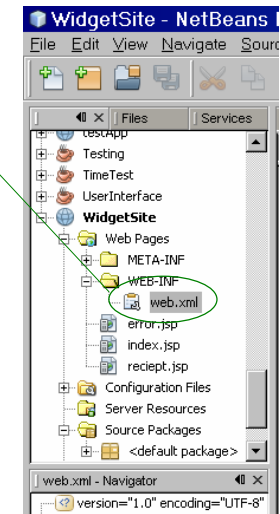


# Server-side Web Development and Programming

## Lecture 6: Java Servlets and the web.xml Configuration File

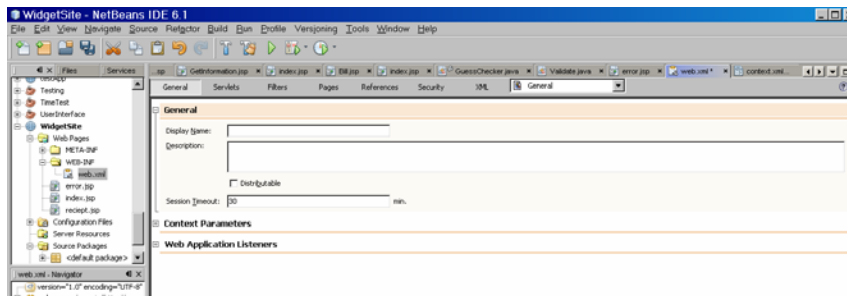
# Configuration Files

- Each webapp has a web.xml file
  - In WEB-INF subdirectory
- Defines:
  - Mapping from servlet names to servlet files
  - Global constants
  - Default error pages
  - Security roles...



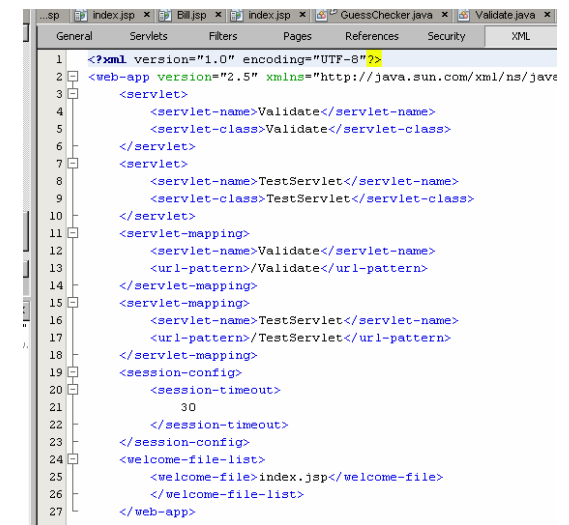
# The web.xml file

- Complex (and finicky) XML
- If buggy, application will not load
  - Errors displayed in Tomcat window when started
- NetBeans provides interface to manipulate its values



# The web.xml file

- Internally, still XML
  - Can view and edit at XML tab
  - Basic form of XML:  
<tag>text</tag> or <tag/>
  - Tags must be properly balanced and nested
  - Must have a root tag <webapp>



# Welcome Page Properties

- Page shown at webapp startup
  - Default: index.jsp

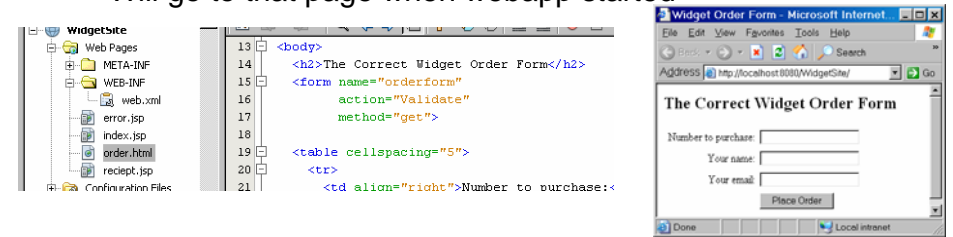


- Can change to a new welcome file



# New Welcome Page

- Will go to that page when webapp started



- Note change in web.xml

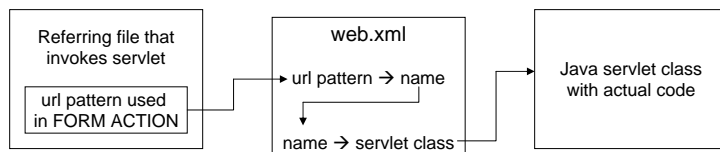
```

22 </session-timeout>
23 </session-config>
24 <welcome-file-list>
25 <welcome-file>order.html</welcome-file>
26 </welcome-file-list>
27 </web-app>
28
        
```

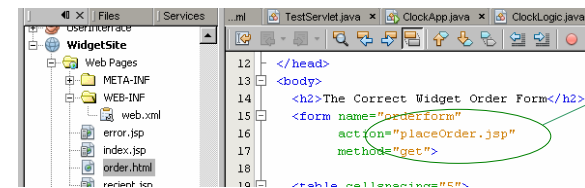
# Servlet Mapping

Web.xml links referrer page to be linked with actual servlet class

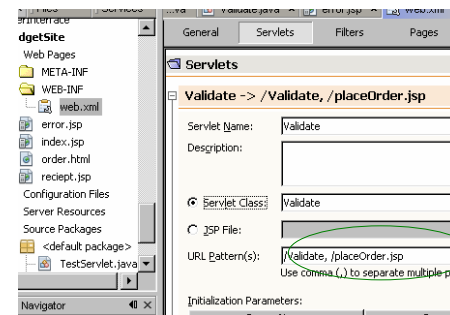
- Mapping from a name to a java class
  - Allows servlet code to be changed without having to rewrite other pages
    - Important since name of class = name of file in Java
- Mapping from a url pattern to a servlet name
  - Allows servlets to be "hidden" from user



# Servlet Mapping

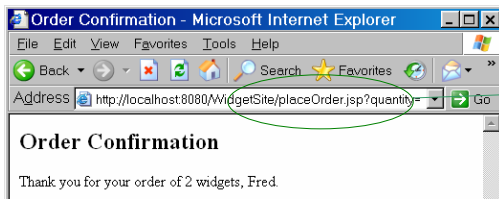


No such file – this is just a url pattern that will refer to an actual servlet via web.xml



Added to the url patterns that map to this servlet

# Servlet Mapping



Note that the url pattern appears in the browser

```

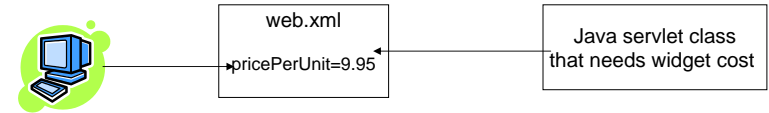
web.xml
.jsp
<.jsp
<.jsp
.r.html
pt.jsp
ation Files
esources
ackages
'ault package>
'estServlet.java
encoding="UTF-8"
xmlns="http://www.
apping
apping
apping
efin
4 <!-- Servlet Name -->
5 <!-- Servlet Class -->
6 </!-- Servlet -->
7 <!-- Servlet -->
8 <!-- Servlet Name -->
9 <!-- Servlet Class -->
10 </!-- Servlet -->
11 <!-- Servlet Mapping -->
12 <!-- Servlet Name -->
13 <!-- Servlet URL Pattern -->
14 </!-- Servlet Mapping -->
15 <!-- Servlet Mapping -->
16 <!-- Servlet Name -->
17 <!-- Servlet URL Pattern -->
18 </!-- Servlet Mapping -->
19 <!-- Servlet Mapping -->
20 <!-- Servlet Name -->
21 <!-- Servlet URL Pattern -->
22 </!-- Servlet Mapping -->

```

Code in web.xml that does the servlet mapping

# Initial Parameters

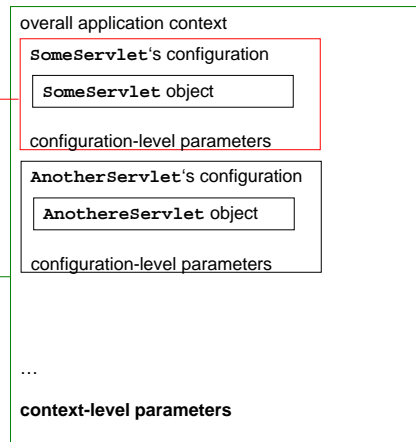
- Global constants stored in web.xml
  - Better than “hardwiring” values directly into the code
    - Easier for nonprogrammer to modify
  - Form like parameter: name/value
    - Example: “price per unit” in widget site



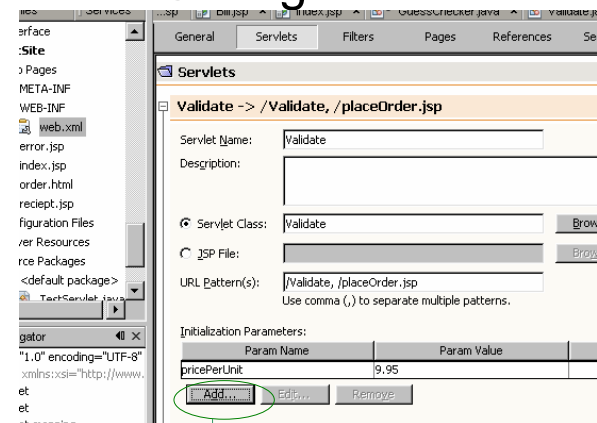
- Note that such information is usually in a database
- However, the name of the database is often an initial parameter!

# Initial Parameters

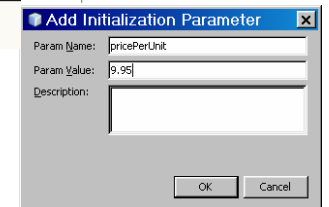
- Configuration level
  - Specific to single servlet
- Context level
  - General to all server pages/servlets



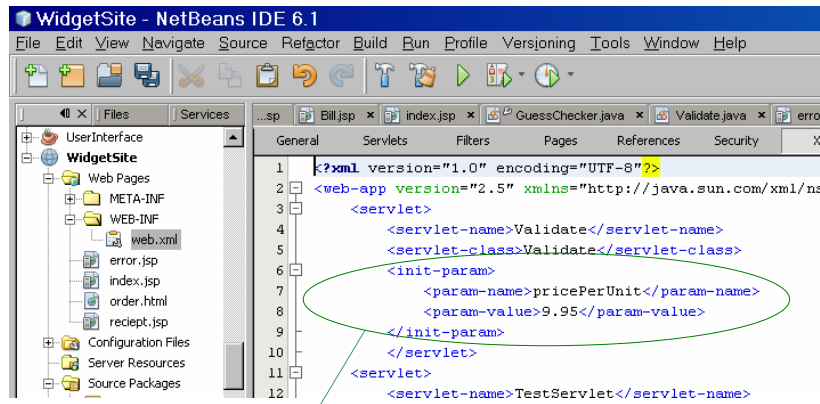
# Configuration Parameters



Can create in servlet tab



# Configuration Parameters



Form of xml tag

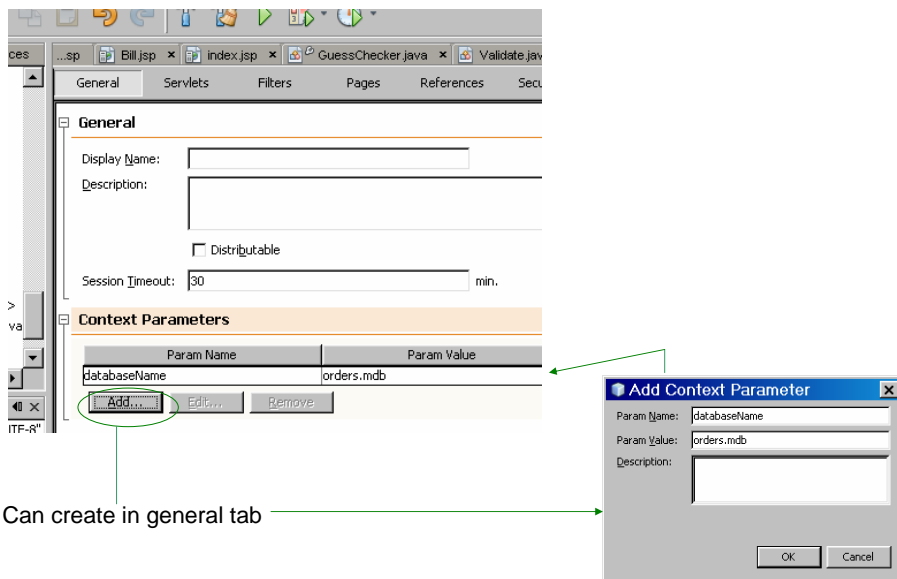
# Configuration Parameters

- Accessing from servlet:
  - Get the servlet configuration  
ServletConfig config = getServletConfig();
  - Get the value corresponding to the name of the parameter  
String value = config.getInitParameter("name");

```

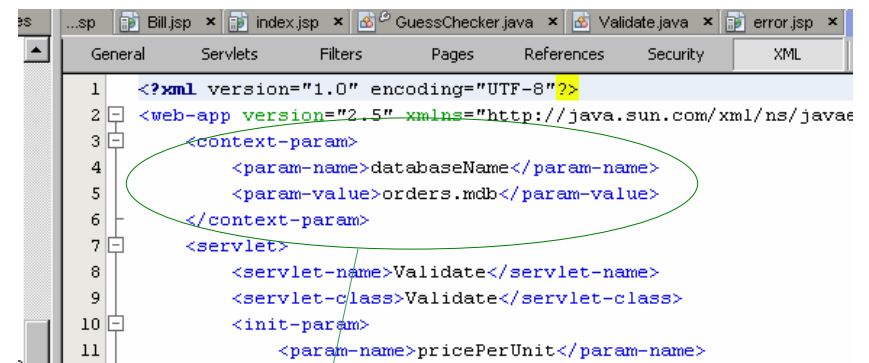
31 }
32
33 else {
34     ServletConfig config = getServletConfig();
35     String priceString = config.getInitParameter("pricePerUnit");
36     double pricePerUnit = Double.parseDouble(priceString);
37     int quantityNumber = Integer.parseInt(quantity);
38     double totalCost = pricePerUnit * quantityNumber;
39     request.setAttribute("pricePerUnit", ""+pricePerUnit);
40     request.setAttribute("cost", ""+totalCost);
41     url = "/receipt.jsp";
42 }
    
```

# Context Parameters



Can create in general tab

# Context Parameters



Form of xml tag

# Configuration Parameters

- Accessing from servlet:
  - Get the servlet configuration  
`ServletConfig config = getServletConfig();`
  - Get the webapp context from the configuration  
`ServletContext context = config.getServletContext();`
  - Get the value corresponding to the name of the parameter  
`String value = context.getInitParameter("name");`
- Accessing from a JSP:
  - `String value = application.getInitParameter("name");`

Built-in object in JSP  
(like "request")

# Configuration Parameters

```
12 </head>
13 <body>
14
15 <%
16     String name = request.getParameter("customerName");
17     String email = request.getParameter("customerEmail");
18     String quantity = request.getParameter("quantity");
19     String totalCost = (String)request.getAttribute("cost");
20     String pricePerUnit = (String)request.getAttribute("pricePerUnit");
21     String databaseName = application.getInitParameter("databaseName");
22 %>
23
24 <h2>Order Confirmation</h2>
25
26 <p>
27     Thank you for your order of <%= quantity %> widgets, <%= pricePerUnit %>
28 </p>
29 <p>
30     At <%= pricePerUnit %>, your bill will be $<%= totalCost %>.
31 </p>
32 <p>
33     You will shortly receive an email confirmation at <%= email %>.
34     Your order will be stored in our <%= databaseName %> database.
35 </p>
36 </body>
```