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

<table>
<thead>
<tr>
<th>Servlet/JSF Spec</th>
<th>Tomcat</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0/2.3</td>
<td>7.0.x</td>
<td>1.6</td>
</tr>
<tr>
<td>2.5/2.1</td>
<td>6.0.x</td>
<td>1.5</td>
</tr>
<tr>
<td>2.4/2.0</td>
<td>5.5.x</td>
<td>1.4</td>
</tr>
</tbody>
</table>

The version attribute of <web-app> in web.xml
Directory Structure of a Java Web Application

- Application Root Directory
  - WEB-INF
    - web.xml
    - classes
      - Compiled Java classes
    - lib
      - Additional Java libraries

Directory Structure of an Eclipse Dynamic Web Project

- Application Root Directory
  - WEB-INF
    - web.xml
    - classes
    - lib
    - Additional Java libraries
  - WebContent
    - build/classes

@WebServlet


@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" )
- @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-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
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

Application Root Directory → WWW
  ▪ JSPs and static resources
  ▪ WEB-INF
    ▪ web.xml
      ▪ classes
        ▪ Compiled Java classes
      ▪ lib
        ▪ Additional Java libraries

Example 2: Guest Book

My Guest Book

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

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

- 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

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`
- `sessionContext`
- `session`
- `request`
- `response`
- `param`, `paramValues`
- `header`, `headerValues`
- `cookie`
- `initParam`

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)

<table>
<thead>
<tr>
<th>Library</th>
<th>URI</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td><a href="http://java.sun.com/jsp/jstl/core">http://java.sun.com/jsp/jstl/core</a></td>
<td>c</td>
</tr>
<tr>
<td>XML Processing</td>
<td><a href="http://java.sun.com/jsp/jstl/xml">http://java.sun.com/jsp/jstl/xml</a></td>
<td>x</td>
</tr>
<tr>
<td>JSTL Core</td>
<td><a href="http://java.sun.com/jsp/jstl/fmt">http://java.sun.com/jsp/jstl/fmt</a></td>
<td>fmt</td>
</tr>
<tr>
<td>Database Access</td>
<td><a href="http://java.sun.com/jsp/jstl/sql">http://java.sun.com/jsp/jstl/sql</a></td>
<td>sql</td>
</tr>
<tr>
<td>Functions</td>
<td><a href="http://java.sun.com/jsp/jstl/functions">http://java.sun.com/jsp/jstl/functions</a></td>
<td>fn</td>
</tr>
</tbody>
</table>

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

JSTL Core

- Flow control
  - `<c:if>`
  - `<c:choose>`
  - `<c:when>`
  - `<c:otherwise>`
  - `<c:forEach>`
  - `<c:forTokens>`
- Variable support
  - `<c:set>`
  - `<c:remove>`

- URL
  - `<c:param>`
  - `<c:redirect>`
  - `<c:import>`
  - `<c:curl>`
- Output
  - `<c:out>`
- Exception handling
  - `<c:catch>`

JSTL Functions

- `<fn:length>()`
- `<fn:contains()>`
- `<fn:containsIgnoreCase()>`
- `<fn:startWith()>`
- `<fn:endsWith()>`
- `<fn:indexOf()>`
- `<fn:replace()>`
- `<fn:trim()>`
- `<fn:toLowerCase()>`
- `<fn:startsWith()>`
- `<fn:contains()>`
- `<fn:trim()>`
- `<fn:toLowerCase()>`
- `<fn:escapeXML()>`

See [http://download.oracle.com/javaee/6/docs/api/java/text/SimpleDateFormat.html](http://download.oracle.com/javaee/6/docs/api/java/text/SimpleDateFormat.html) for the date formatting patterns.
Example 3: Login and Logout

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

<table>
<thead>
<tr>
<th>My Guest Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>John says: Hello!</td>
</tr>
<tr>
<td>Jane says: Your website looks nice.</td>
</tr>
<tr>
<td>Joe says: Nice to meet you. I'm from China.</td>
</tr>
</tbody>
</table>

Add Comment  Logout

Topics Reviewed in Example 3

- Session tracking
- Basic login/logout mechanism

Example 4: Email

Web Mail

From:
To:
Subject:
Content:
Send

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

<table>
<thead>
<tr>
<th>Pizza</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large Original Crust, Normal Cheese, with Pepperoni, Bacon</td>
<td>1</td>
<td>$13</td>
</tr>
<tr>
<td>Large Thin Crust, No Cheese, with Pineapple</td>
<td>2</td>
<td>$22</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>$35</strong></td>
</tr>
</tbody>
</table>

Add Another Pizza  Update Order  Place Order

Summary

Server-side Programming

ASP, PHP  Servlet  ...

Exploring Technologies

- JSP
- Scripting Elements
- Bean
  - (Data Modeling)
- Tag Library
  - (Property Access)
- (Display Logic)

Filter web.xml  Java Web Application  Static content

Other libraries