Here's a complete **JSP custom tag tutorial** using:

✅ TagSupport  
✅ BodyTagSupport

We’ll cover these cases:

| **Case** | **Tag Class** | **Attributes** | **Body Content** |
| --- | --- | --- | --- |
| 1. Simple tag | TagSupport | ❌ | ❌ |
| 2. Tag with attribute | TagSupport | ✅ | ❌ |
| 3. Tag with body | BodyTagSupport | ❌ | ✅ |
| 4. Tag with attr + body | BodyTagSupport | ✅ | ✅ |

**🗂 Directory Structure**

CustomTagSupportExample/

├── src/

│ └── com/customtags/

│ ├── SimpleTag.java

│ ├── AttributeTag.java

│ ├── BodyOnlyTag.java

│ └── AttributeBodyTag.java

├── WebContent/

│ ├── index.jsp

│ └── WEB-INF/

│ ├── custom.tld

│ └── web.xml

**✅ 1. Simple Tag (No Attribute, No Body) – TagSupport**

package com.customtags;

import java.io.IOException;

import javax.servlet.jsp.tagext.TagSupport;

import javax.servlet.jsp.JspException;

public class SimpleTag extends TagSupport {

@Override

public int doStartTag() throws JspException {

try {

pageContext.getOut().println("Hello from Simple Tag (TagSupport)");

} catch (IOException e) {

throw new JspException("Error in SimpleTag", e);

}

return SKIP\_BODY;

}

}

**✅ 2. Tag with Attribute – TagSupport**

package com.customtags;

import java.io.IOException;

import javax.servlet.jsp.tagext.TagSupport;

import javax.servlet.jsp.JspException;

public class AttributeTag extends TagSupport {

private String message;

public void setMessage(String message) {

this.message = message;

}

@Override

public int doStartTag() throws JspException {

try {

pageContext.getOut().println("Attribute Message: " + message);

} catch (IOException e) {

throw new JspException("Error in AttributeTag", e);

}

return SKIP\_BODY;

}

}

**✅ 3. Tag with Body – BodyTagSupport**

package com.customtags;

import java.io.IOException;

import javax.servlet.jsp.tagext.BodyTagSupport;

import javax.servlet.jsp.JspException;

public class BodyOnlyTag extends BodyTagSupport {

@Override

public int doEndTag() throws JspException {

try {

String content = bodyContent.getString().trim();

pageContext.getOut().println("Body Content: " + content);

} catch (IOException e) {

throw new JspException("Error in BodyOnlyTag", e);

}

return EVAL\_PAGE;

}

}

**✅ 4. Tag with Attribute + Body – BodyTagSupport**

package com.customtags;

import java.io.IOException;

import javax.servlet.jsp.tagext.BodyTagSupport;

import javax.servlet.jsp.JspException;

public class AttributeBodyTag extends BodyTagSupport {

private String prefix;

public void setPrefix(String prefix) {

this.prefix = prefix;

}

@Override

public int doEndTag() throws JspException {

try {

String content = bodyContent.getString().trim();

pageContext.getOut().println(prefix + ": " + content);

} catch (IOException e) {

throw new JspException("Error in AttributeBodyTag", e);

}

return EVAL\_PAGE;

}

}

**📘 TLD File: 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>CustomTagSupport</short-name>

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

<tag>

<name>simple</name>

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

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

</tag>

<tag>

<name>attribute</name>

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

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

<attribute>

<name>message</name>

<required>true</required>

<rtexprvalue>true</rtexprvalue>

</attribute>

</tag>

<tag>

<name>bodyOnly</name>

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

<body-content>scriptless</body-content>

</tag>

<tag>

<name>attrBody</name>

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

<body-content>scriptless</body-content>

<attribute>

<name>prefix</name>

<required>true</required>

<rtexprvalue>true</rtexprvalue>

</attribute>

</tag>

</taglib>

Explaination for

<tag>

<name>attribute</name>

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

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

<attribute>

<name>message</name>

<required>true</required>

<rtexprvalue>true</rtexprvalue>

</attribute>

</tag>

This is a configuration snippet from a **Tag Library Descriptor (TLD)** file used to define a **custom JSP tag**. It describes a tag named <attribute> and provides metadata required by the JSP container to process and validate the tag. Let's break it down line-by-line:

**🔧 Full Breakdown:**

<tag>

* Begins the definition of a custom tag.

<name>attribute</name>

* This is the **name of the tag** as used in a JSP file:

<mytags:attribute message="Hello World" />

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

* This is the **fully qualified Java class** that implements the tag logic (likely extends TagSupport or SimpleTagSupport).

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

* This indicates the tag does **not allow any body content** (i.e., it must be self-closing or cannot have inner text).

<attribute>

<name>message</name>

<required>true</required>

<rtexprvalue>true</rtexprvalue>

</attribute>

This block defines a **single attribute** (message) for the tag:

| **Sub-element** | **Description** |
| --- | --- |
| <name> | Name of the attribute (in this case, message) |
| <required> | This attribute must be specified by the JSP user |
| <rtexprvalue> | **Runtime Expression Value allowed**: the attribute can accept values like ${...} or <%= ... %> |

For example, this is valid:

<mytags:attribute message="${user.name}" />

</tag>

* Ends the definition of this custom tag.

When you set:

<body-content>scriptless</body-content>

in a **Tag Library Descriptor (TLD)** file, you're telling the JSP container the following:

**🔍 Meaning of scriptless:**

* **Allows body content**, but the body **cannot contain scriptlets**.
* Scriptlets refer to the old-style JSP code like:

<% int x = 5; %>

* However, **Expression Language (EL)** and **custom tags** are allowed:

Hello, ${user.name}!

<c:forEach ...>...</c:forEach>

**✅ Valid body content with scriptless:**

<mytags:greeting>

Hello, ${user.name}!

</mytags:greeting>

**❌ Invalid (will cause an error):**

<mytags:greeting>

<% int x = 10; %>

</mytags:greeting>

**📝 JSP File: index.jsp**

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

<html>

<head><title>TagSupport and BodyTagSupport Examples</title></head>

<body>

<h2>1. Simple Tag</h2>

<ct:simple />

<h2>2. Tag with Attribute</h2>

<ct:attribute message="This is an attribute value" />

<h2>3. Tag with Body</h2>

<ct:bodyOnly>

This is inside the tag body only.

</ct:bodyOnly>

<h2>4. Tag with Attribute and Body</h2>

<ct:attrBody prefix="Note">

This is a tagged message with prefix.

</ct:attrBody>

</body>

</html>

**🛠 web.xml (Optional for modern servers)**

<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 in Browser**

1. Simple Tag

Hello from Simple Tag (TagSupport)

2. Tag with Attribute

Attribute Message: This is an attribute value

3. Tag with Body

Body Content: This is inside the tag body only.

4. Tag with Attribute and Body

Note: This is a tagged message with prefix.