

# CS 416

## Web Programming

### Java Server Pages

Dr. Williams  
Central Connecticut State University

# Java Server Pages

- **Java Server Pages (JSPs)**
- Servlets are good for handling form data but they get messy if there is a lot of HTML
- JSPs use a combination of static HTML and dynamic content to render web pages
- Result is JSPs are a lot easier to maintain than servlets with HTML

# Model-View-Controller

- Modern architecture typically follows the approach referred to as Model-View-Controller (MVC)
- The purpose of the MVC architecture is to separate the object model and control of the application from how it is displayed
  - Object model is Model component
  - JSPs are the View component
  - Servlets the Controller component

# What is a JSP

- A JSP page is a text-based document that contains two types of text
  - Static template data (HTML, XML, etc.)
  - JSP elements which construct dynamic content
- Essentially HTML code with snippets of Java code referred to as **scriptlets**, or by using standard or custom JSP tags
- Page looks like a standard HTML page except there are additional sections to render dynamic content that are surrounded by “<%” and “%>”

# Simple JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.Date" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <p>Server date and time: <% out.println(new Date()); %>
  </body>
</html>
```

Page  
directives

Scriptlet

# Heading of JSP page

- The beginning of a JSP page contain the web page's JSP **page directives**
- Page directives define attributes that apply to the entire JSP page
- Indicate properties of page, type of page, and what the page has access to
- Directives can appear on single line or more than one line

# Directives

- Complete list of directives online, but the most commonly used are:
  - **import** – classes imported by this page
  - **extends** – indicates the class which this page extends (must be a subclass of `javax.servlet.GenericServlet` all JSPs extend this class)
  - **errorPage** – Indicates which page to navigate to when this page throws an Exception
  - **isErrorPage** – Determines if this page is an error page, by being marked as an error page the page will have access to the *exception* object

# JSP implicit objects

- JSP implicit objects are objects that can be used in the JSP without having to be declared or initialized
- These objects are declared and initialized behind the scenes to provide quick access to the most commonly used functionality
- For example writing to the page, reading request, session, and context variables



# Implicit objects

Implicit object	Object class	Description
application	<code>javax.servlet.ServletContext</code>	Equivalent to calling the <code>getServletContext()</code> method
config	<code>javax.servlet.ServletConfig</code>	Equivalent to calling <code>getServletConfig()</code>
exception	<code>java.lang.Throwable</code>	Only accessible if page directive <code>isErrorPage</code> is set to true
out	<code>javax.servlet.jsp.JspWriter</code>	Equivalent to calling <code>HttpServletResponse.getWriter()</code>
page	<code>java.lang.Object</code>	Provides access to the page's generated servlet

# Implicit objects

Implicit object	Object class	Description
pageContext	javax.servlet.jsp.PageContext	Provides several methods for managing various web application scopes (request, session, application)
request	javax.servlet.HttpServletRequest	Equivalent to the instance of HttpServletRequest passed into doGet() / doPost()
response	javax.servlet.HttpServletResponse	Equivalent to the instance of HttpServletResponse passed into doGet() / doPost()
session	javax.servlet.http.HttpSession	Equivalent to return value of HttpServletRequest.getSession() method

# Implicit objects

```
<tr>
  <td>session counter</td>
  <td>
    <%
      Integer sessionCounter =
        (Integer)session.getAttribute("counter");
      if (sessionCounter == null) {
        sessionCounter = new Integer(0);
      }
      sessionCounter += 1;
      session.setAttribute("counter", sessionCounter);
      out.println(sessionCounter);
    %>
  </td>
</tr>
```

# JSP Expressions

- Within JSP there is a short hand when you want to simply output a variable referred to as **JSP expressions**
- Syntax is `<%= variable %>`

`<%= session.getAttribute("myVar") %>`

Note no semi-colon  
at end of statement

# Errors and error pages

- It is a good idea to specify an error page for any page you develop
- This will ensure that any errors that occur can be gracefully handled by your application rather than displaying the server default page for when an exception occurs
- To specify an error page:

```
errorPage="MyErrorPage.jsp"
```

Must be declared in the page directive

# Error pages

- To declare that a page is an error page it must declare:

`isErrorPage="true"`

in the page directive

- An error page then has access to the thrown exception in the implicit `exception` object
- The error page can then handle the error by logging it to the server etc. and then display a user friendly error to the user

# JSPs and JavaBeans

- JSPs are designed to make it very easy to retrieve and store information in JavaBean objects
- A JavaBean is a class that must possess the following attributes:
  - Public constructor with no arguments
  - Variables must be accessed via getter and setter methods
  - Must implement `java.io.Serializable`
  - Good practice for variables to be private

## Bean ex.

```
public class Person implements
    Serializable{
    private String firstName;

    public Person() {
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String
firstName) {
        this.firstName = firstName;
    }
}
```



# Using Beans in a page

- Specify using a bean at the top of the page

```
<jsp:useBean id="myPerson"  
scope="page" class="edu.ccsu.Person"  
>
```

- id – how to refer to the bean throughout the rest of the page, can specify multiple beans of same class by specifying different ids
- Scope of the bean  
page/request/session/application
- class – the fully specified class name

# Accessing Bean properties

- Bean properties can be used through out the page

```
<jsp:getProperty name="beanName"  
property="beanProperty" />
```

- The name refers to the id specified at top of page
- Property refers to any attribute with a getter and setter on the bean

# Beans in JSPs

```
<%@page contentType="text/html" pageEncoding="UTF-8"
    errorPage="e.jsp"%>
<jsp:useBean id="myPerson" scope="page"
    class="edu.ccsu.Person" />
<html>
    <head><title>Bean test</title></head>
    <body>
        First name:<input type="text" name="firstName"
        value="<jsp:getProperty name="myPerson"
            property="firstName" />" />
```

## Setting bean properties

- Setting bean properties can be done in two ways: setting the property directly

```
<jsp:setProperty name="beanName"  
property="beanProperty" value="val"/>
```

- Or be set based on request values:

```
<jsp:setProperty name="beanName"  
property="beanProperty"  
param="requestParam"/>
```

- Alternatively you can set **all** parameters on the bean that match parameters on the request

```
<jsp:setProperty name="beanName"  
property="*" />
```

# Reusing JSP content

- To prevent duplicating JSP content JSPs allow you to include the content from other pages

```
<%@ include  
    file="navigation.jspf"%>
```

- File is included at location the tag is entered
- Naming standard is “\*.jspxf” for Java Server Page **fragment**

# JSP demos

- Bean counter – page, session context
- Page context – accessing information available to page
- Page with error, error Page

# Creating a page

- Create a JSP and Java Bean
- The input form CDEntry take a CDs artist, title, and rating
- After submitting the form the next page should display both name and artist in a table. The values should be set and read from bean properties
- Create your own error page and link to it on error
- Add a context variable that keeps track of the total number of cds entered

## CD ratings

- Display the average rating of CDs for the current user on the next page (Using a bean or otherwise)



# Keeping track of CDs

- Keep track of all CDs entered by the user (either on a bean or otherwise)
- On the display page list all CDs entered in a table

# Using MVC

- Remember MVC basics
  - Object Model – Model
  - JSP - View
  - Servlets – Controller
- JSPs should not interact with DB directly – it violates MVC principles
- Servlet should be responsible for data access and forward to JSP (the view) for display
- See DisplayPeopleFromDBServlet