CS520 Web Programming
Servlet and JSP Review

Chengyu Sun
California State University, Los Angeles

Review Examples

1. Add two numbers
2. Guest Book
3. Login and logout
4. Email
5. Pizza Order

Example 1: Add Two Numbers

- Take two integer numbers as request parameters and display the sum

Topics Reviewed in Example 1

- Web project
- Servlet
  - `@WebServlet`
- Request parameter
- HTML form
  - `GET` and `POST`
- Deployment

Create a Web Project

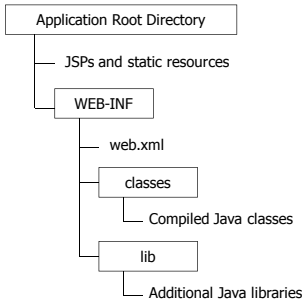
- Eclipse
  - http://csns.calstatela.edu/wiki/content/cysun/course_materials/cs520/development
  - Dynamic Web Project
  - `web.xml`

Versions, Versions

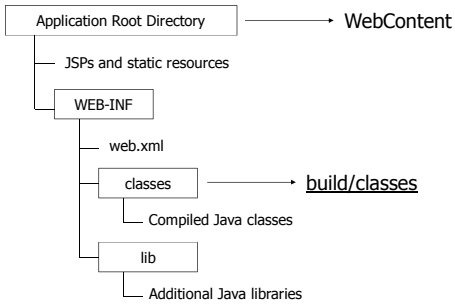
Servlet/JSP Spec	Tomcat	Java
3.0/2.2	7.0.x	1.6
2.5/2.1	6.0.x	1.5
2.4/2.0	5.5.x	1.4

The `version` attribute of `<web-app>` in web.xml

Directory Structure of a Java Web Application



Directory Structure of an Eclipse Dynamic Web Project



@WebServlet

- http://download.oracle.com/javaee/6/api/javax/servlet/annotation/WebServlet.html

@WebServlet Elements for URL Patterns

- `value`
  - URL pattern(s) of the servlet
  - The default element
- `urlPatterns`
  - Same purpose as `value`
  - Usually used when more than one element is specified
  - Only one of `value` and `urlPatterns` can be specified

@WebServlet Examples

```
@WebServlet( "/HelloServlet" )

@WebServlet( {"/HelloServlet", "/member/*"} )

@WebServlet( name="Hello", urlPatterns={"/HelloServlet", "/*.html"} )

@WebServlet(
        urlPatterns="/MyPattern",
        initParams={@WebInitParam(name="ccc", value="333")}
)
```

Wildcard in Servlet Mapping

- A string beginning with a **/** and ending with a **/***
  - E.g. `/*`, `/content/*`
- A string beginning with a ***.**
  - E.g. `*.html`, `*.do`

See Servlet Specification 3.0, Section 12

HTTP Request Example

http://cs3.calstatela.edu:8080/whatever

GET /whatever HTTP/1.1
Host: cs3.calstatela.edu:4040
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.3) ...
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,...
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: nxt/gateway.dll/uid=4B4CF072; SITESERVER=ID=f1675...

HTTP Request

- Request line
  - Method
  - Request URI
  - Protocol
- Header
- [Message body]

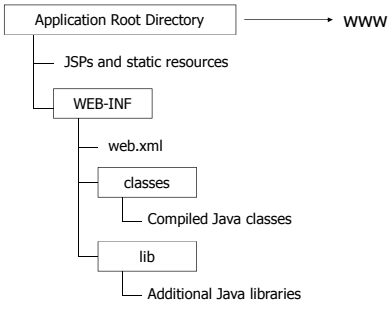
Request Methods

- Actions to be performed regarding the resource identified by the *Request URI*
- Browser
  - GET
  - POST
- Editor
  - PUT
  - DELETE
- Diagnosis
  - HEAD
  - OPTIONS
  - TRACE

Deploy a Web Project on CS3

- Understand directory structure
- "touch" or re-upload `web.xml` to force Tomcat to reload your application

Directory Structure on CS3



Example 2: Guest Book

My Guest Book

John says:	Hello!	Edit | Delete
Jane says:	Your website looks nice.	Edit | Delete
Joe says:	Nice to meet you. I'm from China.	Edit | Delete

Add Comment

My Guest Book – Add Comment

Your name:

Submit

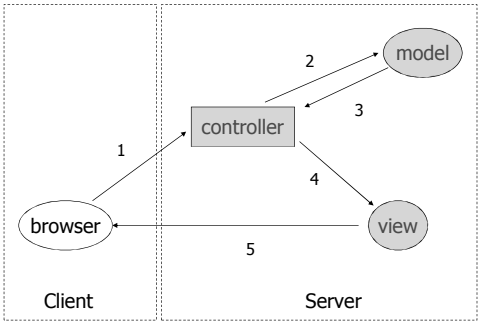
Topics Reviewed in Example 2

- MVC
- Scopes
- Expression Language
- JSTL

Web Application

- Web application = Data + Operations
- Data
  - Guestbook entries, blog entries, forum posts, wiki pages, twitter message …
- Operations
  - Add/create, search, display, edit, delete …

MVC Architecture



About MVC

- Models represent data in the application
- Controllers implement the actions
  - Handle user input
  - Access and process data
  - Implement business logic
  - Pass data to views for display
- Views render the display to the users

MVC Using Servlet and JSP

- Model: Bean (a.k.a. POJO)
- Controller: Servlet
- View: JSP
  - HTML, CSS, JavaScript
  - Expression Language (EL)
  - Custom tags (e.g. JSTL)
  - *No scripting elements*
    - <%! %>
    - <%= %>
    - <% %>

Understand Bean Properties

- Bean properties are defined by getters and/or setters
  - E.g. `getFoo()` ➔ `foo`
  - Read-only: only getter
  - Write-only: only setter
  - Read-write: both getter and setter
- For a boolean property, the getter starts with `is` instead of `get`
  - E.g. `isFoo()` instead of `getFoo()`

Scopes and Data Sharing

- Application scope – data is valid throughout the life cycle of the web application
- Session scope – data is valid throughout the session
  - redirect, multiple separate requests
- Request scope – data is valid throughout the processing of the request
  - forward
- Page scope – data is valid within current page

Common Usage of Scopes

- Application scope
  - Store data shared by all users
- Session scope
  - Store data associated with a session, e.g. login credentials, shopping cart
- Request scope
  - Pass data from controller to view
- Page scope
  - Local variables in a JSP

Access Scoped Variables in Servlet

- Application scope
  - `ServletContext`
- Session scope
  - `HttpSession`
- Request scope
  - `HttpServletRequest`
- Page scope (in JSP scriptlet)
  - `pageContext`

Expression Language

- Expression Language (EL)
  - A JSP 2.0 standard feature
  - A more concise way to write JSP expressions
    - vs. <%= expression %>
  - Java's answer to scripting languages
- EL Syntax

${ *expression* **}**

Expression

- Literals
- Operators
- Variables
- Functions
  - see Custom Tag Libraries

EL Operators

- Arithmetic
  - +, -, *, /, %
  - div, mod
- Logical
  - &&, ||, !
  - and, or, not
- Relational
  - ==, !=, <, >, <=, >=
  - eq, ne, lt, gt, le, ge
- Conditional
  - ? :
- empty
  - check whether a value is null or empty
- Other
  - [], ., ()

Implicit Objects

- pageContext
  - servletContext
  - session
  - request
  - response
- param, paramValues
- header,headerValues
- cookie
- initParam
- pageScope
- requestScope
- sessionScope
- applicationScope

Common Usage of EL

- Access scoped variables
  - `${applicationScope.foo}`
  - `${sessionScope.foo}` `${requestScope.foo}`
  - `${pageScope.foo}`
  - `${foo}`
- Access object properties, e.g. `${foo.bar}`
- Simple operations, e.g. `${not empty param.foo}`

JSP Standard Tag Library (JSTL)

Library	URI	Prefix
Core	http://java.sun.com/jsp/jstl/core	c
XML Processing	http://java.sun.com/jsp/jstl/xml	x
I18N Formatting	http://java.sun.com/jsp/jstl/fmt	fmt
Database Access	http://java.sun.com/jsp/jstl/sql	sql
Functions	http://java.sun.com/jsp/jstl/functions	fn

http://csns.calstatela.edu/file/view?id=4514026

JSTL Core

- Flow control
  - <c:if>
  - <c:choose>
    - <c:when>
    - <c:otherwise>
  - <c:forEach>
  - <c:forToken>
- Variable support
  - <c:set>
  - <c:remove>
- URL
  - <c:param>
  - <c:redirect>
  - <c:import>
  - <c:url>
- Output
  - <c:out>
- Exception handling
  - <c:catch>

Format Date and Time

```
<fmt:formatDate value="${date}" type="date" />

<fmt:formatDate value="${date}" type="time" />

<fmt:formatDate value="${date}" type="both" />

<fmt:formatDate value="${date}"
        pattern="yyyy-MM-dd hh:mm:ss a" />
```

See http://download.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html for the date formatting patterns.

JSTL Functions

- fn:length()
- fn:contains()
- fn:containsIgnoreCase()
- fn:startWith()
- fn:endsWith()
- fn:indexOf()
- fn:replace()
- fn:trim()
- fn:toUpperCase()
- fn:toLowerCase()
- fn:substring()
- fn:substringAfter()
- fn:substringBefore()
- fn:split()
- fn:join()
- fn:escapeXML()

Example 3: Login and Logout

- A user must login before he or she can add comments to the guest book.

My Guest Book

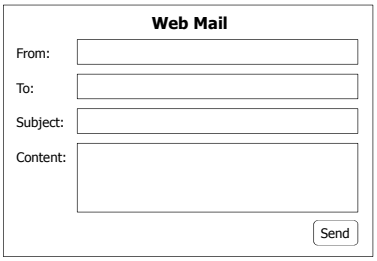
John says:	Hello!	Edit
Jane says:	Your website looks nice.	Edit
Joe says:	Nice to meet you. I'm from China.	Edit

Add Comment Logout

Topics Reviewed in Example 3

- Session tracking
- Basic login/logout mechanism

Example 4: Email

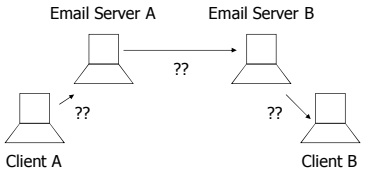


Topics Reviewed in Example 4

- Email basics
- Use of `JavaMail`

How Email Works

- SMTP, IMAP, POP



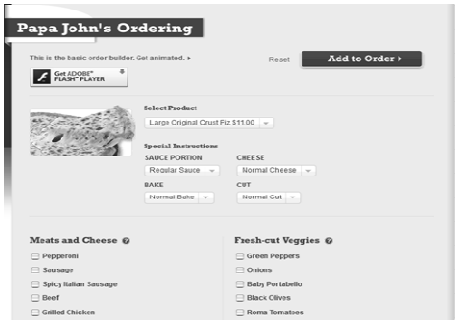
JavaMail

- http://java.sun.com/products/javamail/

```
Properties props = System.getProperties();
props.put("mail.smtp.host", mailhost);
Session session = Session.getInstance( props );

Message msg = new MimeMessage(session);
...
Transport.send( msg );
```

Example 5: Pizza Order



Create Your Own Pizza

- Crusts
  - Large Original, $11
  - Medium Original, $9
  - Large Thin, $11
- Cheese
  - Normal cheese, no cheese (-$1)
- Toppings ($1 each)
  - Pepperoni, sausage, bacon, pineapple

UI – Customize Pizza

Crust: Large Original $11

Cheese: Normal ● No cheese ○

Toppings: Pepperoni ☐

Sausage ☐

Bacon ☐

Pineapple ☐

Add to Order

UI – Review Order

Pizza	Quantity	Price
Large Original Crust, Normal Cheese, with Pepperoni, Bacon	1	$13
Large Thin Crust, No Cheese, with Pineapple	2	$22
	Total	$35

Add Another Pizza | Update Order | Place Order

Summary