Session 22

Intra Server Control

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
We can pass control from a servlet to a JSP with a redirect or a forward.

**Servlets and JSP**

A redirect is an HTTP status code that causes the browser to load a different page.

```java
response.sendRedirect("results-page.jsp");
```

**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

```java
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
Forward / Include

RequestDispatcher provides 2 methods

1. forward
   - Forwards a request to another resource on the server
   - The destination resource generates the response
   - Called before response buffer is flushed

2. 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 project – it is mainly for library tags

Forward / Include Issues

1. Use the include method of RequestDispatcher to have the servlet provide response body

2. Included page:
   - Must be dynamic
   - Cannot set the status code
   - Cannot set headers
   - Must use the flush attribute (and set it to true)

3. 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, 2016-2018
Example

The servlet forwards to next.jsp

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

A request attribute is not the same as a request parameter

```
<p>The value that was forwarded is \${myBean.value}</p>
```

... Example

Bean used in the example:

```
package lectures;

public class InfoBean {

    private String value;

    public void setValue(String s) {
        value = s;
    }

    public String getValue() {
        return value;
    }
}
```
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