

JSP Directives – page, include and taglib example

JSP Directives are used to give special instruction to container for translation of JSP to Servlet code. JSP Directives are placed between `<%@ %>`.

JSP provides three directives for us to use;

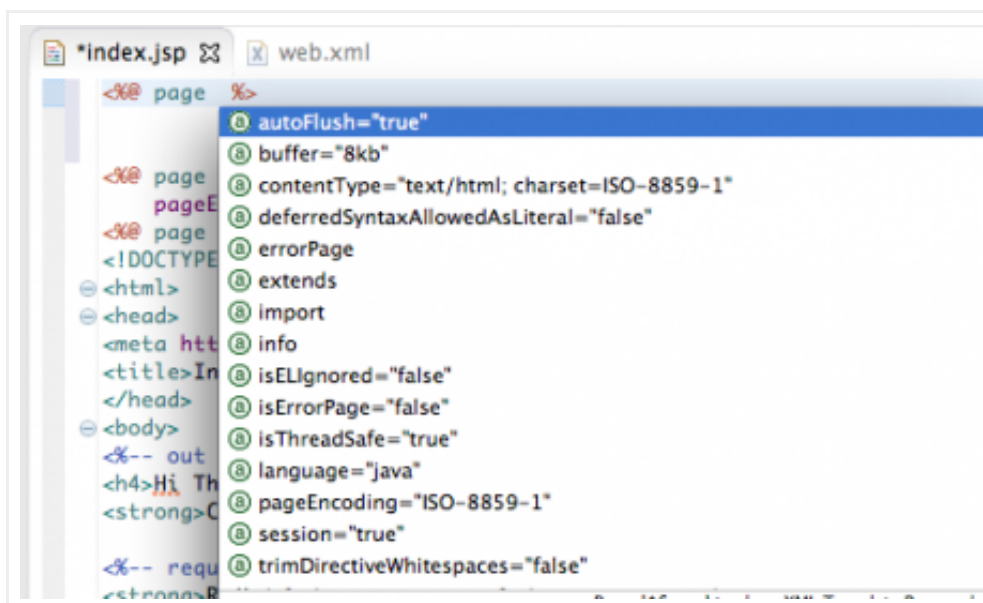
- page directive
- include directive
- taglib directive

Every directive has a set of attributes to provide specific type of instructions. So usually a JSP directive will look like `<%@ directive attribute="value" %>`.

If you have directly landed here, you might want to check out [JSP Tutorial for Beginners](#) and [JSP Implicit Objects](#).

JSP page directive

page directive provide attributes that gets applied to the entire JSP page. page directive has a lot of attributes that we will look now. We can define multiple attributes in a single page directive or we can have multiple page directives in a single JSP page.



1. **import attribute:** This is one of the most used page directive attribute. It's used to instruct container to import other java classes, interfaces, enums etc. while generating servlet code. This is similar to import statements in java classes, interfaces. An example of import page directive usage is:

```
<%@ page import="java.util.Date,java.util.List,java.io.*" %>
```

2. **contentType attribute:** This attribute is used to set the content type and character set of the response. The default value of contentType attribute is "text/html; charset=ISO-8859-1". We can use it like below.

```
<%@ page contentType="text/html; charset=US-ASCII" %>
```

3. **pageEncoding attribute:** We can set response encoding type with this page directive attribute, its default value is "ISO-8859-1".

```
<%@ page pageEncoding="US-ASCII" %>
```

4. **extends attribute:** This attribute is used to define the super class of the generated servlet code. This is very rarely used and we can use it if we have extended HttpServlet and overridden some of its implementations. For example;

```
<%@ page extends="org.apache.jasper.runtime.HttpJspBase" %>
```

5. **info attribute:** We can use this attribute to set the servlet description and we can retrieve it using Servlet interface getServletInfo() method. For example;

```
<%@ page info="Home Page JSP" %>
```

6. **buffer attribute:** We know that JspWriter has buffering capabilities, we can use this attribute to set the buffer size in KB to handle output generated by JSP page. Default value of buffer attribute is 8kb. We can define 16 KB buffer size as;

```
<%@ page buffer="16kb" %>
```

7. **language attribute:** language attribute is added to specify the scripting language used in JSP page. Its default value is "java" and this is the only value it can have. May be in future, JSPs provide support to include other scripting languages like C++ or PHP too.

```
<%@ page language="java" %>
```

8. **isELIgnored attribute:** We can ignore the Expression Language (EL) in JSP using this page directive attribute. Its datatype is **Java Enum** and default value is *false*, so EL is enabled by default. We can instruct container to ignore EL using below directive;

```
<%@ page isELIgnored="true" %>
```

9. **isThreadSafe attribute:** We can use this attribute to implement SingleThreadModel interface in generated servlet. Its an Enum with default value as true. If we set its value to false, the generated servlet will implement SingleThreadModel and eventually we will lose all the benefits of servlet **multi-threading** features. You should never set its value to false.

```
<%@ page isThreadSafe="false" %>
```

10. **errorPage attribute:** This attribute is used to set the error page for the JSP, if the JSP throws exception, the request is redirected to the error handler defined in this attribute. Its datatype

is URI. For example;

```
<%@ page errorHandler="errorHandler.jsp" %>
```

11. **isErrorPage attribute:** This attribute is used to declare that current JSP page is an error page. It's of type Enum and default value is false. If we are creating an error handler JSP page for our application, we have to use this attribute to let container know that it's an error page. [JSP implicit attribute](#) exception is available only to the error page JSPs. For example;

```
<%@ page isErrorPage="true" %>
```

12. **autoFlush attribute:** autoFlush attribute is to control the buffer output. Its default value is true and output is flushed automatically when buffer is full. If we set it to false, the buffer will not be flushed automatically and if it's full, we will get exception for buffer overflow. We can use this attribute when we want to make sure that JSP response is sent in full or none. For example;

```
<%@ page autoFlush="false" %>
```

13. **session attribute:** By default JSP page creates a session but sometimes we don't need session in JSP page. We can use this attribute to indicate compiler to not create session by default. Its default value is true and session is created. To disable the session creation, we can use it like below.

```
<%@ page session ="false" %>
```

14. **trimDirectiveWhitespaces attribute:** This attribute was added in JSP 2.1 and used to strip out extra white spaces from JSP page output. Its default value is false. It helps in reducing the generated code size, notice the generate servlet code keeping this attribute value as true and false. You will notice no `out.write("\n")` when it's true.

```
<%@ page trimDirectiveWhitespaces ="true" %>
```

JSP include directive

JSP include directive is used to include the contents of another file to the current JSP page. The included file can be HTML, JSP, text files etc. Include directive is very helpful in creating templates for user views and break the pages into header, footer, sidebar sections.

We can include any resource in the JSP page like below.

```
<%@ include file="test.html" %>
```

The file attribute value should be the relative URI of the resource from the current JSP page.

JSP taglib directive

JSP taglib directive is used to define a tag library with prefix that we can use in JSP, we will look into

more details in JSP Custom Tags tutorial.

We can define JSP tag libraries in like below;

```
<%@ taglib uri="/WEB-INF/c.tld" prefix="c"%>
```

JSP taglib directive is used in JSP standard tag libraries, please read [JSTL Tutorial](#).

Thats all for JSP directives, we will look into Expression Language (EL) and JSP Actions next.

Update: [JSP EL](#) and [JSP Action Tags](#) articles are live now.
