Creating a **custom tag in JSP** involves several steps using the JSP API. Below is a **complete guide with explanation and full working code**.

**✅ Overview**

Custom tags are reusable components in JSP that encapsulate logic in Java classes and can be reused across multiple pages.

**✅ Step-by-Step Guide**

**1. Create a Tag Handler Class**

Use SimpleTagSupport (part of javax.servlet.jsp.tagext) to define your custom logic.

**2. Define TLD (Tag Library Descriptor) File**

Tells the JSP engine about your tag (name, class, attributes, etc.).

**3. Configure in JSP**

Use the taglib directive and call your custom tag.

**📁 Project Structure**

pgsql

CopyEdit

MyCustomTagApp/

│

├── WebContent/

│ ├── index.jsp

│ ├── WEB-INF/

│ │ ├── custom.tld

│ │ └── web.xml

│

└── src/

└── com/customtags/

└── HelloTag.java

**1️⃣ Java Code: HelloTag.java**

package com.customtags;

import java.io.IOException;

import javax.servlet.jsp.JspException;

import javax.servlet.jsp.tagext.SimpleTagSupport;

import javax.servlet.jsp.JspWriter;

public class HelloTag extends SimpleTagSupport {

private String name;

// Setter for attribute (must match TLD)

public void setName(String name) {

this.name = name;

}

@Override

public void doTag() throws JspException, IOException {

JspWriter out = getJspContext().getOut();

if (name != null && !name.trim().isEmpty()) {

out.println("Hello, " + name + "! Welcome to custom JSP tag.");

} else {

out.println("Hello, guest! Welcome to custom JSP tag.");

}

}

}

**2️⃣ TLD File: custom.tld**

Place this inside WEB-INF/custom.tld

<?xml version="1.0" encoding="UTF-8" ?>

<taglib xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-jsptaglibrary\_2\_1.xsd"

version="2.1">

<tlib-version>1.0</tlib-version>

<short-name>CustomTag</short-name>

<uri>http://customtags</uri>

<tag>

<name>hello</name>

<tag-class>com.customtags.HelloTag</tag-class>

<body-content>empty</body-content>

<attribute>

<name>name</name>

<required>false</required>

<rtexprvalue>true</rtexprvalue>

</attribute>

</tag>

</taglib>

**3️⃣ JSP File: index.jsp**

<%@ taglib prefix="ct" uri="http://customtags" %>

<html>

<head><title>Custom Tag Example</title></head>

<body>

<h2>Using Custom JSP Tag:</h2>

<ct:hello name="Manjiri" />

<br/>

<ct:hello />

</body>

</html>

**4️⃣ web.xml (Optional for JSP 2.1+ but good practice)**

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee" version="3.1">

<jsp-config>

<taglib>

<taglib-uri>http://customtags</taglib-uri>

<taglib-location>/WEB-INF/custom.tld</taglib-location>

</taglib>

</jsp-config>

</web-app>

**✅ Output**

When index.jsp is accessed:

Hello, Manjiri! Welcome to custom JSP tag.

Hello, guest! Welcome to custom JSP tag.

**📦 Required JARs**

These are provided by your servlet container (e.g., Tomcat):

* javax.servlet-api.jar
* javax.servlet.jsp-api.jar

Make sure they’re in your lib folder or classpath during compilation if you're not using a server IDE.