Carmen Mendoza

CSD 430

07/17/2025

Module 10

Creating Custom Tags

In web development, code reusability and modularization are necessary for improving efficiency and maintainability. Developers address this by using custom tags. Custom tags are a way to make your HTML or JavaServer Pages (JSP) cleaner and more reusable. Instead of repeating the same code repeatedly, custom tags let you write it once and use it anywhere you may need it. Custom tags can be created in different environments such as HTML5, Java (JSP), or ASP.NET. In this paper I will discuss what I have learned about custom tags, their pros and cons, how to create custom tags, and if it is necessary to use them.

One of the biggest advantages of custom tags is reusability. Say you have a component like a button, a styled alert box, or a table that appears on multiple pages; you can turn it into a custom tag and use it multiple times with just a one-line code. Essentially, this saves time and keeps the code clean and organized.

Custom tags also help with separating logic from presentation. For example, in JSP, instead of writing Java Code inside the HTML, you can move that logic into a custom tag. In doing so, this makes the HTML file easier to read and manage, especially if there are multiple members on the team that are handling front-end design. Another benefit is readability. A long code that builds a complex layout can be made into a single custom tag that will state its purpose, such as <user-profile />. This way when another team member is working on the project, they can tell what it does just by reading the tag.

Now while custom tags have a variety of benefits, they are not perfect. One of its disadvantages is the setup time. You cannot create custom tags instantly. When using JSP for example, you would have to write a tag handler, register the tag in an XML file, and ensure that it is connected properly in the JSP file. This can be more time consuming than just writing a few lines of HTML or Java code.

Another disadvantage is its learning curve. If you have someone new to the team and they glance at your code and see for example “<my-alert>,” they will not necessarily know what it does unless they dive into the custom tag code. Another time-consuming step can also occur if the team does not have good documentation; this can possibly slow down development.

One other disadvantage is performance concerns. If a tag is doing a lot of behind-the-scenes work and you use it multiple times on a page, it could slow things down. So, keep in mind that you will need to be careful with how much logic is included in a single tag.

When creating a tag, it varies depending on what technology you use. In JSP you would do the following: Create a tag handler class in Java first, then write a TLD file to define the tag’s name and attributes. Next, you would import the tag library in your JSP file and use the tag. Below I will show a JSP Page that shows the current date. <%@ taglib uri="http://example.com/mytags" prefix="custom" %>

<custom:showDate format="yyyy-MM-dd" />. Here is an example of a Java Tag Handler Class: public class ShowDateTag extends SimpleTagSupport {

private String format;

public void setFormat(String format) {

this.format = format;

}

public void doTag() throws JspException, IOException {

SimpleDateFormat sdf = new SimpleDateFormat(format);

JspWriter out = getJspContext().getOut();

out.print(sdf.format(new Date()));

}

}

I personally think using custom tags can be helpful, especially in big projects. Codes are easier to manage and can help you avoid repeating yourself. That said, you do not always need custom tags; it varies depending on the situation. Say for example, if a feature is only used once or is simple, making a custom tag might not be worth the extra effort. I believe they can be a great tool if they are used wisely. They can help to reduce errors, help with organization, and can make a project easier to scale.

Custom tags are great for building clean, modular projects, and reusable code in web development. Some of its advantages are readability, reusability, and a clean separation of login and presentation. But it can be tricky to set up and understand if it is not well documented. After researching custom tags, they are a smart solution for large or complex web applications.

References:

(n.d.). *Custom Tags in JSP*. Geeks for Geeks. <https://www.geeksforgeeks.org/java/custom-tags-in-jsp/> c

(n.d.). *Chapter 8 Custom Tags in JSP Pages*. Oracle. <https://docs.oracle.com/cd/E19316-01/819-3669/bnalj/index.html>

(n.d.). *17.12 Asynchronous Processing*. Oracle. <https://docs.oracle.com/javaee/7/tutorial/servlets012.htm>

(n.d.). *Using custom elements*. Modzilla. <https://developer.mozilla.org/en-US/docs/Web/API/Web_components/Using_custom_elements>

Anderson, R. (n.d.). *Tag Helpers in ASP.NET Core*. Microsoft. https://learn.microsoft.com/en-us/aspnet/core/mvc/views/tag-helpers/intro?view=aspnetcore-9.0