

Objectives

- Servlets Review
- JSPs
- Web Application Organization

Servlets Review

- What is the servlet life cycle?
- Init parameters
 - Where are init parameters defined?
 - How do we access a servlet's init parameter?
 - Why do we use init parameters?
- How can we save state across multiple requests from a user?
 - What are the pros and cons of each?
- Since multiple users can access a servlet at the same time, what problem should we be concerned about?
 - What is the cause of the issue?
 - How do we solve it (theoretically; we haven't shown how to implement yet)?

Review: Servlet Life Cycle



Web Application Server

- Web application server creates **one** instance of servlet
 - Calls `init` method of servlet created
- As requests come in, WAS calls `service` method of appropriate servlet
 - In turn, servlet calls appropriate `doMethod`
- When web application server shuts down, calls `destroy` method of each servlet

JAVASERVER PAGES (JSPS)

Discussion

What made writing servlets difficult?

Motivation: JavaServer Pages (JSPs)

- Simplify web application development
- Separate UI from backend code
 - Separate presentation layer
- Difficult to write HTML in print statements

Moving to here



JavaServer Pages (JSPs)

- Merge HTML and Java
 - Separate static HTML from dynamic
 - Make HTML templates, fill in dynamic content
 - Encourages separation of tasks
- Web application server compiles JSPs into servlet code
 - Clean and efficient
- Easier to develop, deploy, modify scripted pages
 - How much trouble did you have with HTML in Strings?

Apr 30, 2019

Sprenkle - CS335

7

JSP Syntax: Expression

- Enclose code in `<%= %>`

```
<html>
<body>
<p>
Hello! The time is now <%= new java.util.Date() %>
</p>
</body>
</html>
```

Expression
Evaluated, turned
into a String

Apr 30, 2019

Sprenkle - CS335

8

JSP Syntax: Scriptlet

```
<html>
<body>
<%
    // This is a scriptlet. The "date" variable
    // we declare here is available in the
    // embedded expression later on.
    java.util.Date date = new java.util.Date();
%>
<p>
Hello! The time is now <%= date %>
</p>
</body>
</html>
```

Apr 30, 2019

Sprenkle - CS335

9

Example: SurveyServlet as a JSP

```
<%
for (int i = 0; i < animalNames.length; i++) {
%>
<tr>
    <td><%=animalNames[i]%></td>
    <td><%=votes[i]%></td>
    <td><%=formattedPercentages[i]%></td>
    <%
        totalVotes += votes[i];
    %>
</tr>
<%
}
%>
```

To be displayed at end

Apr 30, 2019

Sprenkle - CS335

10

JSP Directives

- Page Directive

- Java files to import (like import statement in Java)

```
<%@ page import="java.util.*,java.text.*" %>
```

```
<%@ page import="ourcode.MyClass"%>
```

- Include Directive

- Include contents of another file: JSP, HTML, or text

- Example: include site's common headers or footers

```
<%@ include file="header.jsp" %>
```

JSP Variables

- By default, JSPs have some variables

- **Not explicitly declared in the file**

- HttpServletRequest **request**

- HttpServletResponse **response**

- HttpSession **session**

These names
must be used

- JSPs can access request parameters, session data

JSP Declarations

- For instance variables and methods

```
<%!  
    private ArrayList users;  
  
    public void jspInit() {  
        // on start up: set up  
    }  
    public void jspDestroy() {  
        // on shut down: clean up  
    }  
%>
```

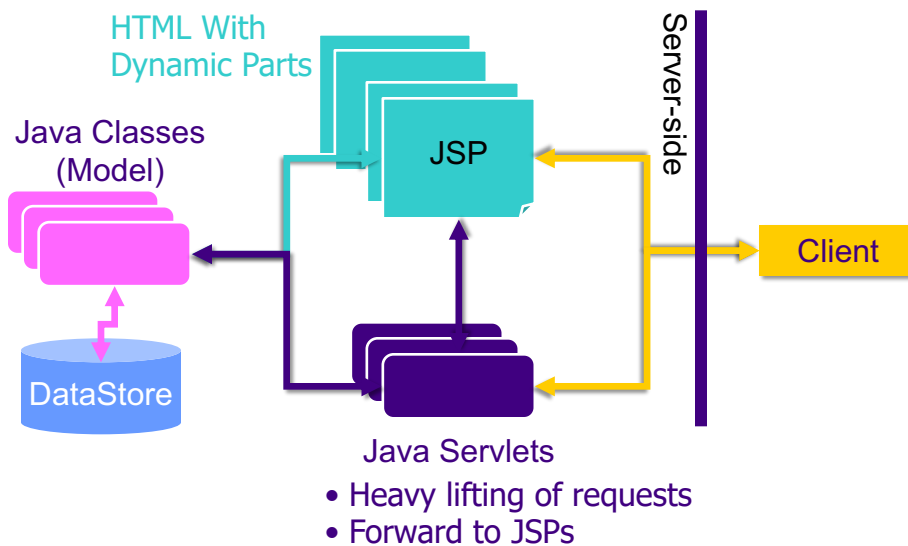
- We won't do too much of this
 - Let servlets do the work

Apr 30, 2019

Sprenkle - CS335

13

Web Application Architecture

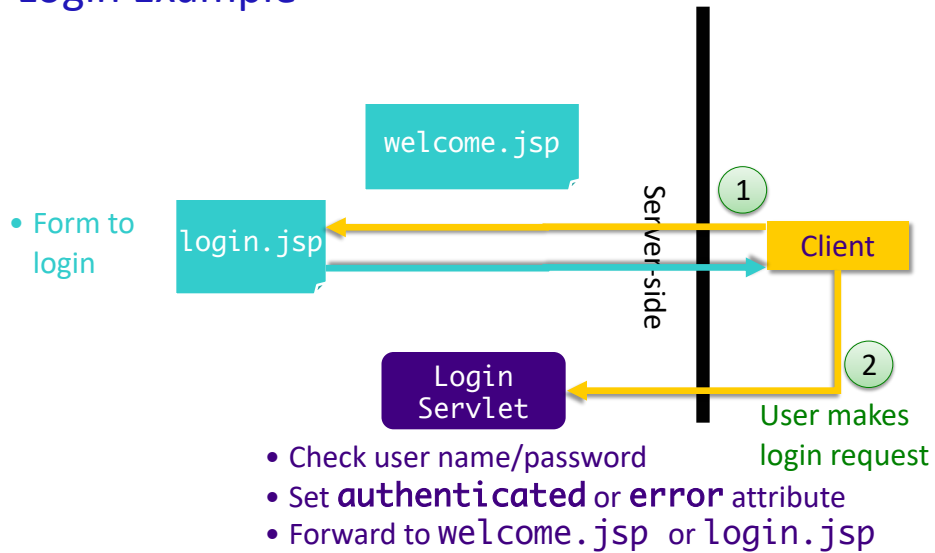


Apr 30, 2019

Sprenkle - CS335

14

Communicating Between Servlets and JSPs: Login Example

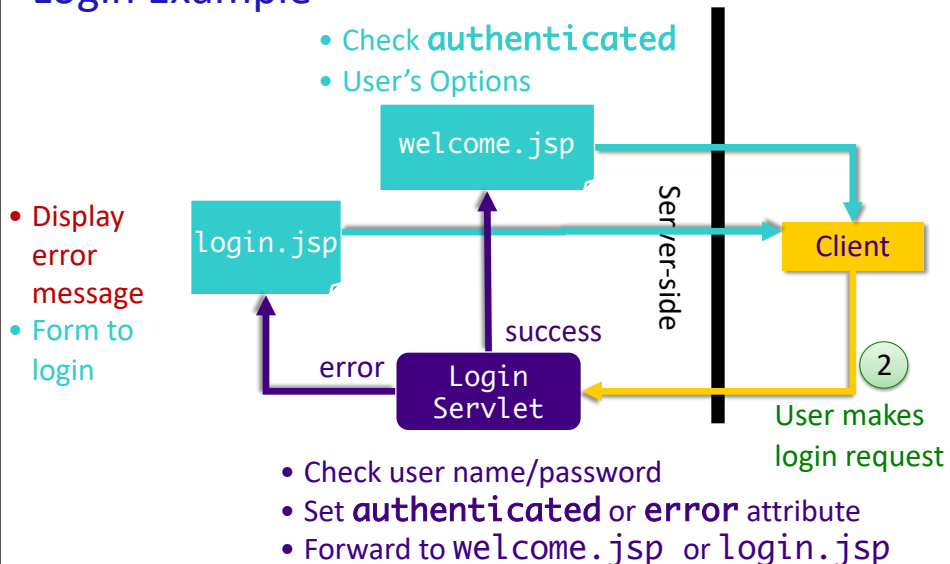


Apr 30, 2019

Sprenkle - CS335

15

Communicating Between Servlets and JSPs: Login Example



Apr 30, 2019

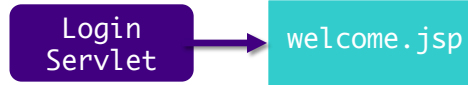
Sprenkle - CS335

16

Communicating Between JSPs and Servlets

- **Attributes**

- Name/Value pairs
- Values are **Objects**
- Can get/set attributes on the **HttpServletRequest** object
 - Similar to *session* attributes but call methods on a *request* object
 - Different lifetime



- **Parameters**

- Names and Values are **Strings**
- From forms or in URLs
- Also in the **HttpServletRequest** object

Apr 30, 2019

Sprenkle - CS335

17

Forwarding Requests from Servlet

- **HttpServletRequest's** **getRequestDispatcher** method

- Returns a **RequestDispatcher** object

The name of the resource
to forward to



```
request.getRequestDispatcher("welcome.jsp").  
forward(request, response);
```

- Can use **RequestDispatcher's** **include** method similarly

Apr 30, 2019

Sprenkle - CS335

18

Adding a JSP to SurveyServlet

- Separate heavy lifting from the HTML
- Think of JSP as a template
 - What is static about the response page?
 - What is dynamic?
- Servlet will handle most of the work

Look at code

Protecting JSPs

- If there are JSPs that you don't want users to be able to access directly by typing in the URL, put them in the WEB-INF directory
 - Web application server blocks access to the JSP
 - Don't need code to check authorization again
 - Only get to JSP through a servlet that checks authorization
- Forward requests from a servlet to the JSP by including WEB-INF in the URI


Using the WEB-INF Directory

- Example: User shouldn't be able to access petResponse.jsp directly

Trick: Ternary Operator

- Alternative `if-then-else` syntax
- Returns a value
- Example of assigning a variable the minimum:

Condition



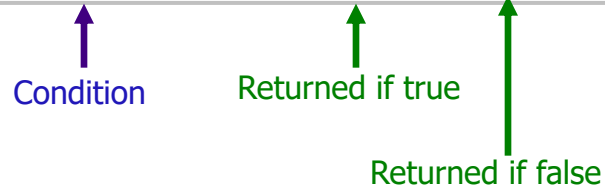
```
minVal = (a < b ? a : b);
```

- Assign `minVal` value `a` if condition is `true`, `b` if condition is `false`

Ternary Operator in JSP

- Allows for more concise code

```
<input type=text name="username"  
value="<%= userName != null ? userName : ""%>">
```



Apr 30, 2019

Sprenkle - CS335

23

HttpServletRequest

- `getContextPath()`
 - Returns the portion of the request URI that indicates the context of the request.
- Example with various Request methods

```
http://example.com:8080/app/dirpath/index.jsp?cat=2&cat=5
```

```
getScheme() → "http"  
getServerName() → "example.com"  
getServerPort() → 8080
```

```
getContextPath() → "/app"  
getServletPath() → "/dirpath"  
getPathInfo() → "/index.jsp"  
getParameter("cat") → "2"  
getParameterValues("cat") → {"2", "5"}
```

Apr 30, 2019

Sprenkle - CS335

24

Use in JSP

```
<a href="<%=request.getContextPath()%>">  
Main Page</a>
```