Java Server Pages, JSP

BWA Slides #6

… change rate is high…

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

- But there are a lot of application out there using them
- The concept is common to other frameworks (ASPX)
- 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 (MVC)

Solution: JSP

- A JSP is a Servlet, ... but written in a tag language!
- Simple for the non-programmer
- Faster development
- In the view part of (MVC)

Specification: JSR 152, Java Server Pages 2.0

JSP Location

JSP's live in projects "Web Pages" folder (Maven: src/main/ webapp)

Best practices: Put in WEB-INF/jsp directory.
- Application private, can't access with browser address field

JSP Life Cycle

At first request…

- Translated to Servlet by container
- Compiled

…them as a Servlet

- …loaded, instantiated, service, destroyed

First response slower

JSP Language

Two different syntax's

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

Select **document** in NetBeans when creating JSP ...
...and of course inspect code samples

JSP Content

- XML prolog
- The HTML tags (static content)
- Some JSP tags
    - All starts with jsp, <jsp:root...>, <jsp:include...>, ...
- Some Java Standard Tag Library, JSTL, tags (upcoming)
    - We use only the core tags, <c:.../>

The Expression Language, a small embedded language, compare XPath

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

To read data (dynamic data in the pages) we use the Expression Language (EL)

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

- Common for bad, deprecated, code samples on web

The Expression Language, EL

Would like to read data from a Java object to display

- EL is **read only**
- Purpose of object is to supply data to a page (aka backing bean or page bean more later...)

In pages get data as ${object.data}

- Will call method named getData() on object
- ${...} delimits the expression
- method must return primitive or String value
- If inside element ok: <body>${...}</body>
- If attribute value ok: <... name="${...}"...>
- String concatenation: "This is ${...} from ${...}"

The Expression Language cont.

Inside ${...} we can have

- Literals: 34, 45.6, ...
- Operators: Arithmetic, Logical, Relational,... mostly same as Java (+, -, *, &&, ||, <, >, ...)
- Some reserved words: and, or, not (same as operators), empty (is value null?)
- [ ] and .
- Only used for "view" calculations, no application logic!

Indexing and dot operators, are (almost) the same a['b'] = a.b

EL Implicit Objects

Many implicit object, accessible from EL

- The scopes
    - ${requestScope},
    - ${sessionScope},
    - ${applicationScope},
    - ...same object as HttpRequest, HttpSession, ServletContext
- Also a ${pageScope},page local variables, only visible in page, new page -> object gone
- Request parameters ${param['myParam']}, normally avoid (requets goes to control parts, i.e. Servlets)
- .. and more

Set and Get Dynamic Page Data

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

Set: Somewhere in application (store a customer object)

```
    session.setAttribute("c", customer);
```

Get: In an EL expression in page (retrieve object):

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

Modular JSP's

Anything a Servlet can do a JSP can

Forward, redirect and include

- Include static: Included compile time (included never change), low workload
- Include dynamic: Included every request, included changes, higher workload

Java Standard Tag Library, JSTL

Need for modification/selection/generation of dynamic content in the JSP's

- Typically a loop to create at table
- Need some statements: assignment, selection, iteration, ...

Solution: JSTL tags for statement

- Used in conjunction with EL

Also other JSTL tags for formatting, database, XML,

- ...not used in course

JSTL is not for application logic, view only

JSTL cont.

JSTL Core tags, must have namespace declaration

```
<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, dependency javax.servlet, artifactId: jstl, version 1.2

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>
...
```