Chapter 10
Servlets and Java Server Pages
10.1 Overview of Servlets

• A servlet is a Java class designed to be run in the context of a special *servlet container*

• An instance of the servlet class is instantiated by the container and is used to handle requests directed to that servlet

• In the most common case, servlets are used to create responses to HTTP requests
10.1 Servlet Request

Browser

Servlet Container

Servlet

HTTP Request

Request object

Response object

HTTP Response

Response

HTTP Response
10.1 Servlet Advantages

• Since servlets stay in existence while the server/container is running, they can remember state
• Java is a more robust development language
• Because the servlet stays running, it is potentially more efficient than CGI
  • CGI programs are started for each request
  • Improvements, such as mod_perl in the Apache web server, reduce much of the overhead of CGI by keeping programs in memory even between requests
10.2 Servlet Details

- Servlet class implement the Servlet interface
- Several convenience classes are provided that implement Servlet
  - GenericServlet
  - HttpServlet
- Since most servlets respond to HTTP requests, the most common way to implement a servlet is to extend the HttpServlet class
10.2 HttpServlet Details

• The class provides four methods to handle different types of HTTP requests
  • doGet
  • doPost
  • doPut
  • doDelete

• An extension class will implement one or more of these methods

• Each method is called with two parameters
  • A request parameter containing data about the request
  • A response parameter that is used by the servlet to create the response

• doGet and doPut are the only methods used in this text
10.2 Responding to HttpServlet Request

- The HTTP request is mapped to a servlet by the servlet container
  - A configuration file provides a standard way of mapping paths to servlet classes

- The HttpServletResponse object passed as a parameter to doGet and doPost provides a PrintWriter object

- Output sent to the PrintWriter object will become part of the response

- The HttpServletResponse object has a setContentType method that takes the MIME type of the response as a parameter
10.2 Generating a Request

• As with CGI, there are two main ways of invoking a servlet
  • A hyperlink that specifies a path to the servlet
  • A form action that specifies a path to the servlet
• The tstGreet.html and Greeting.java files give a simple example in which no data is sent with the request
10.3 A Survey Example

- This example presents a simple survey
- Site visitors fill out a simple survey
- Survey results are recorded and stored in a file
- A summary of survey results is presented
- The getParameter method of HttpServletRequest is used to get the data sent from the survey form
10.3 Survey Example: Race Condition

• Since multiple requests may be processed at roughly the same time, some mechanism is needed to prevent the requests from interfering with each other
  • Such possible interference is known as a race condition

• The Java synchronized clause is used to prevent multiple threads executing file access code at the same time
10.4 Cookies

- HTTP is a *stateless* protocol, that is, the server treats each request as completely separate from any other.
- This, however, makes some applications difficult.
  - A shopping cart is an object that must be maintained across numerous requests and responses.
- The mechanism of cookies can be used to help maintain state by storing some information on the browser system.
- A cookie is a key/value pair that is keyed to the domain of the server.
  - This key/value pair is sent along with any request made by the browser of the same server.
- A cookie has a lifetime which specifies a time at which the cookie is deleted from the browser.
10.4 Cookies and Security

• Cookies are only returned to the server that created them.

• Cookies can be used to determine usage patterns that might not otherwise be ascertained by a server.

• Browsers generally allow users to limit how cookies are used.
  • Browsers usually allow users to remove all cookies currently stored by the browser.

• Systems that depend on cookies will fail if the browser refuses to store them.
10.4 Servlet Support for Cookies

- The Java servlet support library defines a `Cookie` class
  - Methods are provided to set the comment, set a maximum age, and set a value
  - Other methods retrieve data from the object
- The `HttpServletResponse` object has an `addCookie` method
- Cookies must be added before setting content type in the response
- The `HttpServletRequest` object has a `getCookies` method that returns an array of `Cookies` from the request
10.4 An Example

• The ballot example has two components
  • Ballot.html has a form used to cast a vote
  • VoteCounter.java defines a servlet which counts the votes for each candidate

• The response page to a user casting a ballot carries a cookie. This is used to ‘mark’ a user as having voted

• The vote tabulating servlet checks for the cookie and refuses to tabulate a vote if the cookie is provided with the request
10.4 Session Tracking

• In the Java servlet framework, sessions are sets of key/value pairs
• The HttpSession object implements a session
• Several methods are provided to manipulate values
  • putValue defines a key/value pair
  • Invalidate destroys the session
  • removeValue removes a key/value pair
  • getValue retrieves a value given the key
• A session object, if defined, is attached to the request object
  • The programmer can access the object
  • The programmer can specify on access that the session be created if it does not yet exist
• An alternate vote counting servlet uses sessions to check for duplicate voting
10.5 Java Server Pages

- Java Server Pages (JSP) provide a way of embedding active content in a web page
- Servlet containers manage JSP’s also
- A Java Server Page is first converted to a servlet which is then operates as previously described
10.5 Motivations for JSP

- Creating HTML documents using println is tedious and error prone
- Separation of coding and web page development can be more efficient for a team of developers
- On the other hand, if there is too much code embedded in the web page, the reverse problem arises
10.5 JSP Documents

• JSP documents can be created in two different ways
  • The classic syntax uses specially formatted tags, generally starting with `<%`
  • The newer XML syntax uses valid XML

• JSP documents contain four kinds of elements
  • XHTML code, called *template text*
  • Action elements
  • Directives
  • Scriptlets

• Template text is passed through to the response unchanged
10.5 Action Elements

• Action elements create content
• There are three categories of action elements
  • Standard action elements
  • Custom action elements
  • JSP Standard Tag Library (JSTL) elements
• Standard action elements are defined by the JSP standard and include basic services such as element generation and file inclusion
• Custom action elements are defined by creating Java code
• The JSTL is a collection of custom tags that provide important utilities
10.5 JSTL

- The JSTL contains five sub-libraries
  - Core tags
  - XML Processing
  - Internationalization and formatting
  - Database access
  - Functions

- JSTL also supports an expression language
10.5 Directives

- Directives are tags that begin with `<%@`
- Directives define the environment in which the JSP is interpreted
- A page directive provides information such as content type
- The taglib directive is used to make libraries of custom tags available to the JSP
  - JSTL tags must be imported with a taglib directive
    `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>`
    - Is used to allow the current JSP refer to the JSTL core library
    - Tags from that library will use the `c:` qualifier
10.5 Scriptlets

- Scriptlets allow embedding programming language code into a JSP
  - Although extensions can be used to support other languages, Java is the one that must be supported

- The expression scriptlet
  
  `<%= expression %>`

  Causes the value of the expression be put into the response

- General Java code can be enclosed within `<% … %>`
- JSP comments `<%-- … --%>` are not put into the response
  - Regular HTML comments `<!-- … -->` are put into the response
10.5 Temperature Conversion Example

- `tempconvert0.html` and `temconvert0.jsp` provide a temperature conversion example
- `Tempconvert1.jsp` is similar but both pages are integrated into the same JSP
  - A Java if is used to conditionally include content in the response
  - If the request comes with a data value with key `ctemp`, it is assumed that this is a request from the form
  - Otherwise, it is assumed that this is the first request and only the form is sent
10.5 Expression Language

• The JSTL expression language (EL) uses ${ .. } to indicate an expression

• The expression language includes standard operators
  • In some cases alternate names are provided to avoid problems with the HTML special characters
  • So, ge is provided as a synonym for >=

• The param object is predefined in EL to provide data submitted with an HTTP request
  • ${param.name} gets the value associated with name
  • ${param['fancy name']]} gets the value if the name is not a proper identifier

• It is usually best to use the JSTL core tag c:out to put the value of an expression into the response

• Tempconvert2.html and tempconvert2.jsp implement temperature conversion using EL
10.5 JSTL Control Action Elements

- The JSTL core library defines a number of control structures
- The `c:if` tag defines a one way branch, no else is allowed
- Tempconvert3.jsp uses the `c:if` tag to determine if the request being sent uses the POST method or not
  - If the POST method is used, it must be a form submission, so data is accessed and the conversion is carried out
  - If the GET method is used, this must be a first request for the page, so the form itself is returned
10.5 JST foreach

• The c:foreach tag provides iteration
  • Iteration through a list of values is supported
  • Iterations through a sequence of numeric values is supported

• If, for example, several checkboxes have the same name attribute, the value of parmValues.name will be a list of the values

  <c:foreach items="${paramValues.name}" var="x">
  • Will step the variable x through each value in the list
10.5 JSTL choose

• The c:choose tag provides a multi-way choice
• The testradio.jsp example uses c:if to determine the method of the request
• If the method is POST, the JSP uses the c:choose construct to determine which text to put into the response