

Java Server Pages

- Java Server Pages (JSPs) provide an alternative to complete generation of pages by servlets
- JSPs allow the designer to embed dynamic calls within an HTML document
- JSP technology:
 - Separates dynamic content generation from page design
 - Emphasizes the use of reusable components
 - Simplifies page development by allowing page changes without reloading
- By convention, the document extension should be .jsp to indicate to the server that it is an html page with JSP calls

How JSP Works

- A Java Server Page is created which consists of HTML or XML tags and content interspersed with JSP elements
- The JSP engine converts the JSP to a servlet
 - The HTML/XML parts are simply passed through
 - The JSP elements are acted on
- You can examine the servlet code generated by looking in the appropriate place
 - These servlets will use the classes `JSPPage` and `HttpJspPage`
 - Three methods are defined
 - `jspInit()`
 - `jspDestroy()`
 - `_jspService(request, response)`
- This means the user can define the init and destroy methods for the servlet created from the JSP page

September 28, 2001

JSP and JavaBeans

3

JSP Page Components

- A JSP page is made up of four types of JSP elements:
 - Directives which describe global page information
 - `<%@ %>`
 - Declarations which are page wide variables and methods
 - `<%! %>`
 - Scriptlets which are actual code fragments
 - `<% %>`
 - Expressions which are variables to be processed to strings for output onto the page
 - `<% = %>`

September 28, 2001

JSP and JavaBeans

4

A Sample JSPPage/Program

```
<HTML><HEAD><TITLE>Hello</TITLE></HEAD>
<BODY>
  <%@ page language="java" %>
  <%! int I=0 %>
  <% for (I=0;I<10;I++)
  { %>
  <BR>This is line
  <%= I %>
  <% } %>
</BODY></HTML>
```

Directives

- Directives take the general form
- `<%@ directivetype attribute = value...%>`
- The directive types are:
 - Page – information for the page
 - Include – verbatim includes
 - taglib – a to be uri for a library of tags to be included
- Selected attribute value pairs include
 - language="java"
 - extends | imports = "package.class,..."
 - session = "true | false"
 - buffer="non | 8kb | nkb"
 - autoFlush="true | false"
 - isThreadSafe="true | false"

Declarations and Expressions

- Declarations are used primarily to define class level variables for the page
- Declarations include both variable definitions and method definitions

```
<%! int count = 0;
    private void incrementcounter()
    {
        count++
    }
%>
```

- Expressions produce a string to be included in the output
 - Your name is <%request.getParameter("name")%>

Scriptlets

- Scriptlets are pieces of Java code embedded within the JSP page
- Scriptlets are like any other piece of Java code and have the ability to make use of a series of objects made available through the `_jspService()` method.
 - request and response
 - pagecontext (set in large part by the directives)
 - session object
 - out (the output stream)
 - application – the servlet context

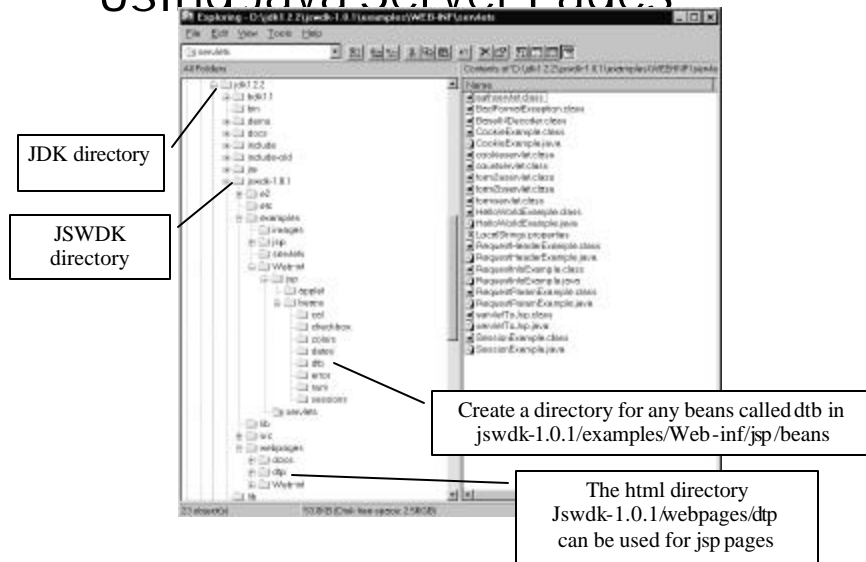
JSPs and Enterprise Java Bean

- JSP's provide a framework for using Enterprise Java Beans(EJBs).
- EJBs provide a way to access reusable code served by an application framework – a component transaction monitor.
- JSP actions or tags of the form `<jsp:action..... />`. At the current time, the primary action is “usebean”. Other actions include:
 - setProperty
 - getProperty
 - forward URL
 - include URL

Attributes of the jsp:usebean tag

- The jsp:usebean tag has five attributes of which three are required
 - id=“name” specifies the name of the bean as it was registered with the CTM
 - scope=“scope” defines the scope for the bean. Normally the page, but possibly a request, session, or application
 - class=“package.class” is the fully qualified name of the bean class

Using Java Server Pages



September 28, 2001

JSP and JavaBeans

11

Preparing a Java Bean

- Java Beans have a number of special features which we will ignore at this point in time. We will work here simply to show how a bean could be used in a JSP
- For our purpose, a Java bean will simply be a class that has methods we wish to use.
- After compilation, store the bean in the directory (jswdk-1.0.1/examples/Web-inf/jsp/beans/ dtb)
- The java source file will start with the line:

```
package dtb;
```

 - Note that the package name, dtb, is the same as the name of the directory our beans are stored

September 28, 2001

JSP and JavaBeans

12

A Simple Bean

```
package dtb;
import java.io.*;
// an addition bean
public class addition
{
    protected int value1;
    protected int value2;
    public String setValues(int i, int j)
    {
        value1 = i;
        value2 = j;
        return "Set Values Complete";
    }
    public String getSum()
    {
        return "<P>The sum of "+value1+" and "
            +value2+" is " +(value1+value2)+"</P>";
    }
}
```

An Simple JSP

```
<%@ page import = "dtb.*" %>
<jsp:useBean id="add" class="dtb.addition"
    scope="page"/>
<jsp:useBean id="sub" class="dtb.subtraction"
    scope="page"/>
<html><head>
<title>A simple servlet page</title>
</head>
<body bgcolor="#FFFFFF">
<h2>HERE IS SOME INFORMATION:</h2>
<%= add.setValues(23,45) %>
<%= add.getSum() %>
<%= sub.setValues(23,45) %>
<%= sub.getSub() %>
</body></html>
```

An Exercise

- Write a bean that returns a standard page footer. It should return the following information in a string with good HTML formatting:
 - A webmaster contact name for reporting problems
 - A corporate identity
 - A date/time stamp
- Write a JSP page that shows your name and contact information and includes the standard bean as a footer.

A Suggested Project

- Development of a prototype corporate website that:
 - Provides a page for users to login to the site (assume they are in the DBMS)
 - Based on successful login, returns a cookie (use the userid) that identifies the user and requests selection of a product category(using a form). This page should include the users name in the text.
 - Based on selection of an item from this form, a new form will be returned allowing the user to select a specific product.
 - When this form is returned, a confirmation form should be sent. If conformation is obtained, a transaction should be entered in the server and a static page showing the transaction presented to the user.
- For extra credit, allow users to register. See the next slide for a summary of the DBMS tables to be used.

Final Project DBMS

- UserID: username, password, userid (PK)
- Customers: userid(PK), lname, fname, add1, add2, city, state, zip, phone#, creditcardtype, cc#, ccexpdate, #transactions, standing, psm (preferred shipping method), EF1(ExtraField1), EF2, EF3, EF4, EF5
- Categories: ID(PK), productcategory, productID
- Products: productID(PK), productname, description, productcost, shippingweight, shippingheight, shippingwidth, shippingdepth
- Transactions: ID(PK), date, userid, productid