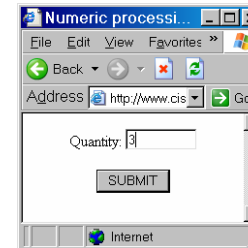


Server-side Web Development and Programming

Lecture 3: Introduction to Java Server Pages

Form Handling

- Form data appended to request string



```
<FORM NAME="purchaseform"
METHOD=GET
ACTION= http://localhost:8080/WidgetSite/receipt.jsp >
Quantity: <INPUT TYPE="text" SIZE="8" NAME="quantity" />
<BR /><BR />
<INPUT TYPE="submit" VALUE="SUBMIT">
</FORM>
```

Generates the request:

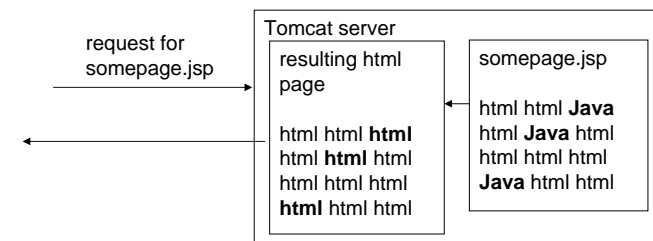
Form Handling

Goal:

- Extract values of parameters from the request
- Generate appropriate response page based on parameter values
- Take other appropriate action based on request
 - Add to shopping cart
 - Modify or query database

Server Page Model

- Html document with executable code interspersed
- When page requested:
 - Code executed
 - Html generated and inserted in its place
 - Final all html document sent back as response



Java Server Pages

```
<HTML>
<HEAD><TITLE>Widget Purchase Page
</TITLE></HEAD>
<BODY>
<P>
Thank you for your order of
<%= request.getParameter("quantity") %>
widgets!
</P>
</BODY>
</HTML>
```

5

JSP Syntax

- Basic tag form: `<% ... %>`
- Simplest form:
`<%= some Java expression %>`
 - Tomcat evaluates expression to get value
 - Inserts that value in place of expression in generated html page

6

JSP Simple Example

- Simple example:

```
<html>
<body>
<p>
Two plus two is <%= 2 + 2 %>.
</p>
</body>
</html>
```

Java
Server
Page

```
<html>
<body>
<p>
Two plus two is 4.
</p>
</body>
</html>
```

2 + 2 evaluated to value of 4

Resulting html Page

7

JSP Syntax

- Basic tag form: `<% ... %>`
- Executes code inside brackets without generating html
 - Set variables later used to generate html later
 - Store/access values in session/databases

```
<html>
<body>
<% int sum = 2 + 2; %>
<p>
Two plus two is <%= sum %>.
</p>
</body>
</html>
```

8

JSP Syntax

```
<html>
<body>
<% int sum = 2 + 2; %>
<p>
Two plus two is <%= sum %>.
</p>
</body>
</html>
```

Stores value of 4 in sum variable

Value of 4 in sum used in this JSP

```
<html>
<body>
<p>
Two plus two is 4.
</p>
</body>
</html>
```

No html here

JSP Scoping

- Variable declared in a block of JSP on a page
- May be accessed by any other JSP on same page
- No access to variable from any other page (purpose of sessions)

```
<html>
<body>
<% int sum = 2 + 2; %>
<p>
Two plus two is <%= sum %>.
</p>
</body>
</html>
```

Html Forms

- The FORM tag

```
<form action="url of response page"
method="get or post">
...
</form>
```

Method by which form data is passed

Relative url if response page is part of same webapp

Get vs. Post Method

<u>Get</u> method	<u>Post</u> method
Default method	Must specify in FORM
Appends form data to end of URL	Form data not visible (some security against shoulder surfing, but still need encryption for total security)
Faster	
No fixed limit to length of form data passed	Parameters limited to ~4k of data
Allows bookmarking of pages	Prevents bookmarking (good if part of session that user should no be able to enter in middle)

Simple Form Elements

- TEXT tag

```
<input type="text" name="elementname"/>
```

Necessary for value to be sent to server

- SUBMIT tag

```
<input type="submit" value="buttonlabel"/>
```

Place Order

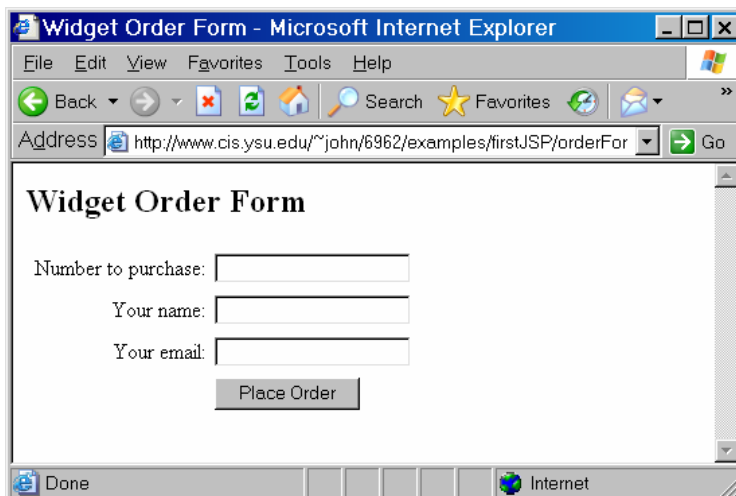
13

Form Element Example

```
<form action="orderReply.jsp" method="get">
  <table cellpadding="5">
    <tr>
      <td align="right">Number to purchase:</td>
      <td><input type="text" name="quantity"></td>
    </tr><tr>
      <td align="right">Your name:</td>
      <td><input type="text" name="customerName"></td>
    </tr><tr>
      <td align="right">Your email:</td>
      <td><input type="text" name="customerEmail"></td>
    </tr><tr>
      <td></td>
      <td><input type="submit" value="Place Order"></td>
    </tr>
  </table>
</form>
```

14

Form Element Example



Widget Order Form

Number to purchase:

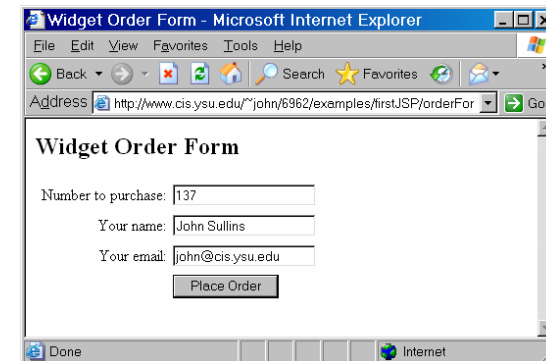
Your name:

Your email:

Place Order

15

Form Parameter Passing



Widget Order Form

Number to purchase:

Your name:

Your email:

Place Order

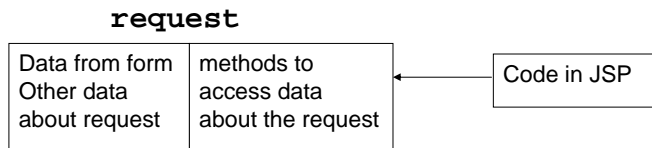
Parameter string passed:

quantity=137&customerName=John+Sullins&
customerEmail=john@cis.yosu.edu

16

Handling Form Data

- request object in JSP
 - Java object created from request string
 - Contains request data and methods to easily access that data
 - Accessed by JSP code



17

Handling Form Data

- Most useful method:
String request.getParameter(String)
 - Returns the corresponding value passed to the server
 - Takes name of form element as parameter
- Example:
String name =
`request.getParameter("customerName");`
sets the value of "name" to "John Sullins"

18

Example JSP

```
<body>
<%
String name = request.getParameter("customerName");
String email = request.getParameter("customerEmail");
String quantity = request.getParameter("quantity");
%>
<h2>Order Confirmation</h2>
<p>
Thank you for your order of <%= quantity %> widgets, <%= name %>.
</p>
<p>
You will shortly receive an email confirmation at <%= email %>.
</p>
</body>
```

19

Acquiring Form Data

- Statements to get and store form data:

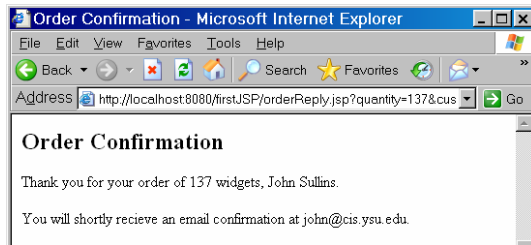
```
<%
String name =
request.getParameter("customerName");
String email =
request.getParameter("customerEmail");
String quantity =
request.getParameter("quantity");
%>
```

name	"John Sullins"
email	"john@cis.ysu.edu"
quantity	"137"

20

Displaying Values in Response

```
<p>
  Thank you for your order of <%= quantity %>
  widgets, <%= name %>.
</p><p>
  You will shortly receive an email
  confirmation at <%= email %>.
</p>
```



21

Commenting JSP Files

- Crucial to future maintenance of site
- Inside of JSP code (between <% and %>):
// comment
/* comment */
- Outside of JSP code (that is, in html)
<!-- comment -->

22

Importing Library Classes

- Much of Java classes in separate libraries
- Must be imported to be used in JSP
- Syntax:
<%@ page import="*list of Java classes*" %>
- Example:
<%@ page import="java.util.Date, ← Date class
 java.io.*" %> in the util
 library
All classes in the io library

23