CS 416
Web Programming

JSP Custom Tags

Dr. Williams
Central Connecticut State University

Review from last class

- **Use the two shells review.jsp and result.jsp**
- **You should create a bean to hold the fields on the review.jsp page**
- **On the results page you should populate the bean with the passed in fields**
- **Display out the fields from the bean**
- **Add functionality to keep track of the total cost of all items added by the user**

JSP custom tags

- To extend the functionality of JSPs developers are given the ability to create their own tags
- By creating your own tags you are able to specify an action to take place on the page based on some passed in parameters

How they work

- Essentially what a custom tag does is allows you to call functionality from within your page and pass in parameters
- The tag code will then receive the passed parameters and performs some functionality and can write some output to the page

Creating custom tags

- There are two ways to create custom tags
  - Extend SimpleTagSupport

    (javax.servlet.jsp.tagext.SimpleTagSupport)
  - Create a tag file
- Both ways essentially provide the same type of functionality but each way has its own advantages

SimpleTagSupport

- By extending SimpleTagSupport you are creating a new Java class to handle calls to a tag you are defining
- The file you are creating is similar to a cross between a servlet and a Java Bean
- The class has a set of attributes and each attribute must have a public setter

SimpleTagSupport cont.

- The other element is the **doTag()** method
- Whenever the custom tag is called from the JSP the page will call all of the classes setters to initialize the values to those specified on the page
- Next the doTag method will be called where it will perform any functionality and possibly write to the page

Tag Library Descriptor

- For a SimpleTagSupport tag to be accessible to JSPs it must be registered in a Tag Library Descriptor file
- The descriptor xml file is similar to the web.xml file it is responsible for:
  - Registering tag names
  - Associating them with classes
  - Then specifying the attributes that the tag handler accepts and if those attributes are required or not
- Once the file is created a URI will be associated with the library descriptor which JSPs will refer to incorporate the tag

Using your tag

- To use your SimpleTagSupport tag within a JSP it must be specified at the top of the JSP file

```
<%@taglib prefix=“myTags” uri=“MyTagLibrary” %>
```

The URI refers to the uri declared in the tag library descriptor that contains the class you just created

Using your tag cont.

- Within the page you can then call your tag by using the tag prefix and your SimpleTagSupport class name

```
<%@taglib prefix=“myTags” uri=“MyTagLibrary” %>

<myTags:myClass param1=“value1” param2=“value2” />
```

This will initialize your class, call the setters for param1 and param2 and then call the doTag() method

SimpleTagSupport demo

- TagIndex.jsp, LabledTextField.java

When to use SimpleTagSupport

- When you would want to extend SimpleTagSupport?
  - The primary advantage of extending SimpleTagSupport is code is straight Java code which can be difficult to read if contained in syntax like a JSP
  - It is good for implementing functionality that focuses on calculations or complex operations rather than display logic
  - It isn’t a good choice if there is a lot of HTML that needs to be output

Tag files

- Tag files provide the same functionality as extending SimpleTagSupport but are aimed more at functionality oriented around display elements
- Syntax is very similar to JSP making it ideal for integrating a mixture of HTML components with functional logic

Tag files cont.

- Tag files are declared in files with the ".tag" extension
- The top of the file it lists the attributes the tag file accepts and indicates if the attribute is required or not

```
<%@attribute name="param1" %>
<%@attribute name="param2" required="true" %>
```

- The default type is java.lang.String, but you can pass any other object type by specifying the type

```
<%@attribute name="person" type="edu.ccsu.Person" %>
```

Tag file syntax

- The contents of a tag file are similar to a JSP
- Straight HTML is mixed with dynamic elements
- Dynamic elements can be included with <% %> like JSP
- Dynamic elements can also be referenced using ${ } syntax

Tag file syntax cont.

- The ${ } syntax is used to access passed in attributes or java beans that it has specified it is using (same as with a JSP)

```
<table>
  <tr><td>${name}</td></tr>

<tr><td>${myBean.attr1}</td></tr
>
</table>
```

Using your tag file

- Unlike the SimpleTagSupport classes you do not need to directly register your tag file
- On the JSP you will instead include the directory where your tag files reside

```
<%@taglib prefix="myTagFs" tagdir="/WEB-INF/tags" %>
```

Once included using the tags is identical

```
<myTagFs:myClass param1="value1" param2="value2" />
```

Tag file demo

- EnterAddress.jsp, TagResult.jsp, Address.tag

When to use tag files

- When you would want to use tag files?
    - The primary advantage of using tag files is when a lot of HTML will be used to prevent it being spread throughout code
    - However like JSPs the syntax checking isn’t as robust at compile time which can lead to more runtime debugging

Custom tag file

- Given Item bean
- Form submits item information to page
- Create tag file for displaying item
- Tag file take one required parameter “store”

Seperating logic - MVC

- Database lookup

Exercise

- Draw application flow for this single call
  - User attempts to access their personal business homepage which requires authentication
  - Home page looks up relevant sales data
  - Page has rich display with complex reusable calculations
- Jot down bullet point explanation of technologies how integrated and how created

Exercise

- Draw application flow for this single call
  - User is accessing page they have been authenticated to (don’t worry about authentication just indicating you know the user)
  - When they click submit should only hit server if more than 3 characters entered
  - Upon looking up matching records if a sensitive record is returned add a SensitiveInfo flag to their session
  - Rich HTML returned, the rich HTML related to record display used several places throughout application
  - CEO wants to track how often SensitiveInfo flag is added to a session
- Jot down bullet point explanation of technologies how integrated and how created