Kyle Marlia-Conner

07/20/2025

Module 10

Creating Custom Tags

Like a hotkey, custom tags are user-defined actions that can quickly be called upon in a JSP. This allows developers to take commonly used functionalities into easy-to-understand, reusable tags. This helps promote cleaner code and reduces the amount of repetitive JSP scriptlets. By delegating logic to custom tags, developers can create applications that are easier to maintain, read, and collaborate on.

One of the primary advantages of custom tags is their code reusability. Tasks that occur frequently (such as formatting and validation) can be wrapped in reusable tag handlers, simplifying code across multiple JSP pages. This results in a more organized codebase and accelerates the development process. Developers no longer need to duplicate scriptlets across files, which helps reduce human errors.

Another advantage these hold is their modularity. By encapsulating logic within tag handler classes, custom tags support a modular structure that isolates functionality. This means that any changes to the internal behavior of a tag can be made without touching the JSP files that use it. According to Oracle (2008), this modular design aligns with Model-View-Controller (MVC) principles, which emphasize the separation of data handling (model), business logic (controller), and presentation (view) layers. The modularity of custom tags thus contributes directly to cleaner and more maintainable code.

Additionally, custom tags help facilitate collaboration between frontend and backend developers. For frontend developers with limited Java experience, working with JSP pages filled with scriptlets can be confusing. Server2Client (2025) explains that custom tags hide backend logic behind simple, readable syntax, enabling developers to use tags like <loop> instead of embedding Java logic directly into the page. This makes the JSP files more accessible to team members with different skill sets and promotes smoother collaboration across teams.

Despite their advantages, custom tags have a learning curve and initial complexity. Developers must understand tag handler classes, lifecycle methods, and how to configure Tag Library Descriptor (TLD) files properly. This alone can be a barrier for new developers or teams unfamiliar with JSP tag libraries. With smaller projects, using JSP scriptlets may be a more practical option due to this.

To implement custom tags, developers must first define a tag handler class. These classes typically extend predefined Java classes such as TagSupport, BodyTagSupport, or SimpleTagSupport, or implement interfaces like Tag, IterationTag, or SimpleTag. The next step is to create a Tag Library Descriptor (TLD) file. This XML configuration file contains metadata for each tag, including its name, associated handler class, attributes, and allowed body content. A properly structured TLD file ensures that tags behave correctly and helps prevent runtime issues due to misconfiguration. Finally, developers must register the custom tag library in each JSP file using the <%@ taglib %> directive. This directive links the JSP file to the TLD, allowing the JSP container to identify and process custom tags during page rendering.

Additionally, a minor performance overhead is introduced by using custom tags. While the performance impact is negligible in most modern web environments, Oracle (2008) notes that it can become a concern in performance-critical scenarios (such as large-scale, high-traffic systems) where even small inefficiencies can accumulate significantly. Because of this, developers should educate themselves and weigh the trade-offs based on requirements.

In my research (and a little bit of tinkering), custom tags seem almost like a necessity for medium-to-large applications where long-term maintainability and team collaboration are important. The knowledge gained in learning and using custom tags is more than justified, though, when thinking about the time it can save and the code clutter that disappears, making it more approachable for others. Because of this, custom tags are a powerful feature of JSP that support clean architecture, encourage reusability, and streamline team development. While they introduce complexity and a learning curve, their advantages make them a valuable tool for creating maintainable web applications.

References:

*Programming weblogic JSP tag extensions*. Understanding and Creating Custom JSP Tags. (2008a, May 23). https://docs.oracle.com/cd/E11035\_01/wls100/pdf/taglib.pdf

*Using Custom Tags*. Using custom tags. (n.d.). https://docs.oracle.com/javaee/1.4/tutorial/doc/JSPIntro9.html

GeeksforGeeks. (2025b, April 28). *Custom tags in JSP*. https://www.geeksforgeeks.org/java/custom-tags-in-jsp/

*Creating our own custom Tags*. LJS. (n.d.). https://server2client.com/jstl/createcustomtags.html

Natarajan, R. (2023, October 2). *#12 How to Create Own Tags in JSP? | Custom Tags | Tag Handler Class | TLD File | doTag() | Taglib*. YouTube. https://www.youtube.com/watch?v=tox6mq8e8KA