

Java Servlets and Java Server Pages (JSP)

Hilary Hutchinson
HCIL Staff/PhD Student
hilary@cs.umd.edu
May 10, 2005

Motivation

- **Accessibility**
 - Not everyone has .NET
 - Not everyone runs Windows
- **Avoid duplication** in HTML code
 - Many web apps have lots of content that can all be presented via same template
- More **efficient** and **powerful** than CGI
 - Java VM threads vs. OS processes
 - Store session info, talk to web server
 - Portable to many platforms



JSP vs. ASP

- Embed logic in an HTML page and parse it on the server-side
- ASP (Active Server Pages)
 - Platform dependent (Microsoft)
 - Uses scripting (VBScript or Jscript)
- JSP (Java Server Pages)
 - Platform independent
 - Uses a real programming language (Java)



JSP vs. Servlets

- 2 ways to do the same thing
- Regular Java programming
- Access HTTP request/response info
- JSP
 - Java code is embedded in HTML page
 - Page gets compiled into a servlet by server
- Servlets
 - Plain old Java classes
 - Extend `javax.servlet.http.HttpServlet`

JSP Example

```
<%@ page
  contentType="text/html; charset=WIN
  DOWS-1252"%>

<HTML>
  <HEAD>
    <TITLE>Hello World </TITLE>
  </HEAD>
  <BODY>
    <% out.println("Hello World"); %>
  </BODY>
</HTML>
```

Servlet Example

```
import java.io.*;
import javax.servlet.http.*;
import javax.servlet.*;

public class HelloWorld extends HttpServlet{

  public void doGet (HttpServletRequest req,
    HttpServletResponse res) throws
    ServletException, IOException {
    PrintWriter out = res.getWriter();
    out.println("Hello World");
    out.close();
  }
}
```



Open Source Tools

- Server: **Tomcat**
- Development Environment: **Eclipse**
- Database: **mySQL**
- Database Connectivity: **JDBC**
- Build tool: **Ant**
- Text Search: **Lucene**



Servers

- Tomcat
 - Free from Apache:
<http://jakarta.apache.org/tomcat>
 - Run alone or integrate into Apache
 - Easy to install on Windows or UNIX:
<http://www.coreservlets.com/Apache-Tomcat-Tutorial>
- Others (\$\$)
 - IBM WebSphere
 - Allaire JRun
 - Oracle Application Server



Tomcat Examples

- JSP

- <http://localhost:8080/HelloWorld.jsp>

- Servlet

- <http://localhost:8080/servlet/HelloWorld>



Real Example: ICDL

- Online library of scanned children's books

- 10,000 books
 - 100 languages

- Technical needs

- Client-side accessibility
 - Templates to handle duplication
 - Multi-lingual support
 - Cheap or free



Why Java Servlets?

- **Accessibility**
 - Processing, db calls done on server side
 - Client just sees HTML, JavaScript, CSS
- **Duplication**
 - All books presented with same page
 - All languages presented with same page
- **Multi-Lingual Support**
 - Java is Unicode-compliant



Saving State

- **HTML is stateless**
- **State needed as you move page to page**
 - What book am I reading?
 - What language am I using?
 - What global information can I access?
- **Servlets provide 3 ways to save state**
 - Per page: url variables
 - Per user: session variables
 - Per servlet: context variables



Per Page: Url Variables

- Quick and dirty – like HTML forms

- Client side:

www.icdlbooks.org/icdl/BookPreview?id=123

- Server side:

```
public void doGet(HttpServletRequest request, HttpServletResponse response) {  
    String id=request.getParameter("id");  
}
```



Per User: Session Variables

- Remembering state about a user

- Setting:

```
public void doGet(HttpServletRequest request, HttpServletResponse response) {  
    HttpSession ses=request.getSession(true);  
    Profile profile=new Profile("Hilary")  
    ses.setAttribute("profile",profile);  
    ses.setMaxInactiveInterval(1000);  
}
```

- Getting:

```
HttpSession ses=request.getSession(true);  
Profile profile=ses.getAttribute("profile")
```



Per Servlet: Context Variables

- Remembering state that all users need

- Setting:

```
public class ContextListener implements
    ServletContextListener {
    public void contextInitialized(
        ServletContextEvent event) {
        context = event.getServletContext();
        String home="www.icdlbooks.org";
        context.setAttribute("home", var);
    }
    public void contextDestroyed
        (ServletContextEvent event) {
        context.removeAttribute("home");
    }
}
```



Per Servlet: Context Variables

- Getting:

```
public class MyServlet extends HttpServlet{
    public void doGet(...){
        ServletConfig config=getServletConfig();
        ServletContext context=
            config.getServletContext();
        String home=
            (String)context.getAttribute("home");
    }
}
```


Internationalization

■ Unicode Compliant Software

- Database (mySQL)
- Driver (com.mysql.jdbc)
- Servlet

```
public void doGet(HttpServletRequest req,
    HttpServletResponse res) {
    response.setContentType("text/html;
        charset=utf-8");
}
```

Internationalization

■ Properties files

- Key/value pairs for strings presented
- English: **myfile_en.properties**
 - helloworld = Hello World
- Spanish: **myfile_es.properties**
 - helloworld = Hola Mundo

■ Servlet Code

```
Locale loc=Locale.US;
ResourceBundle bundle =
    ResourceBundle.getBundle("myfile",loc);
String hello =
    messages.getString("helloworld");
```