tomcat)

11



Custom URLs (Servlet Mappings)

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Defining Custom URLs

- **Java code**

```
package myPackage; ...
public class MyServlet extends HttpServlet { ... }
```
- **web.xml entry (in <web-app...>...</web-app>)**
 - Give name to servlet

```
<servlet>
  <servlet-name>MyName</servlet-name>
  <servlet-class>myPackage.MyServlet</servlet-class>
</servlet>
```
 - Give address (URL mapping) to servlet

```
<servlet-mapping>
  <servlet-name>MyName</servlet-name>
  <url-pattern>/MyAddress</url-pattern>
</servlet-mapping>
```
- **Resultant URL**
 - `http://hostname/webappName/MyAddress`

More Notes on Custom URLs

- **Normal usage**
 - `<url-pattern>/blah</url-pattern>`
 - Should start with /
 - Resultant URL
 - `http://somehost/someApp/blah`
- **Option: can use wildcards for:**
 - File extension (note: no / in this case)
 - `<url-pattern>*.asp</url-pattern>`
 - Directory (still start with /)
 - `<url-pattern>/dir1/dir2/*</url-pattern>`
- **Order matters in web.xml version 2.3 (old!)**
 - All `servlet` entries before any `servlet-mapping` entries

14

Disabling Invoker Servlet

- **Default servlet URL:**
 - `http://host/webAppPrefix/servlet/ServletName`
- **Convenient during development, but wrong for deployment**
 - Init parameters, security settings, filters, etc. are associated only with custom URLs
 - Default URL is long and cumbersome
 - You might want to hide implementation details
- **Disabling it:**
 - In each Web application, redirect requests to other servlet
 - ```
<servlet-mapping>
 <servlet-name>...</servlet-name>
 <url-pattern>/servlet/*</url-pattern>
</servlet-mapping>
```
  - Globally
    - Server-specific mechanism

15

# Disabling Invoker Servlet: Example

```
public class NoInvokerServlet extends HttpServlet {
 public void doGet(HttpServletRequest request,
 HttpServletResponse response)
 throws ServletException, IOException {
 ...
 String title = "Invoker Servlet Disabled.";
 out.println
 (docType +
 "<HTML>\n" +
 "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
 "<BODY BGCOLOR=\"#FDF5E6\">\n" +
 "<H2>" + title + "</H2>\n" +
 "Sorry, access to servlets by means of\n" +
 "URLs that begin with\n" +
 "http://host/webAppPrefix/servlet/\n" +
 "has been disabled.\n" +
 "</BODY></HTML>");
 }

 public void doPost(...) { // call doGet }
}
```

16

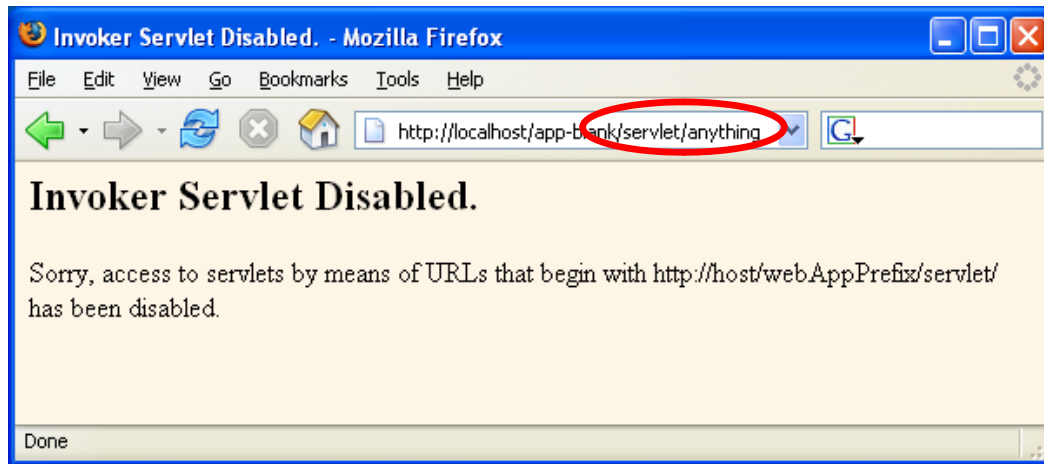
# Disabling Invoker Servlet: Example (Continued)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app ...>
 <!-- Disable the invoker servlet. -->
 <servlet>
 <servlet-name>NoInvoker</servlet-name>
 <servlet-class>
 coreservlets.NoInvokerServlet
 </servlet-class>
 </servlet>
 <servlet-mapping>
 <servlet-name>NoInvoker</servlet-name>
 <url-pattern>/servlet/*</url-pattern>
 </servlet-mapping>
</web-app>
```

17



## Disabling Invoker Servlet: Example (Continued)



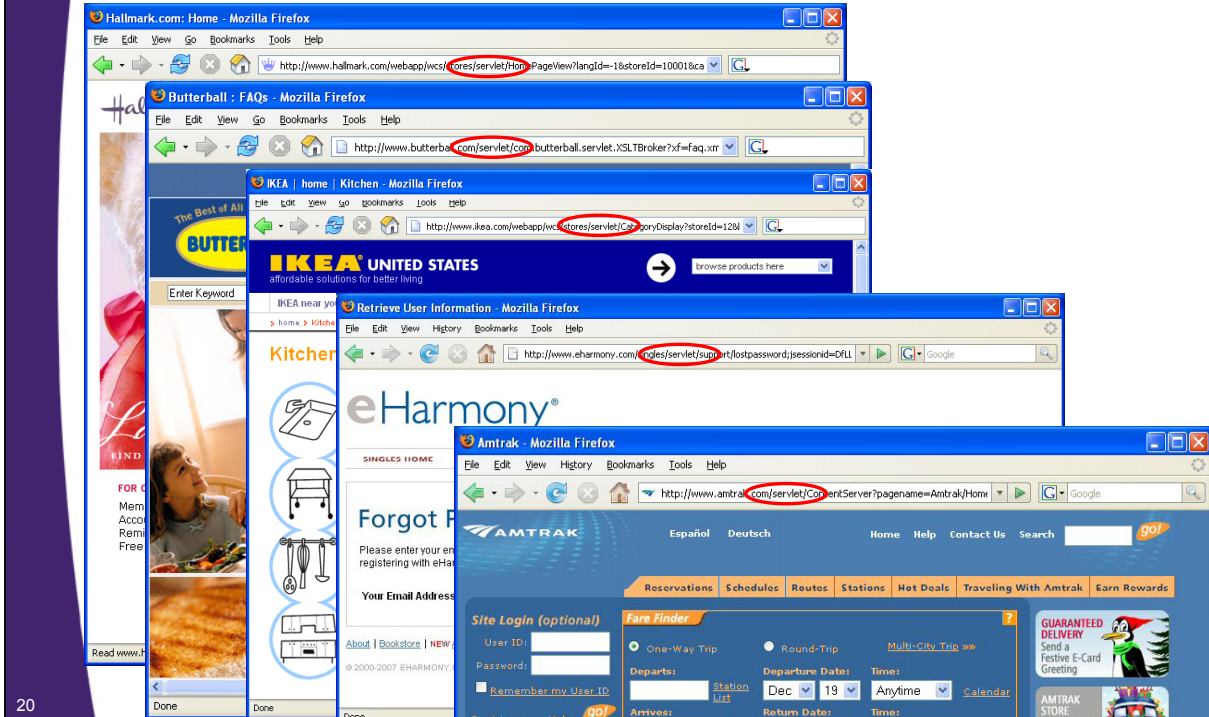
18

## Failing to Define Custom URLs

- You should always use custom URLs on deployed projects
  - URLs look cleaner and simpler and shorter
  - URLs have more meaningful names
  - You don't expose possibly proprietary class file names
  - You can use web.xml to assign init params later
    - Does not work with .../servlet/myPackage.MyServlet
  - You can apply filters and security settings later (via web.xml) in a more predictable and controllable manner
  - Most importantly of all, you can avoid being added to Marty's "Hall of Shame"
    - The kiss of death for any self-respecting Java EE developer

19

# The Hall of Shame (Deployed Sites with Ugly .../servlet/... URLs)



20

© 2009 Marty Hall



## Init Params

**Customized Java EE Training:** <http://courses.coreservlets.com/>  
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

# Who Needs to Customize Servlet Behavior?

- **Author**
  - Change the actual code
- **End user**
  - Supply value in HTML form
- **Deployer**
  - Put initialization values in *web.xml*
- **Parallels applet behavior**
  - Author changes the code
  - End user manipulates GUI controls
  - Deployer uses PARAM element inside APPLET element in the HTML file.

22

# Types of Initialization

- **Servlets**
  - Call ServletConfig.getInitParameter from the init method
- **JSP pages**
  - Call ServletConfig.getInitParameter from the jspInit method. Use jsp-file instead of servlet-class.
- **Servlet context**
  - Call ServletContext.getInitParameter
  - What method to call this from? See lecture on listeners!
- **Filters**
  - See later lecture
- **Listeners**
  - See later lecture

23

# Assigning Init Params: Problems with Invoker Servlet

```
<servlet>
 <servlet-name>InitTest</servlet-name>
 <servlet-class>moreservlets.InitServlet</servlet-class>
 <init-param>
 <param-name>firstName</param-name>
 <param-value>Larry</param-value>
 </init-param>
 <init-param>
 <param-name>emailAddress</param-name>
 <param-value>ellison@microsoft.com</param-value>
 </init-param>
</servlet>
<servlet-mapping>
 <servlet-name>InitTest</servlet-name>
 <url-pattern>/showInitValues</url-pattern>
</servlet-mapping>
```

24

# Reading Servlet Initialization Parameters

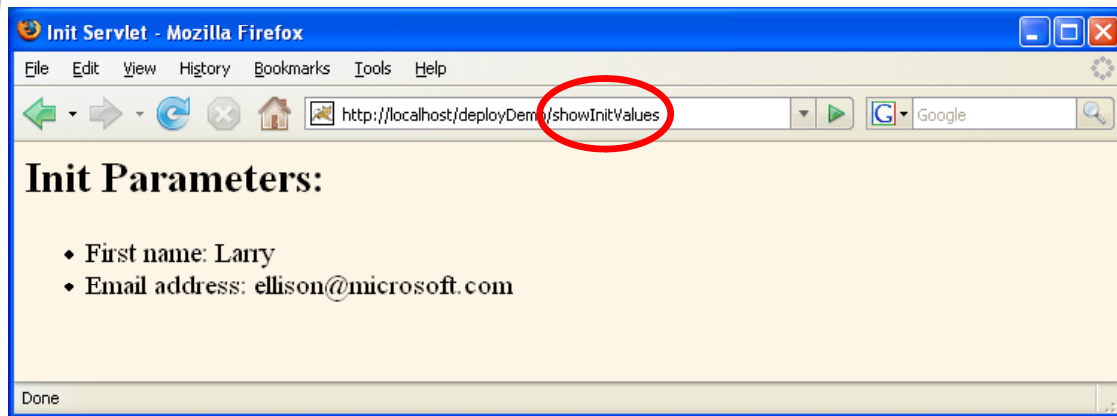
```
public class InitServlet extends HttpServlet {
 private String firstName, emailAddress;

 public void init() {
 ServletConfig config = getServletConfig();
 firstName =
 config.getInitParameter("firstName");
 if (firstName == null) {
 firstName = "Missing first name";
 }
 emailAddress =
 config.getInitParameter("emailAddress");
 if (emailAddress == null) {
 emailAddress = "Missing email address";
 }
 }

 public void doGet(...) ... { ... }
}
```

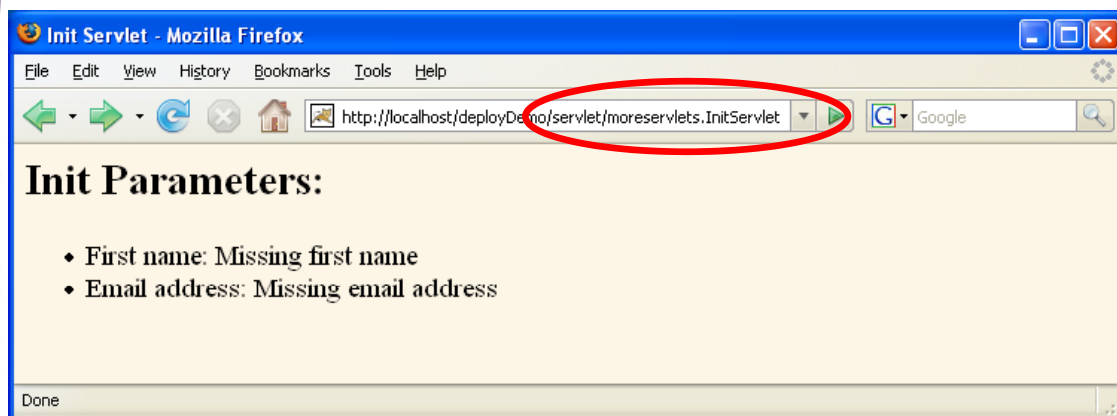
25

# Servlet Initialization Parameters: Successful Result



26

# Servlet Initialization Parameters: Failed Result



- **One address works; one fails**
- **This is too hard to remember!**
  - Disable invoker servlet, so there is only one address: the one that works!

27



# Assigning JSP Initialization Parameters

```
<servlet>
 <servlet-name>InitPage</servlet-name>
 <jsp-file>/InitPage.jsp</jsp-file>
 <init-param>
 <param-name>firstName</param-name>
 <param-value>Bill</param-value>
 </init-param>
 <init-param>
 <param-name>emailAddress</param-name>
 <param-value>gates@oracle.com</param-value>
 </init-param>
</servlet>
```

28

# Assigning JSP Initialization Parameters (Continued)

```
<servlet-mapping>
 <servlet-name>InitPage</servlet-name>
 <url-pattern>/InitPage.jsp</url-pattern>
</servlet-mapping>
```

- **If you leave invoker turned on and have declaration on previous page:**
  - Initialized JSP page could be accessed with *http://host/webAppPrefix/servlet/InitPage*. Yuck!
  - Assign URL back to original URL of JSP page instead.

29

# Reading JSP Initialization Parameters

```
...

 First name: <%= firstName %>
 Email address: <%= emailAddress %>

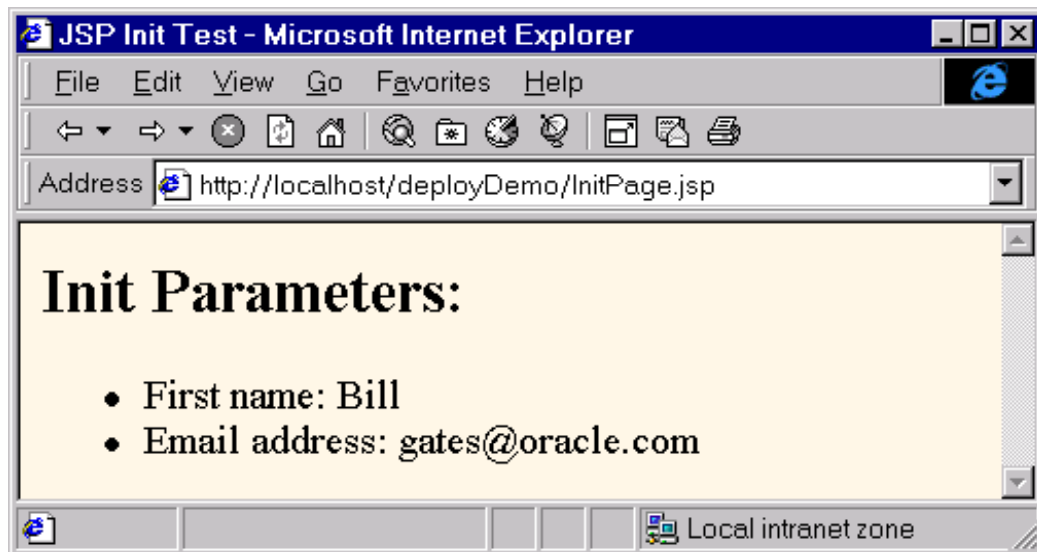
...
<%!
private String firstName, emailAddress;

public void jspInit() {
 ServletConfig config = getServletConfig();
 firstName = config.getInitParameter("firstName");
 if (firstName == null) { firstName = "No first name"; }
 emailAddress = config.getInitParameter("emailAddress");
 if (emailAddress == null) { emailAddress = "No email"; }
}
%>
```

This results in ugly and hard-to-maintain JSP pages. Consider using MVC and never using direct init params in JSP.

30

## JSP Initialization Parameters: Result



31

## Assigning Application-Wide Initialization Parameters

- web.xml element: context-param

```
<context-param>
 <param-name>support-email</param-name>
 <param-value>blackhole@mycompany.com</param-value>
</context-param>
```

- Read with the `getInitParameter` method of **ServletContext** (not **ServletConfig**)
- Problem: who should call `getInitParameter`?
  - `load-on-startup` gives partial solution
  - Listeners give much better answer

32

## Loading Servlets or JSP Pages When Server Starts

- What if servlet or JSP page defines data that other resources use?

```
<servlet>
 <servlet-name>...</servlet-name>
 <servlet-class>...</servlet-class>
 <!-- Or jsp-file instead of servlet-class -->
 <load-on-startup>1</load-on-startup>
</servlet>
```

- You can also specify relative order of multiple preloaded resources

```
<load-on-startup>1</load-on-startup>
...
<load-on-startup>2</load-on-startup>
```

33



# Welcome and Error Pages

**Customized Java EE Training:** <http://courses.coreservlets.com/>  
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Specifying Welcome Pages

- **What result do you get for `http://host/webAppPrefix/someDirectory/`?**
  - `index.jsp`?
  - `index.html`?
  - `index.htm`?
  - `default.htm`?
  - A 404 error?
  - A directory listing?

– Answer: it depends on the server.
- **Make at least the file ordering portable:**

```
<welcome-file-list>
 <welcome-file>index.jsp</welcome-file>
 <welcome-file>index.html</welcome-file>
</welcome-file-list>
```

# Designating Pages to Handle Errors

- **Pages to use for specific HTTP status codes**
  - Use the `error-code` element
    - Within `error-page`
- **Pages to use when specific uncaught exceptions are thrown**
  - Use the `exception-type` element
    - Within `error-page`
- **Page-specific error pages**
  - Use `<%@ page errorPage="Relative URL" %>`
    - In individual JSP page, not in `web.xml`

36

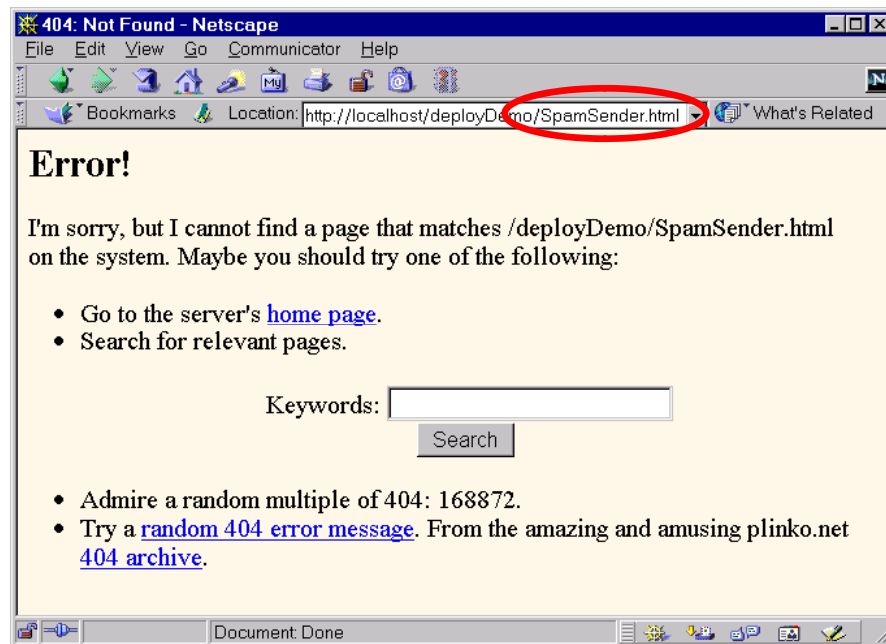
# Error Pages and Status Codes

```
<web-app...>
 <error-page>
 <error-code>404</error-code>
 <location>/WEB-INF/NotFound.jsp</location>
 </error-page>
 ...
</web-app>
```

37



# Error Pages and Status Codes: Result



38

## A Dangerous Computation

```
package moreservlets;

/** Exception used to flag particularly onerous
 programmer blunders. Used to illustrate the
 exception-type web.xml element.
 */

public class DumbDeveloperException extends Exception {
 public DumbDeveloperException() {
 super("Duh. What was I *thinking*?");
 }

 public static int dangerousComputation(int n)
 throws DumbDeveloperException {
 if (n < 5) {
 return(n + 10);
 } else {
 throw(new DumbDeveloperException());
 }
 }
}
```

39

# A Risky Page

```
<!DOCTYPE HTML PUBLIC
 "-//W3C//DTD HTML 4.0 Transitional//EN">

<HTML>
<HEAD><TITLE>Risky JSP Page</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2>Risky Calculations</H2>
<%@ page import="moreservlets.*" %>
<% int n = ((int)(10 * Math.random())); %>

 n: <%= n %>
 dangerousComputation(n) :
 <%= DumbDeveloperException.dangerousComputation(n) %>

</BODY></HTML>
```

40

# Declaring Error Page for DDE

```
<web-app...>
 <error-page>
 <exception-type>
 moreservlets.DumbDeveloperException
 </exception-type>
 <location>/WEB-INF/DDE.jsp</location>
 </error-page>
 ...
</web-app>
```

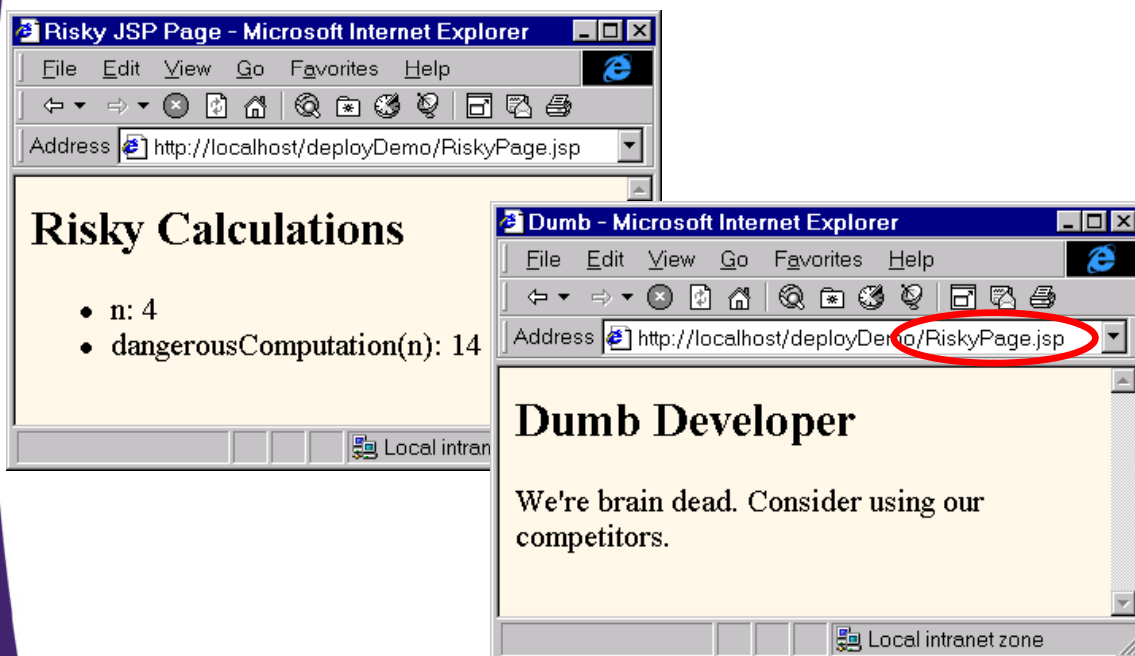
41

# WEB-INF/DDE.jsp

```
<!DOCTYPE HTML PUBLIC
 "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Dumb</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2>Dumb Developer</H2>
We're brain dead. Consider using our competitors.
</BODY></HTML>
```

42

## Error Pages and Exceptions: Results



43



## Other Capabilities

**Customized Java EE Training:** <http://courses.coreservlets.com/>  
Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Session Timeouts

- **You can explicitly deactivate sessions**
  - `session.invalidate()`
- **You can also set session timeout**
  - `session.setMaxInactiveInterval(...)`
- **You can set Web-app-wide default timeout**

```
<session-config>
 <session-timeout>
 time-in-minutes
 </session-timeout>
</session-config>
```

  - A value of 0 or negative number indicates that default sessions should never automatically time out
- **If no session-config**
  - Default session timeout is server-specific

## Deactivating Scripting or Expression Language (2.4+ only)

- **Disabling scripting**

- Used when you want to enforce pure-MVC approach

```
<jsp-property-group>
 <url-pattern>*.jsp</url-pattern>
 <scripting-invalid>true</scripting-invalid>
</jsp-property-group>
```

- **Disabling Expression Language**

- Used when you have JSP 1.2 pages that might accidentally contain `${blah}`. Note that EL is disabled automatically if you use version 2.3 of web.xml.

```
<jsp-property-group>
 <url-pattern>*.jsp</url-pattern>
 <el-ignored>true</el-ignored>
</jsp-property-group>
```

46

## Other web.xml Capabilities

- **Documenting Web app**

- icon, display-name, description

- **Mapping files to MIME types**

- mime-mapping

- **Allowing execution on multiple systems in cluster**

- distributable

- **Setting encodings for groups of pages (2.4 only)**

- page-encoding within jsp-property-group

- **Implicit includes (2.4 only)**

- include-prelude, include-coda within jsp-property-group
  - Includes files at beginning/end of each of set of JSP pages

- **More that we'll see later in the course**

- Designating security settings
- Declaring filters
- Setting up listeners
- Specifying tag library validators

47



# Summary

- **URLs**
  - servlet (servlet-class, servlet-name, load-on-startup)
  - servlet-mapping (servlet-name, url-pattern)
- **Init parameters**
  - init-param
- **Welcome pages**
  - welcome-file-list
- **Error pages**
  - error-page (error-code, exception-type)
- **Default session timeouts**
  - session-config (session-timeout)
- **Disabling scripting or EL**
  - jsp-property-group (url-pattern, scripting-invalid/el-ignored)

48

© 2009 Marty Hall



## Questions?

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

Servlets, JSP, JSF 1.x & JSF 2.0, Struts Classic & Struts 2, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.