**Creating Custom Tags**

Liz Hinz

Professor Darrell Payne

CSD430-A339: Server Side Development

July 20, 2025

**Creating Custom Tags**

Tags are utilized for a wide range of reasons, including physical and digital. Whether they hold information about a product, like a label in a store, or are included on social media platforms using hashtags, they are more important than one might realize. In software development, tags can help developers and users organize information. Custom tags have a history, advantages and disadvantages, requirements to develop one correctly, proper times to utilize them, and when they should not be used.

Java or Jakarta Server Pages (JSP) are utilized to craft dynamic web content without depending on a platform (Oracle, n.d.). Within JSP exists the Standard Tag Library to help bring increased practicality to web-based applications (Oracle, n.d.). JSP has the ability to implement custom tags. In JSP, a custom tag is assigned a tag handler that separates business logic, which means scriptlet tags do not have to be implemented (GeeksforGeeks, 2022). Avoiding scriptlet tags makes the JSP simpler to comprehend (GeeksforGeeks, 2022). Using custom tags instead of scriptlets has additional benefits, such as improved readability, reusability, and front-end development can implement them (GeeksforGeeks, 2022).

Requirements exist that are needed to correctly develop a custom tag. A custom tag needs a "tag handler, tag library descriptor file (TLD), and a Taglib directive in the JSP file" (GeeksforGeeks, 2022). A tag handler is an object that the JSP container establishes during runtime (GeeksforGeeks, 2022). Within a tag handler, there is a pageContext property with the tag location and a parent tag property extending from a Java class or interface (GeeksforGeeks, 2022). After the JSP container creates a tag handler object, the setter and getter Parent methods are called from pageContext (GeeksforGeeks, 2022). The JSP container manages over tag handlers in case changes need to be made (GeeksforGeeks, 2022). The TLD holds tags aligned with tag handlers and description and is implemented in a file using a root <taglib> or a sub of <tag> (GeeksforGeeks, 2022). The taglib holds the library version for the tag and JSP and a URI (GeeksforGeeks, 2022). The tag contains the body, a distinct name, a tag handler class, and an attribute (GeeksforGeeks, 2022). The attribute in a tag contains a distinct name, uses a boolean to decide if it is required, and sets whether a "runtime expression" can be added to the attribute value, the type, and whether it is a fragment (GeeksforGeeks, 2022). The taglib directive accesses the library in a JSP and "specifies the URI" and the prefix (GeeksforGeeks, 2022).

Custom tags in JSP come with many positives and drawbacks. Some advantages include the fact that tags are portable after being entered into a library, they are simple to implement, and they allow for many functionalities (DataFlair Team, 2020). Custom tags can also work with a wide range of scripting languages, are reusable, do not require a scriptlet tag, and separate business logic (DataFlair Team, 2020). As explained earlier, excluding scriptlets improves the code's readability and reusability, and allows front-end developers to implement custom tags (GeeksforGeeks, 2022). As great as custom tags in JSP can be, they come with some negatives that may deter developers. Some disadvantages include that they may not fit the developer's goal, mistakes in the process can lead to faulty or poorly performing tags, and an understanding of how tags work needs to exist.

Based on the information that was discovered throughout the process of this paper, many opinions regarding custom tags have been formed. Some of those include when to consider using them and when it is best to avoid them. Custom tags can be vital to the development process and should be considered when wanting a reusable element. They offer many benefits that can make them worth their creation. If no JSPs or any Java language is being used, then implementing custom tags might not make sense. The best practice is to gather with the development team to decide what features should and should not be included based on project requirements.

Custom tags are a multifunctional feature that can be implemented in software development. They offer unique features and identification marks that help reuse them in different areas, including front-end development. Once a general understanding of how custom tags work and how they can be implemented is gained, they can offer an even richer experience for all.

Here are the basic formats of creating custom tags provided by GeeksforGeeks (2022):

" Empty custom tag(without body):

*<prefix : suffix attribute = "value"/>*

Non-empty custom tag:

*<prefix : suffix attribute = "value"/> "*

**References**

DataFlair Team. (2020, September 2). *JSP Custom Tags with Examples*. DataFlair. https://data-flair.training/blogs/jsp-custom-tags/

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

Oracle. (n.d.). *JavaServer Pages Technology*. Www.oracle.com. https://www.oracle.com/java/technologies/jspt.html