



Controlling Web Application Behavior

The Deployment Descriptor: web.xml

JSP, Servlet, Struts, JSF, AJAX, & Java 5 Training: <http://courses.coreservlets.com>
J2EE Books from Sun Press: <http://www.coreservlets.com>



For live J2EE training, see training courses on JSP, servlets, Struts, JSF, AJAX, and Java 5 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.
Additional topics available upon request.

Agenda

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

4

J2EE training: <http://courses.coreservlets.com>

Deployment Descriptor: Basics

- **Filename:** web.xml
- **Location:** *yourWebApp/WEB-INF*
 - *install_dir/conf/web.xml* is Tomcat-specific!
- **Read: usually only when server starts**
 - Many servers have "hot deploy" option
 - Tomcat 5.5 monitors web.xml and reloads Web 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>
```

5

J2EE training: <http://courses.coreservlets.com>

Deployment Descriptors: Version 2.4 vs. 2.3

• 2.4: Needed for 2.4 features

```
<?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">
```

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

• 2.3: Works in 2.3 and 2.4

```
<?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>
```

- Order of entries matters in 2.3 version of web.xml

6

J2EE training: <http://courses.coreservlets.com>

Complete web.xml Ordering (Matters in 2.3 Version Only)

- icon
- display-name
- description
- distributable
- context-param
- filter
- filter-mapping
- listener
- servlet
- servlet-mapping
- session-config
- mime-mapping

7

J2EE training: <http://courses.coreservlets.com>

Complete web.xml Ordering (Continued)

- welcome-file-list
- error-page
- taglib
- resource-env-ref
- resource-ref
- security-constraint
- login-config
- security-role
- env-entry
- ejb-ref
- ejb-local-ref

8

J2EE training: <http://courses.coreservlets.com>

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`

9

J2EE training: <http://courses.coreservlets.com>

More Notes on Custom URLs

- **Can use wildcards for:**
 - File extension
 - `<url-pattern>*.asp</url-pattern>`
 - Directory
 - `<url-pattern>/dir1/dir2/*</url-pattern>`
- **Order matters in version 2.3**
 - servlet before servlet-mapping
- **Init params for JSP**
 - Use jsp-file instead of servlet-class

10

J2EE training: <http://courses.coreservlets.com>

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

11

J2EE training: <http://courses.coreservlets.com>

# 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 }
}
```

12

J2EE training: <http://courses.coreservlets.com>

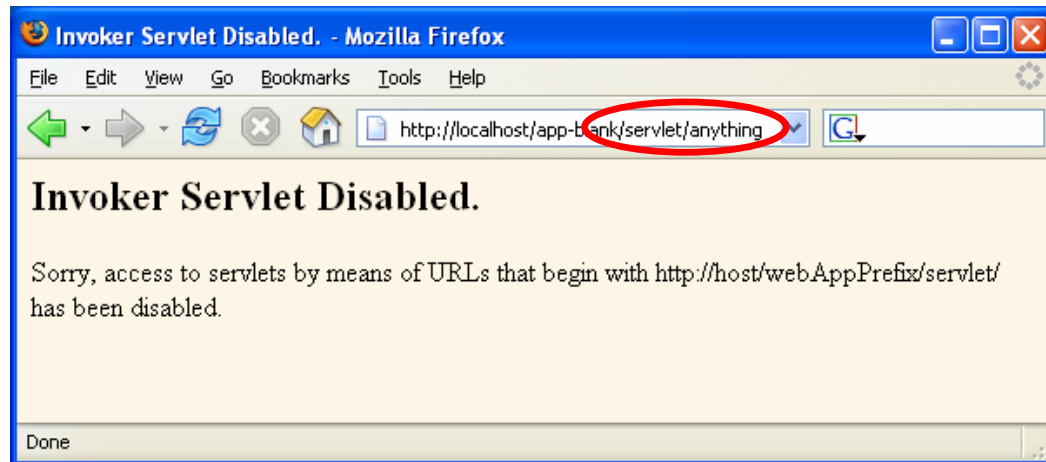
# Disabling Invoker Servlet: Example (Continued)

```
<?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>
```

13

J2EE training: <http://courses.coreservlets.com>

# Disabling Invoker Servlet: Example (Continued)

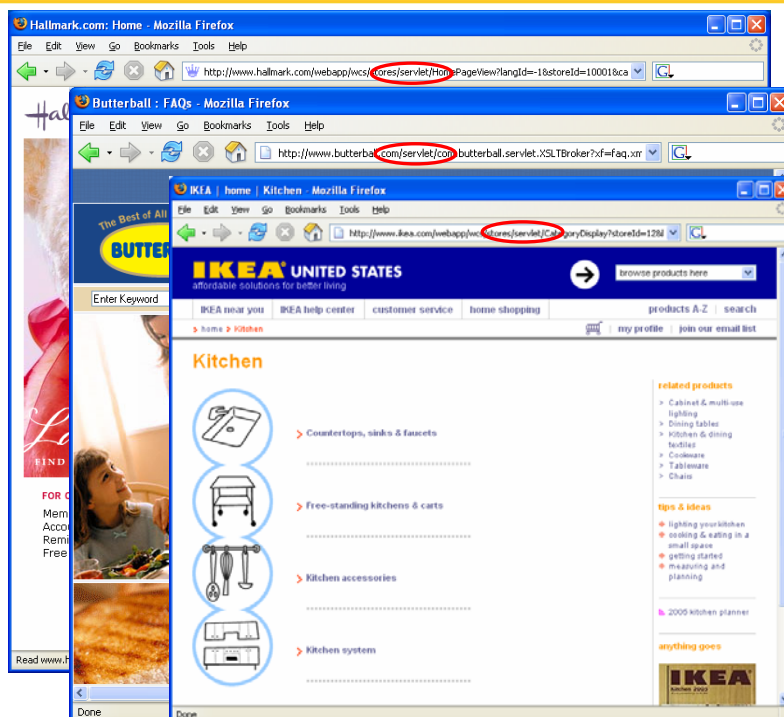


14

J2EE training: <http://courses.coreservlets.com>

# Failing to Disable Invoker Servlet: Definite Deployment-Time Error

- **Causes problems with**
  - Init params
  - Security settings
  - Filters



15

# Making Your Own Web Apps

- **Start with the "app-blank" Web app**
  - From <http://archive.moreservlets.com>
  - Contains
    - WEB-INF
    - WEB-INF/web.xml
    - WEB-INF/classes
- **Copy it to yourWebAppName**
  - Leave yourWebAppName in your development directory (e.g., C:\Servlets+JSP) and add servlets, JSP, etc.
- **Deploy and test**
  - Copy yourWebAppName onto the shortcut to webapps
  - If you are using Tomcat 5.0.x or earlier and you modified web.xml, you also need to restart the server

16

J2EE training: <http://courses.coreservlets.com>

# 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.

17

J2EE training: <http://courses.coreservlets.com>

# Types of Initialization

- **Servlets**
  - Call `ServletConfig.getInitParameter` from the `init` method
- **JSP pages**
  - Call `ServletConfig.getInitParameter` from the `jspInit` method
- **Servlet context**
  - Call `ServletContext.getInitParameter`
  - What method to call this from? See lecture on listeners!
- **Filters**
  - See later lecture
- **Listeners**
  - See later lecture

18

J2EE training: <http://courses.coreservlets.com>

## 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>
```

19

J2EE training: <http://courses.coreservlets.com>

# Reading Servlet Initialization Parameters

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

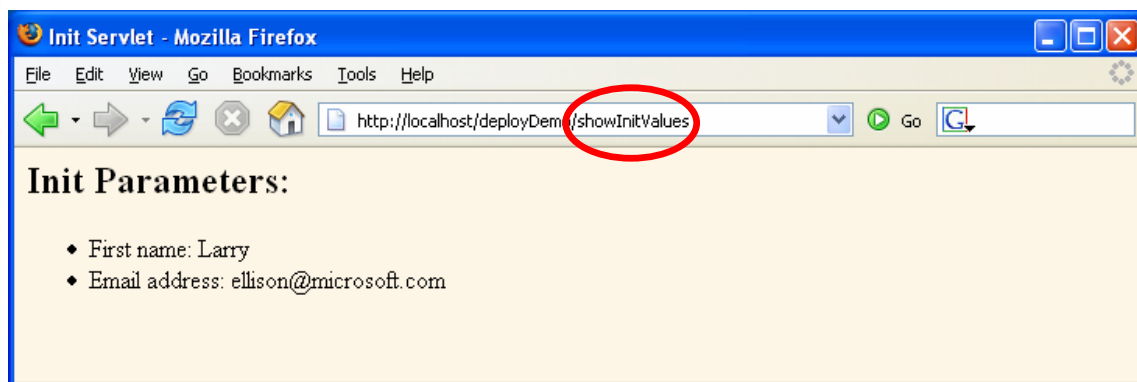
 public void init() {
 ServletConfig config = getServletConfig();
 firstName =
 config.getInitParameter("firstName");
 emailAddress =
 config.getInitParameter("emailAddress");
 }

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

20

J2EE training: <http://courses.coreservlets.com>

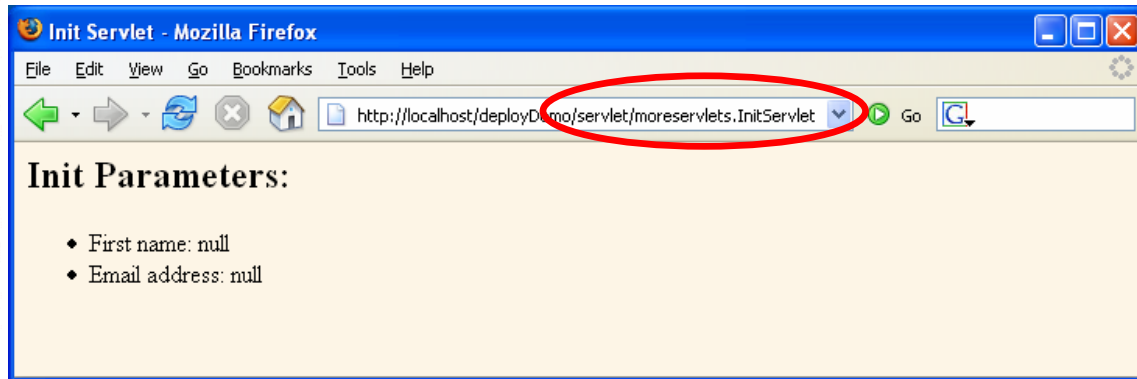
## Servlet Initialization Parameters: Successful Result



21

J2EE training: <http://courses.coreservlets.com>

# Servlet Initialization Parameters: Failed Result



- One address works; one fails
- This is too hard to remember!
  - Disable invoker servlet, so there is only 1 address: the 1 that works!

22

J2EE training: <http://courses.coreservlets.com>

# 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>
```

23

J2EE training: <http://courses.coreservlets.com>

## 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.

## Reading JSP Initialization Parameters

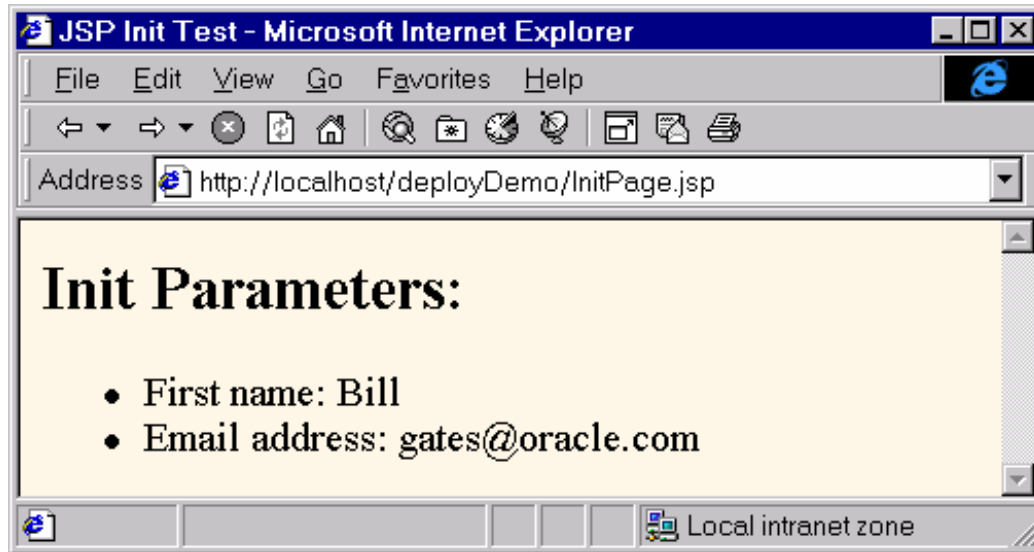
```
...

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

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

public void jspInit() {
 ServletConfig config = getServletConfig();
 firstName =
 config.getInitParameter("firstName");
 emailAddress =
 config.getInitParameter("emailAddress");
}
%>
```

# JSP Initialization Parameters: Result



26

J2EE training: <http://courses.coreservlets.com>

## 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

27

J2EE training: <http://courses.coreservlets.com>

## 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/>
</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>
```

## 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>
```

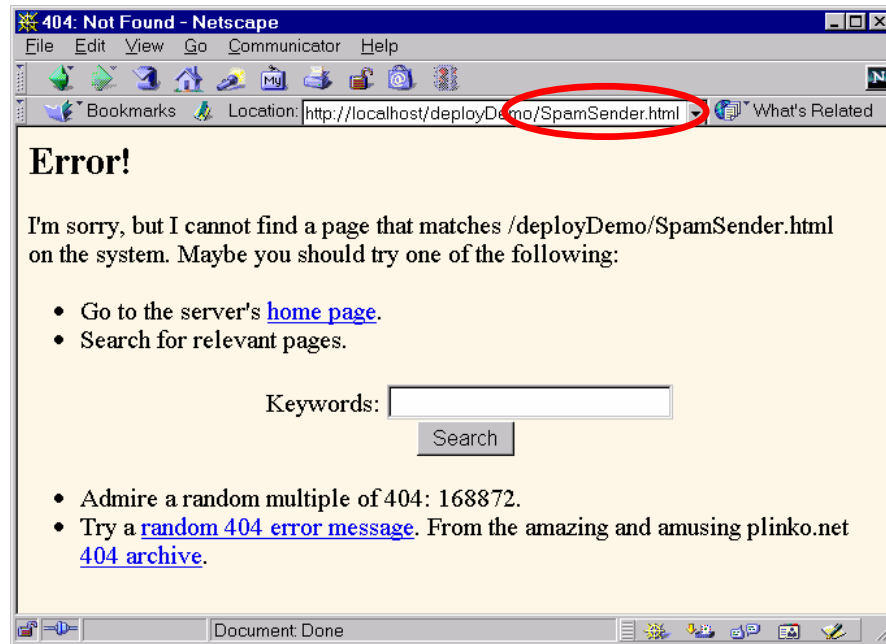
# Designating Pages to Handle Errors

- **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*

# Error Pages and Status Codes

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

# Error Pages and Status Codes: Result



32

J2EE training: <http://courses.coreservlets.com>

## 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());
 }
 }
}
```

33

J2EE training: <http://courses.coreservlets.com>

# 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>
```

# 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>
```

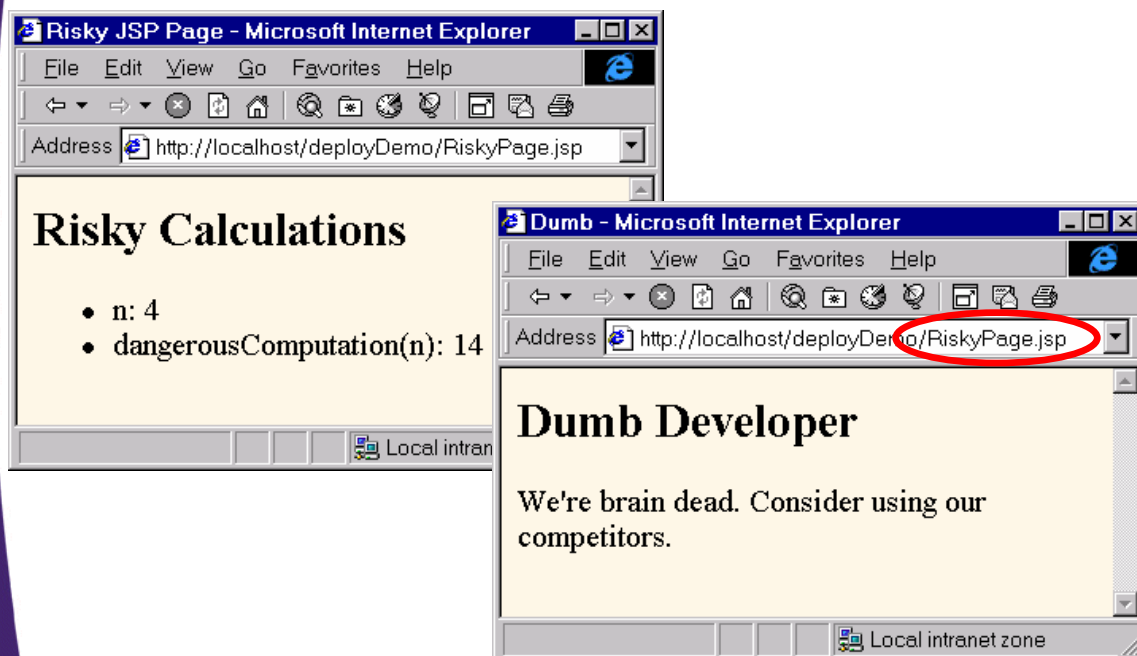
# WEB-INF/DDE.jsp

```
<!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>
```

36

J2EE training: <http://courses.coreservlets.com>

## Error Pages and Exceptions: Results



37

J2EE training: <http://courses.coreservlets.com>

# 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

38

J2EE training: <http://courses.coreservlets.com>

# Deactivating Scripting or Expression Language (ver. 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>
```

39

J2EE training: <http://courses.coreservlets.com>

# 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

40

J2EE training: <http://courses.coreservlets.com>

# Summary

- **URLs**
  - servlet (servlet-class, servlet-name)
  - 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)

41

J2EE training: <http://courses.coreservlets.com>



# Questions?

**Servlet, JSP, Struts, and JSF Training Courses:**  
**[courses.coreservlets.com](http://courses.coreservlets.com)**

**Core Servlets & JSP book: [www.coreservlets.com](http://www.coreservlets.com)**

**More Servlets & JSP book: [www.moreservlets.com](http://www.moreservlets.com)**

**JSP, Servlet, Struts, JSF, AJAX, & Java 5 Training: <http://courses.coreservlets.com>**

**J2EE Books from Sun Press: <http://www.coreservlets.com>**