Jason Schriner

Custom Tags

Module 10

7/20/25

Code reusability is essential for building scalable and maintainable applications. In Java Server Pages (JSP), one tool that supports this principle is the custom tag. According to Oracle (2025), “custom tags are user-defined elements that encapsulate recurring tasks,” allowing developers to streamline common functionality and reuse it across multiple JSP files. To understand the role of custom tags in JSP, it is important to explore how they are created and examine their advantages and disadvantages.

A custom tag allows developers to define an action in Java that can later be implemented in a JSP file through the use of a tag handler. The tag handler contains the logic associated with the tag and is executed when the JSP page is processed by the server. There are three components used in creating a custom tag. The first is the tag handler which works similarly to a Java class and contains the logic for the custom tag. The second component is a tag library or TLD (Tag Library Descriptor) file. This is an XML file that lists the custom tags, their attributes, and the tag handler classes they are linked to. The last component is the taglib directive. This imports the custom tag library into the JSP file so it can be used.

To create a custom tag, we can start by creating the tag handler class. It contains the logic that you wish the tag to perform. A tag handler is made by creating a class that extends the tag support class. Next, you need to define any variables that the tag may need. If we need to pass any values to the tag, we need to write a setter method for each variable. Finally, we create the doTag() method. This method is automatically called, so we need to override it. Inside the doTag() method is where we would place our custom tag logic that we wish to execute. After creating the tag handler class, the next step is to define the tag in a Tag Library Descriptor (TLD) file. A TLD file is an XML file that describes one or more custom tags and tells the JSP how to handle them. Each tag entry in the TLD maps a tag name to its corresponding tag handler class. It can also define attributes for the tag, such as whether they are required and whether they accept runtime expressions. The TLD file is usually stored in the WEB-INF directory of the project or in a dedicated tlds folder. Without the TLD, the JSP container would not recognize the custom tag or know which Java class to execute when the tag is used in a JSP page. Once the TLD file is created and properly configured, the final step is to use the custom tag in a JSP page. To do this, we add a taglib directive at the top of the JSP file. This works the same as a regular import statement and allows the developer to use the custom tag in the JSP file.

Custom tags are often used to replace scriptlets in JSP files. One of their main advantages is improved readability. Because the logic is contained in the tag handler class and not embedded directly in the JSP file, the code becomes cleaner, better organized, and more accessible especially for team members who may not be familiar with Java syntax. As GeeksforGeeks explains, “Custom tags are user‑defined action tags that can be used within Java Server Pages… it separates the business logic from JSP and helps avoid the use of scriptlet tags” (Dec, 2022). This highlights how custom tags support better separation of concerns. Scriptlets mix Java code with HTML, which can lead to messy and hard-to-maintain code. Custom tags solve this by isolating logic inside tag handlers, allowing the JSP page to focus solely on layout and content. This makes applications easier to maintain and allows developers and designers to work more independently. Finally, custom tags offer strong reusability. Once a tag is created, it can be reused across multiple JSP files without duplicating code. This not only saves time but also centralizes the logic, making future updates simpler and reducing the risk of bugs caused by inconsistent changes.

The disadvantage to custom tags is that there is a higher learning curve. Anyone with an understanding of Java can utilize scriplets. Learning how to create and use custom tags requires more time and experience. It also requires more setup. When writing a JSP file, you can instantly insert a scriplet. Where a custom tag requires you to create a tag and a library file before you can begin to utilize it. This extra set up can be a negative for small tasks that require frequent changes. Additionally custom tags require good documentation to ensure that anyone trying to maintain them later can easily understand their purpose.

I personally like the benefits of a custom tag. Code reusability and loose coupling are two major concerns in programing. In our project I used scriplets and they are much easier to use however I had to recode some simple things that could have easily been done with a custom tag and then my JSP files would have contained much less code but would have been easier to read and still accomplished the same task. The major drawback I have is the steeper learning curve of using a custom tag over a scriplet. If you are new to Jakarta and have any coding experience you can quickly build JSP files with scriplets. However, if you wished to use custom tags it would require additional time to learn. I believe the learning curve is worth it in the end and though I would like to take the easier path at times I believe this is where the real skill in programming comes in that it seems that the better option always is more difficult to learn.

Custom tags can greatly improve the structure and readability of JSP web pages. By moving logic out of the JSP and into separate Java classes, they help keep the presentation layer clean and easier to understand. Compared to scriptlets, custom tags offer better organization, reusability, and maintainability, especially in larger projects where consistent structure matters. While there may be a learning curve for developers who are new to JSP or Java, the long-term benefits are worth it. Once a developer becomes familiar with the setup process, custom tags can actually speed up development and reduce the chances of errors. Overall, custom tags are a smarter and more scalable solution for modern JSP development.

References:

Using Custom Tags. (2025). Oracle.com. <https://docs.oracle.com/javaee/1.4/tutorial/doc/JSPIntro9.html>

GeeksforGeeks. (2022, December 28). Custom Tags in JSP. GeeksforGeeks. <https://www.geeksforgeeks.org/java/custom-tags-in-jsp/>