

# Session 13

## JSP/Servlet Interaction

1

## Reading & Reference

- Reading
  - Head First (pages 205-207)

© Robert Kelly, 2001-2012

2

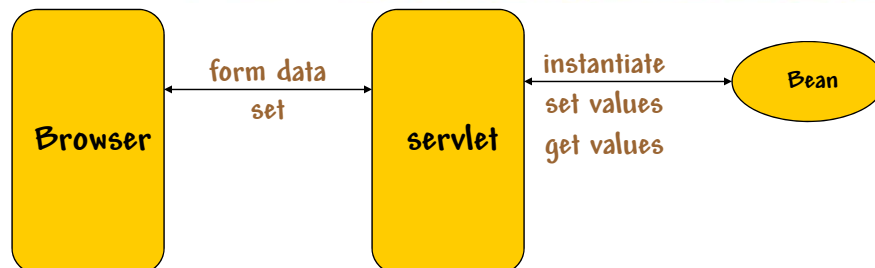
## Lecture Objectives

- Understand the differences between a server side forward and a redirect
- Understand the differences between an include and a forward - and when each should be used

© Robert Kelly, 2001-2012

3

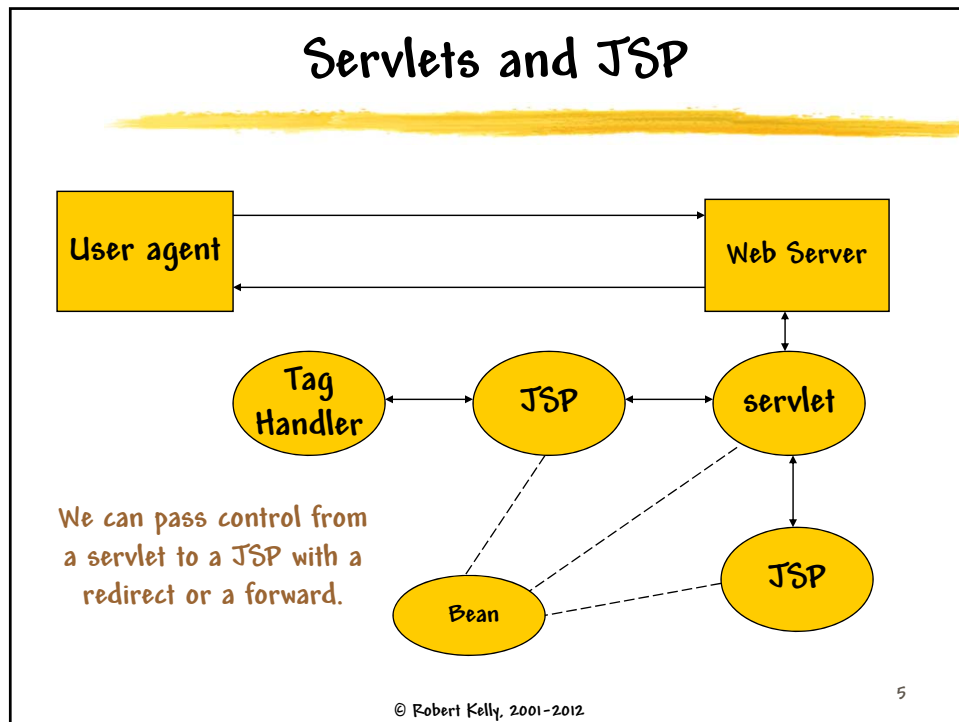
## Form Data and Beans



- A Web module (e.g., servlet) will usually read the form data set and set the values of the form in a bean so that they can be used by other Web modules (i.e., JSPs and servlets)

© Robert Kelly, 2001-2012

4



## RequestDispatcher Interface

- An object that receives a request (and response) and sends these objects to a named resource (e.g., servlet, JSP file) on the server
- Operates entirely within the server

```
String url = "/presentations/presentation1.jsp";
RequestDispatcher dispatcher =
    request.getRequestDispatcher(url);
dispatcher.forward(request, response);
```

You can also get a handle from the ServletContext

the forward method transfers control

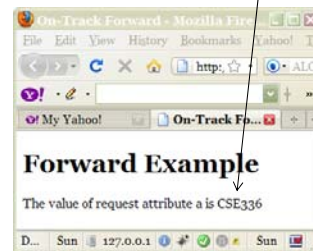
/ indicates the path is relative to the root of the Web application. Otherwise relative to the original request

© Robert Kelly, 2001-2012

## Are We on Track?

- Using your laptop, code a servlet that
  - Adds an attribute to the request object
  - Forwards to a JSP that displays (using EL) the value of the request attribute (in a paragraph tag)

RequestDispatcher is in the  
javax.servlet package



© Robert Kelly, 2001-2012

7

## Were We on Track?

### Servlet

```
...
request.setAttribute("a", "CSE336");
RequestDispatcher r =
    request.getRequestDispatcher("JSPs/TrackForward.jsp");
r.forward(request, response);
```

### JSP

```
<h1>Forward Example</h1>
<p>The value of request attribute a is ${requestScope.a}</p>
```

© Robert Kelly, 2001-2012

8

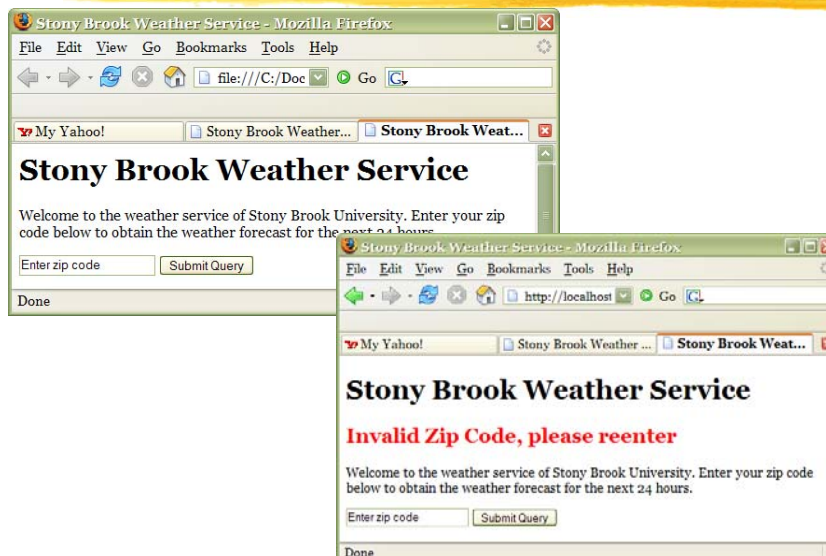
## Example - SBU Weather Service

- In assignment 4, you are working on a servlet solution that uses a redirect
- But, when possible, server-side forwards are better **Why?**
- In this example, we show an example that uses a controller servlet that contains a dispatcher.
- Dispatcher will check for a valid zip code and
  - Forward to the input form JSP (if zip is not valid)
  - Redirect to the weather display JSP (if valid)

© Robert Kelly, 2001-2012

9

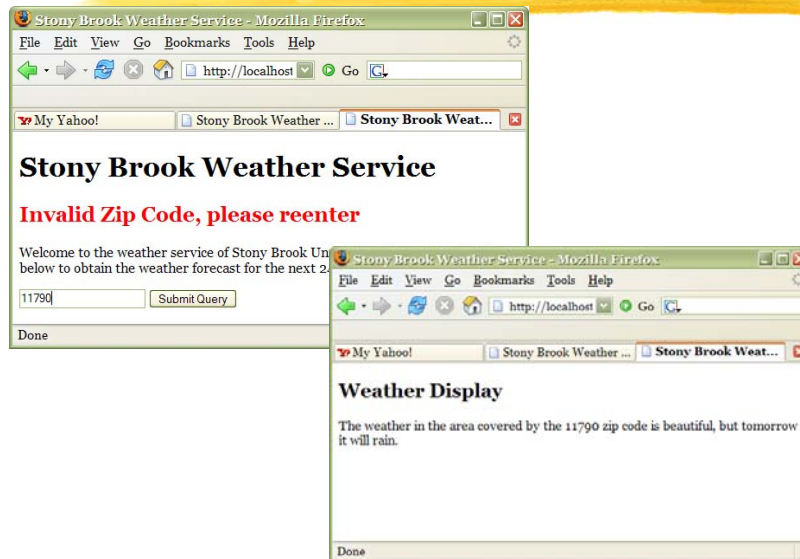
## SBU Weather Service ...



© Robert Kelly, 2001-2012

10

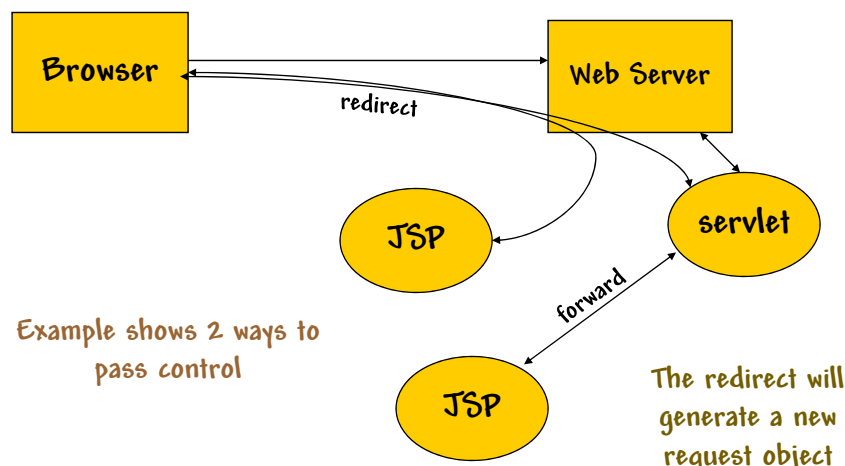
## ...SBU Weather Service



© Robert Kelly, 2001-2012

11

## SBU Weather Flow



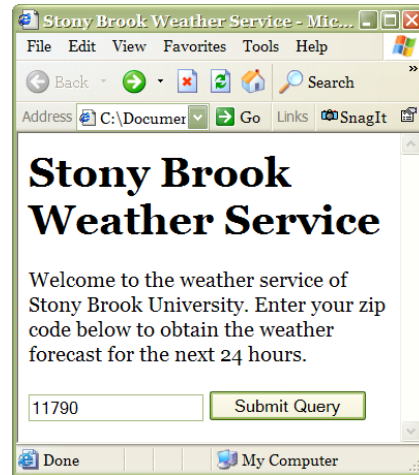
© Robert Kelly, 2001-2012

12

## SBUWeatherService.jsp

```
...
<body>
<h1>Stony Brook Weather Service</h1>
<p>Welcome to the weather service of
Stony Brook University. Enter
your zip code below to obtain
the weather forecast for the
next 24 hours.</p>
<form action="http://localhost:8080/
CodeCSE336/localweather.xyz"
method="get">
<input value="Enter zip code"
name="zip">
<input type="submit" value="Submit
Query">
</form>
</body>
</html>
```

Submit invokes  
the dispatcher  
servlet



© Robert Kelly, 2001-2012

13

## SBUWeatherDispatcher Servlet

localweather.xyz maps to the  
SBUWeatherController.java servlet

Note that we enter the  
error message in a  
shared scope

```
String zip = request.getParameter("zip");
if (zip == null | zip.equals("Enter zip code") | zip.equals(""))
{
    request.setAttribute("errorMessage",
        "Invalid Zip Code, please reenter");
    RequestDispatcher dispatcher =
        getServletContext().getRequestDispatcher(
            "/SBUWeatherService.jsp");
    dispatcher.forward(request, response);
}
response.sendRedirect(
    "http://localhost:8080/CodeCSE336/JSPs/SBUWeatherDisplay.jsp
    ?zip=" + zip);
```

Otherwise, we pass the  
zip in the query string

© Robert Kelly, 2001-2012

14

## SBUWeatherService.jsp

```
<head>
  <title>Stony Brook Weather Service</title>
  <meta http-equiv=Content-Type content="text/html;
  charset=iso-8859-1">
  <style type="text/css">
    .error {color:red;}
  </style>
</head>
<body>
  <h1>Stony Brook Weather Service</h1>
  <h2 class="error">${errorMessage}</h2>
  <p>Welcome to the weather service of Stony Brook
  University. Enter your zip code
  below to obtain the weather forecast for the next 24
  hours.</p>
  <form
  action="http://localhost:8080/CodeCSE336/localweather.xyz"
  method="get">
    <input value="Enter zip code" name="zip">
    <input type="submit" value="Submit Query">
  </form>
</body>
```

errorMessage is in the  
request object

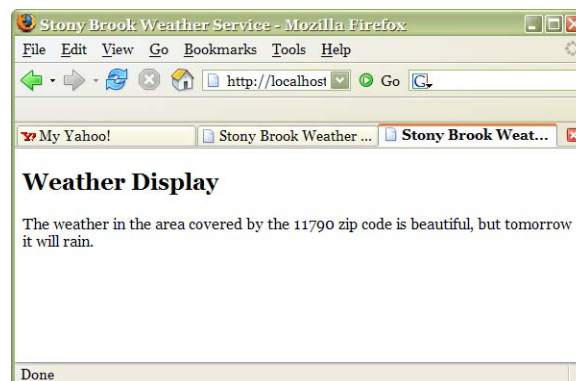
© Robert Kelly, 2001-2012

15

## SBUWeatherDisplay.jsp

```
<body>
<h2>Weather Display</h2>

<p>The weather in the area covered by the ${param.zip} zip
  code is beautiful, but tomorrow it will rain.
</p>
</body>
```



© Robert Kelly, 2001-2012

16



## Supplying Info to the Destination

- When you forward a request from your servlet you can:
  - 1. Forward all the information in the request object (e.g., attributes), and have the receiver (JSP or servlet) process the information
  - or
  - 2. Your servlet can process the request and store the information in a bean - so a subsequent request can access the bean

The second solution is more in line with the MVC architecture

© Robert Kelly, 2001-2012

17

## Forward / Include

- RequestDispatcher provides 2 methods
  - forward
    - | Forwards a request to another resource on the server
    - | The destination resource generates the response
    - | Called before response buffer is flushed
  - include
    - | Includes the response of the target in the response generated by the servlet using the dispatcher

You will not likely use the include method in your Sun form project - it is mainly for library tags

© Robert Kelly, 2001-2012

18

## Forward / Include Issues

- Use the include method of RequestDispatcher to have the servlet provide response body
- Included page:
  - Must be dynamic
  - Cannot set the status code
  - Cannot set headers
  - Must use the flush attribute ( and set it to true)
- JSP provides support for includes and forwards

`<jsp:include page="pathName" flush="true" />` occurs at request time

*c:import is more powerful, so you may not need to use jsp:include*

© Robert Kelly, 2001-2012

19

## Example

**servlet**

*The servlet forwards to next.jsp*

```
...
InfoBean b = new InfoBean();
b.setValue(request.getParameter("value"));
request.setAttribute("myBean", b);
RequestDispatcher r = request.getDispatcher("next.jsp");
r.forward(request, response);
}
```

*A request attribute is not the same as a request parameter*

**next.jsp**

```
...
<jsp:useBean scope="request" id="myBean" class="InfoBean" />
${myBean.value}
```

© Robert Kelly, 2001-2012

20

## Have You Satisfied the Lecture Objectives?

- Understand the differences between a server side forward and a redirect
- Understand the differences between an include and a forward - and in which cases each would be used

© Robert Kelly, 2001-2012

21

## Assignment 4c

- Extend your solution in the Form Project (Assignment 4a) so that you
  - Modify your controller servlet so that you forward to the appropriate JSP (instead of redirect)
  - Enhance the congratulations page with an include of the form page header and footer

(which include?)

© Robert Kelly, 2001-2012

22