Creating Custom Tags

Forey, Truman

Bellevue University, CSD-430

Module 10 Assignment

7/20/2025

**Using Custom Tags**

One of the primary benefits of using JSP pages is that it can easily integrate Java code into the webpage, without having to rely on managing JavaScript. These are called “scriplets”, and are highly useful for server-side programming. However, these tools need to be used responsibly. In many cases, developers may find themselves using similar scriplets across multiple pages in multiple contexts. In these cases, developers may benefit from creating custom HTML tags that can be referenced across multiple pages. While implementing custom tags can introduce a lot of complexity in how they’re implemented, they can also streamline development by allowing developers access to reliable pieces of code and make it easier to scale up websites by making it all more modular.

The process of implementing custom tags can be broken down into four distinct actions: declaring a tag library that the .jsp file draws from, assigning a prefix to the tag library that the .jsp page is able to parse with, calling the custom tag within the .jsp page, and finally creating the java code for the tag. (The J2EE 1.4 Tutorial, 2025) The first step, declaring a tag library, is done with the ‘taglib’ command and the ‘uri’ command within it. The ‘taglib’ command tells the page to start looking for a custom tag within a tag library, while the ‘uri’ command tells the program where to search for the tag library from the root folder. Next, the ‘prefix’ command is used to give the page a unique identifier for the tag library while it’s being used. Once this is done, the custom tag can be used by using the assigned prefix of the java library, followed by the name of the custom tag’s java class. In addition, any additional parameters that the tag needs – such as a String or Integer – can be passed to the class in this command. Finally, the java code within the tag can be written. When put all together, using a custom tag will look something like this:

<body>

<%@ taglib uri=”/CustomTagName” prefix=”tagName” %>

<tagName:CustomTagName customAttributeOne=”value”>

</tagName:CustomTagName>

</body>

The above code implements a Java class, stored in a file called “CustomTagName.java”, which may look like this:

public class CustomTagName extends javax.servlet.tagext.BodyTagSupports (String input = customAttributeOne) {

//

}

One of the primary benefits of utilizing custom HTML tags is that it’s usually faster and easier for developers to reuse tags than it is for them to re-write similar scriplets across multiple .jsp pages. Custom tags are also a critical part of allowing .jsp pages and websites to be able to scale up. The usage of custom tags help .jsp pages become more modular and standardized, and helps keep .jsp applications stable as they grow larger and larger. In addition, creating a custom tag allows developers who are less experienced in Java to utilize Java code without needing experience in the language – something that cannot be done with scriplets. (Medium, 2025) Finally, using tag libraries instead of scriplets keep .jsp pages cleaner since less space is used for scriplets, making the pages easier to parse and work on for developers.

However, custom tags also introduce a new layer of complexity to a project, which can lead to a variety of possible complications and points of failure with their implementation. The connection between the .jsp page and the code library that it draws from is a layer of obfuscation; this means that it takes more time and effort to develop the pages that use a custom tag than it would take to simply use a one-time scriplet. A part of this effort comes from ensuring that the custom tag itself is well-written. Custom tags need to be flexible for all the possible situations that they can be called in, compared to individual scriplets that can be coded to be precisely what is needed for that specific circumstance. (GeeksforGeeks, 2025) And in general, using a custom tag and tag library requires a higher skill floor in Java, HTML, and general software development than scriplets do.

Overall, custom HTML tags are a critical feature of trying to create a large application with .jsp files. The ability to create reusable tags empowers developers to build larger applications than they would be able to otherwise. However, implementing custom tags does come with a non-trivial amount of challenges, such as increasing code obfuscation and the skill necessary. But these disadvantages are worth working around, as the benefits of creating a custom tag library are typically worth it, and doing so is standard practice when relying on .jsp pages.

# References

Pandey, S. (2025). Advanced Java — Custom Tags in JSP. Medium. Retrieved from https://medium.com/@SachinPandeyOnline/advanced-java-custom-tags-in-jsp-da3f324e8ee0

Custom Tags in JSP. (2025). GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/java/custom-tags-in-jsp/

Using Custom Tags. (2025). The J2EE 1.4 Tutorial. Retrieved from https://docs.oracle.com/javaee/1.4/tutorial/doc/JSPIntro9.html