Katie Hilliard

07/17/2025

CSD430 Module 10 Assignment

**Creating Custom Tags**

Custom tags are JSP language elements used to encapsulate recurring tags. They’re user defined and they separate business logic from JSP. This eliminates the need to use scriplet tags. Syntax for custom tags are written as:

<prefix:tag attr1="value" ... attrN="value" />

or

<prefix:tag attr1="value" ... attrN="value" >

body

</prefix:tag> .

The prefix separates library tags. Tag is the tag identifier, and attr1 … attrN are the attributes which modify the tag’s behavior. There are three required components to create a custom tag: tag handler, Tag Library Descriptor (TLD) file, and Taglib directive in jsp file.

Since the purpose of creating a custom tag is to make recurring tags, that would be one of the advantages of using custom tags. The reusability of code means this tag can be used on several pages to do things like provide formats for dates, birth years, or phone numbers. There’s no need to rewrite code if you can make a quick shortcut, which would be a custom tag. Another benefit would be the fact that the code will be easier to read and manage. Using custom tags makes the code cleaner and shorter, which also makes it more organized.

There’s so much you can do with custom tags, hence the fact that they are “custom.” You can define and reference new objects used as scripting variables. Also, empty tags have the ability to perform server-side work based on the attributes of the tag. Whatever action is done will determine if the rest of the page will be interpreted or if a redirect will take place. This helps check to see if the user is logged in prior to accessing a page, and will redirect them to log in if they aren’t. This is something most of us may be familiar with. After working on BlackBoard for so long, the login session will timeout and we’ll be required to login again if we want to go to another page from the idle screen.

Let’s go over the required components for creating custom tags. Starting with the tag handler, this object is container-managed, and it’s created by a JSP container during runtime. A Java class implements the tag’s processing logic and could have properties that correspond with the tag’s attributes or body. The Tag Library Descriptor (TLD) is a saved file with a .tld extension with a set of related tags. These tags are mapped to their tag handlers, along with a description of the tag with its’ attributes. Multiple tags can be encapsulated using the <tag> tag, and it’s kept in the WEB-INF folder. The tag handlers and TLD constitutes the tag library which is used in the JSPs after being distributed. Finally, the Taglib directive is what we use to access a certain tag library in a JSP. The tag library and it’s prefix are specified by the Taglib directive.

While the use of custom tags has several advantages, the one disadvantage I found would be how complex it can make the code. You go from having a lengthy code to shortening it using custom tags when needed, but using many tags can increase complexity. You would need to know how to use the tab libraries, which isn’t very beginner friendly. We must also keep in mind that JSP is constantly changing and its’ evolution may eventually affect the effectiveness of using custom tags.

In my opinion, new developers can always find time to familiarize themselves with the tag libraries so it’ll be easier to use them when coding. I would definitely suggest to consider creating custom tags. I find them very beneficial to developers as we may use JSP-based applications. We can save time since we don’t have to rewrite code and we can use multiple tags on several pages.

To conclude, creating custom tags in JSP is a useful method that is still relevant in web design today. The code reusability and separation of logic supports the idea of modular design. The complexity of using custom tags is only a disadvantage for beginners, but once familiarized, it can be a very beneficial technique. This way, developers are able to effectively use the tags created by others within their code.

Here's an example of the header included in the <taglib> tag, which is the root tag of the TLD document:

<taglib version="2.0" xmlns="http://java.sun.com/xml/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary\_2\_0.xsd">

References:

Oracle. (n.d.). *Understanding and creating custom JSP tags*. In *Programming WebLogic Server JSP tag extensions: Quick start*. Retrieved July 17, 2025, from <https://docs.oracle.com/cd/E11035_01/wls100/taglib/quickstart.html#wp362707>

GeeksforGeeks. (2025, April 28). *Custom tags in JSP*. Retrieved July 17, 2025, from <https://www.geeksforgeeks.org/java/custom-tags-in-jsp/>