Java Server Pages, JSP

BWA Slides #6

… change rate is high…

Java Server Pages, JSP's, is possible a bit "previous generation" technology

- But there are a lot of application out there using them
- The concept is common to other frameworks (classical ASP, PHP)
- Clarify the basics
- Sometimes handy

We'll later inspect Java Server Faces the "GUI-standard" for JEE

Why JSP?

Problems

- Servlets need programmer skills
- Servlets mix Java and HTML
- Tedious: out.println("<h1>…</h1>")…
- Servlets should not be in the view part of MVC

Solution: JSP

- A JSP <u>is a Servlet</u>, … but written in a "<u>tag language</u>"!
- Simple for non-programmers to write a "page"
- Faster development
- In the view part of MVC

JSP Location in Project

JSPs lives in "Web Pages" folder (Maven: src/main/webapp)

Best practices: Put in src/main/webapp/WEB-INF/jsp directory.

- I.e. application private, can't access using browser address field

JSP Life Cycle

At first request...
- Translated to Servlet by container
- Compiled

...then as a Servlet
- ...loaded, instantiated, service, destroyed

First response slower

JSP Syntax

Two different syntaxes

- Standard (default in NetBeans), banned ...very common in examples on web (file suffix *. jsp)
- JSP document, an XML syntax, use (file *.jspx)!
- Select document in NetBeans when creating JSPs

JSP Content

JSP possible content
- XML prolog (mandatory)
- The HTML tags (static content)
- Some JSP tags. Starts with jsp, <jsp:root… >, <jsp:include… >, …
- Some JSP Standard Tag Library (JSTL) tags, all starts with c <c:if … >, … upcoming
- The Unified Expression Language, an embedded expression language, we just say Expression Language (EL)

The Expression Language

The EL allows page authors to use simple expressions to dynamically access data from Java objects

- String expressions in the pages matches “named” object
- Used in JSP pages and Java Server Faces (upcoming)
- In JSP pages using immediate evaluation
    - Expression immediately evaluated and result put into page at page display
    - Immediate syntax: **$**{ … expression … } (**$**=immediate)
    - Immediate syntax is read only (good, this i the view part)
- Also arithmetic, string concatenation, functions, …

Set and Get Dynamic Page Data

The scopes are "automagically" searched in EL expressions (direction wider scopes)

In Servlet

```
// Store customer in session
session.setAttribute("c", customer);
```

In JSP

```
// All scopes searched for key 'c'
// get object,call getId().toString()
${c.id}    // EL
```

NOTE: Final method call must return primitive type or String

JSP Standard Tag Library, JSTL

Need for modification/selection/generation of dynamic content in the JSP's
- Typically a loop to create at table
- Need statements in pages: assignment, selection, iteration, ...

Solution: JSTL tags
- Used in conjunction with EL
- JSTL is not for application logic, view only
- We only need the <c: ... > tags

JSTL cont.

JSTL core tags, must declare namespace

```
// Namespace (c:)
<jsp:root  xmlns:c="http://java.sun.com/jsp/jstl/core” ...>
// Assignment
 <c:set var="mB" value="${requestScope['myBean']}" />
// Selection
 <c:choose>
     <c:when test="${i lt 5 }">...</c:when>
     <c:otherwise>...</c:otherwise>
 </c:choose>
// Iteration
 <c:forEach var="row" items="${table.data}">
     ...
 </c:forEach>
```

pom.xml, must add dependency javax.servlet, artifactId: jstl, version 1.2

JSP Design Issues

JSP is a view technology

- Should only display data i.e. do "reads" to get it
- Should normally not handle incoming requests (will be handled by a Servlet), see later MVC model

We never put inline Java code in the JSP pages (aka "scriptlets")

- Common for bad, deprecated, code samples on web

Modular JSP's

Anything a Servlet can do a JSP can

- Forward, redirect and include
- Using include in conjunction with EL we can build modular pages
- Tags to include content: `<jsp:include … >`,`<jsp:directive.include … >`
- Possible to pass parameters to included pages using

  `<jsp:param … >`
- Sample next slide

JSP Modular Page

```
<jsp:root>
    …
    <html>
         <jsp:include page="partials/head.jspx" >
             <jsp:param name="title" value="${title}" />
         </jsp:include>
         <body>
             <!-- Static content so use static include -->
             <jsp:directive.include file="partials/header.jspx" />
             <jsp:directive.include file="partials/mainMenu.jspx"/>
             <section id="content">
                 <!-- Insert the dynamic content -->
                 <jsp:include page="${content}.jspx" />
             </section>
             <jsp:directive.include file="partials/footer.jspx" />
         </body>
     </html>
</jsp:root>
```

JSP and CSS

No problem, same as HTML

```
...
<head>
 <link rel="stylesheet" href="default.css" type="text/css"/>
 <title>tempate.jspx</title>
</head>
<body>
...
</body>
...
```