LEARNING jstl

#jstl

Table of Contents

About

Chapter 1: Getting started with jstl

Remarks

JSTL (JSP Standard Tag Library) is a JSP based standard tag library which offers `<c:xxx>` tags to control the flow in the JSP page, `<fmt:xxx>` tags for date/number formatting and internationalization facilities and several `${fn:xxx()}` utility EL functions.

Note that JSTL also offers SQL and XML taglibs which enable a declarative manner of executing SQL queries and parsing XML inside a JSP page. This is however discouraged for other purposes than quick prototyping. In the real world both tasks need to be done by real Java classes which are (in)directly controlled/delegated by a Servlet.

Examples

Installation

JSTL is part of the Java EE API and included in Java EE application servers such as WildFly, TomEE, GlassFish, but not in barebones servletcontainers such as Tomcat and Jetty. JSTL are the taglibs which you import from `http://java.sun.com/jsp/jstl/*` namespace. JSTL must not be confused with a "custom JSP tag library" (wherein you define a `.tld` file *yourself*). JSTL must also not be confused with taglibs of 3rd party frameworks such as JSF, Spring MVC, Struts, Displaytag, etcetera. JSTL must also not be confused with Expression Language (EL) (which are those `${}` things).

1. *Only* when your servletcontainer doesn't ship with JSTL builtin (e.g. Tomcat and Jetty), then just drop the jstl-1.2.jar straight in webapp's `/WEB-INF/lib` folder (which is covered by the default webapp's classpath, so in a bit smart IDE you don't need to do anything else). For starters, do **not** fiddle around in IDE project's *Build Path* setting. This is Wrong.

   In case you're using Maven, this is the coordinate:

   ```
   <dependency>
       <groupId>javax.servlet</groupId>
       <artifactId>jstl</artifactId>
       <version>1.2</version>
   </dependency>
   ```

   This is by the way the JSTL API bundled with Apache's JSTL implementation in a single JAR flavor. This does **not** require the `standard.jar` (it's for JSTL 1.1 only). Note that there's also a `jstl:jstl` dependency, but it's exactly the same file, only with a wrong group ID. Further there's also a `javax.servlet.jsp.jstl:jstl` dependency, but it is empty.

2. Declare the taglib in JSP file with the right TLD URI. You can find here the TLD documentation that applies to both JSTL 1.1 and JSTL 1.2. Click the taglib of interest to get the declaration examples. For example the JSTL core taglib

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

If you're using Facelets or JSPX instead of JSP, it should be declared as XML namespace instead

```
<anyxmlelement xmlns:c="http://java.sun.com/jsp/jstl/core">
```

You only need to ensure that you have no duplicates of older JSTL versions in the classpath (includes JDK/JRE's `/lib` and server's `/lib`) to avoid collisions. If you have full admin-level control over the server, then you could also place the JAR file in server's `/lib` instead of webapp's `/WEB-INF/lib` so that they get applied to all deployed webapps. At least do NOT extract the JAR file(s) and clutter the classpath with their contents (the loose TLD files) and/or declare the taglibs in *your* webapp's `web.xml` as some poor online tutorials suggest.

Read Getting started with jstl online: https://riptutorial.com/jstl/topic/2885/getting-started-with-jstl

Credits

S. No	Chapters	Contributors
1	Getting started with jstl	BalusC, Community