



CS544 EA
Applications

Web: JSP with EL & JSTL

View Technology

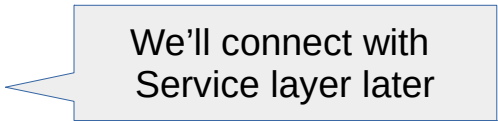
Pre-compiler transforms
XML into Java

- JSP is a type of Servlet written in XML
 - Java's oldest web based view technology
- Write logic with XML tags (for loops etc)
 - JSTL: Java Standard Tag Library

Originally JSP had you write Java code
inside HTML
This is now considered a bad practice
Deprecated!

Model View Control

- Best practice:
 - Controller receives request
 - Gets data from Service layer
 - Forwards data to (JSP) View



We'll connect with
Service layer later

Servlet Controller

```
@WebServlet(name = "Hello", urlPatterns = { "/Hello" })
public class Hello extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        request.setAttribute("now", new Date());
        request.setAttribute("one", 1);
        request.setAttribute("two", 2);

        ServletContext context = this.getServletContext();
        String jsp = "/Hello.jsp";
        RequestDispatcher dispatcher = context.getRequestDispatcher(jsp);
        dispatcher.forward(request, response);

    }
}
```

Instead of retrieving data
we'll just create some here
and store inside request

Then forward to
Hello.jsp

Hello.jsp with JSTL and EL

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello JSP</h1>
    ${now}

    <div>
      <c:if test="${two gt one}">
        ${two}
      </c:if>
    </div>

  </body>
</html>
```

EL can be used to print
or used in statement tag

Uses now, one, two
stored in request

<c:if> to create if
statement

Browser shows:

Hello JSP

Tue Apr 30 12:52:13 CEST 2019
2

Expression Language

- `${expression}`
- `${object.value}`

Uses `.getValue()`
method

Easier for
non-Java people?

- Supported Operators
 - Arithmetic: `+`, `-`, `*`, `/` (div), `%` (mod)
 - Relational: `==` (eq), `!=` (ne), `<` (lt), `>` (gt), `<=` (lte), `>=` (gte)
 - Logical: `&&` (and), `||` (or), `!` (not)
 - Other: `()`, empty, `[]`

EL Scopes

- When using a variable in EL it uses the name as a key to find a value in the following scopes:
 - Page scope
 - Request scope
 - Session scope
 - Application scope

First to Last
↓

What we used in our example,
and what you should mostly use

JSTL

- Example JSTL core tags:

Other libs can
also be loaded

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

- Import the core tag lib with the prefix “c”

```
<c:if test="${two gt one}"> ... </c:if>
```

- Use the <c:if> tag to make an if statement

```
<c:forEach items="${customers}" var="customer"> ... </c:forEach>
```

- Use the <c:forEach> tag to make a loop

JSTL Core Tags

Tag	Description
<c:out>	Output result of evaluating an expression
<c:set>	Sets a key in one of the scopes
<c:remove>	Removes a key in one of the scopes
<c:catch>	Catches any java.lang.Throwable inside it
<c:if>	Executes content if test is true
<c:choose>	For a multi condition if (if/else)
<c:when>	Specify one (or more) conditions inside a <c:choose>
<c:otherwise>	Specify what should happen if none of the <c:when> are true
<c:import>	Ability to import data from a url (does http request)
<c:forEach>	Basic Foreach
<c:forTokens>	Iterates over tokens separated by the supplied delimiter
<c:redirect>	Redirects to a URL (HTTP 3xx)
<c:url>	Encodes a URL with optional parameters
<c:param>	To provide parameters to the url and import tags

Formatting Tags

Tag	Description
<fmt:message>	Gets message from bundle (for internationalization)
<fmt:param>	To give parameter to <fmt:message>
<fmt:bundle>	Specify a resource bundle to use
<fmt:setLocale>	Sets the locale
<fmt:requestEncoding>	Sets the request's character encoding
<fmt:timeZone>	Specifies the timezone for formatting
<fmt:setTimeZone>	Stores the specified timezone into a variable
<fmt:formatNumber>	Format a number based on the set locale
<fmt:parseNumber>	Parses a string of a (number, percentage, currency)
<fmt:formatDate>	Formats dates and times in a locale sensitive way
<fmt:parseDate>	Parses string representations of times and dates

Spring Form TagLib

Tag	Description
<form:form>	Parent element for all other spring form elements
<form:input>	Typical input element (now also supports HTML5 types)
<form:checkbox>	Checkbox element
<form:radiobutton>	Radio button element
<form:password>	Password element, optional showPassword attribute
<form:select>	Dropdown
<form:option>	You can specify a single option in the dropdown
<form:options>	Or specify a list of items
<form:textarea>	Textarea
<form:hidden>	Hidden field
<form:errors>	To show validation errors

Older Spring tags exist but most of the functionality moved into Spring Form tags

Spring Form Example

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
```

```
...
```

```
<form:form modelAttribute="user">
```

```
  <table>
```

```
    <tr>
```

```
      <td>First Name:</td>
```

```
      <td><form:input path="firstName" /></td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>Last Name:</td>
```

```
      <td><form:input path="lastName" /></td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td colspan="2">
```

```
        <input type="submit" value="Save Changes" />
```

```
      </td>
```

```
    </tr>
```

```
  </table>
```

```
</form:form>
```

Takes firstName from user
automatically displays
validation Error (if present)

Java is Deprecated inside JSP

- There were several ways to write Java in JSP
 - Pure Java code inside JSP (scriptlets)
- And elements that let you use Java:
 - JSP Expressions: `<%= user.getName() %>`
 - JSP Declarations: `<%! int a = 5 %>`
- These are all deprecated (**don't use them**)