

Can not find the tag library descriptor for "http://java.sun.com/jsp/jstl/core"

stackoverflow.com/questions/13285826/can-not-find-the-tag-library-descriptor-for-http-java-sun-com-jsp-jstl-core

Ask Question

I have included this at the very top of my JSP page:

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

I already placed the JSTL JAR file in the `WEB-INF/lib` directory. But still, the JSP can't resolve the taglib. I get the below error:

Can not find the tag library descriptor for "<http://java.sun.com/jsp/jstl/core>"

I am using Eclipse Juno and the project structure is shown below:

asked Nov 8 '12 at 9:10



[user962206](#)

4,98751145231



11 Answers

[up vote](#) 86 [down vote](#) [accepted](#)

Can not find the tag library descriptor for "<http://java.sun.com/jsp/jstl/core>"

Based on [one of your previous questions](#) you're using Tomcat 7. In that case you need JSTL 1.2. However, you've there a `jstl.jar` file while JSTL 1.2 has clearly the version number included like so `jstl-1.2.jar`. The sole filename `jstl.jar` is typical for JSTL 1.0 and 1.1. This version requires a `standard.jar` along in `/WEB-INF/lib` which contains the necessary TLD files. However, in your particular case the `standard.jar` is clearly missing in `/WEB-INF/lib` and that's exactly the reason why the taglib URI couldn't be resolved.

To solve this you must remove the wrong JAR file, download [jstl-1.2.jar](#) and drop it in its entirety in `/WEB-INF/lib`. That's all. You do not need to extract it nor to fiddle in project's *Build Path*.

Don't forget to remove that loose `c.tld` file too. It absolutely doesn't belong there. This is indeed instructed in some poor tutorials or answers elsewhere in the Internet. This is a myth caused by a major misunderstanding and misconfiguration. There is never a need to have a

loose JSTL TLD file in the classpath, also not in previous JSTL versions.

In case you're using Maven, use the below coordinate:

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

You should also make sure that your `web.xml` is declared conform *at least* Servlet 2.4 and thus not as Servlet 2.3 or older. Otherwise EL expressions inside JSTL tags would in turn fail to work. Pick the highest version matching your target container and make sure that you don't have a `<!DOCTYPE>` anywhere in your `web.xml`. Here's a Servlet 3.0 (Tomcat 7) compatible example:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
  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-app_3_0.xsd"
  version="3.0">

  <!-- Config here. -->

</web-app>
```

See also:

answered Nov 9 '12 at 12:02



[BalusC](#)

836k29431023185

[up vote](#) [8 down vote](#)

The URI depends on the version of JSTL you are using. For Version 1.0 use:

`http://java.sun.com/jstl/core`

and for 1.1 (and later), you need to use:

`http://java.sun.com/jsp/jstl/core`

answered Nov 8 '12 at 9:13



[nfechner](#)

14.8k43860

up vote 5 down vote

I had same problem and despite having jstl

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

I had to add 'standard' as well:

```
<dependency>
  <groupId>taglibs</groupId>
  <artifactId>standard</artifactId>
  <version>1.1.2</version>
</dependency>
```

Also, as mentioned in previous post:

- for version 1.0 use: <http://java.sun.com/jstl/core>
- for 1.1 (and later) use: <http://java.sun.com/jsp/jstl/core>

answered Apr 18 '17 at 19:05



Nenad Bulatovic

4,48565578

up vote 0 down vote

If you are using maven your pom's `<dependencies>` tag should look like:

```
<dependencies>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
  </dependency>
  <dependency>
    <groupId>javax.servlet.jsp.jstl</groupId>
    <artifactId>javax.servlet.jsp.jstl-api</artifactId>
    <version>1.2.1</version>
  </dependency>
  <dependency>
    <groupId>taglibs</groupId>
    <artifactId>standard</artifactId>
    <version>1.1.2</version>
  </dependency>
</dependencies>
```

It works for me.

answered Dec 10 '15 at 13:52



Anakii Eugene

6510

up vote 0 down vote

Make sure that both the below jars are present in the build path:

1. standard-1.1.2.jar
2. jstl-api-1.2.jar

Maven Dependencies are:

```
<dependency>
  <groupId>javax.servlet.jsp.jstl</groupId>
  <artifactId>jstl-api</artifactId>
  <version>$1.2</version>
</dependency>
```

```
<dependency>
  <groupId>taglibs</groupId>
  <artifactId>standard</artifactId>
  <version>1.1.2</version>
</dependency>
```

answered Feb 16 at 6:25



Anshul Sood

113

up vote 0 down vote

you need to configure this in web.xml as well. Please refer below code.

```
<taglib>
  <taglib-uri>http://java.sun.com/jsp/jstl/core</taglib-uri>
  <taglib-location>/WEB-INF/lib/c.tld</taglib-location>
</taglib>
```

Please let me know if you still face any issue.

answered Nov 8 '12 at 9:50



Gautam

1,93021944

up vote 0 down vote

To make it work:

Add **jstl** and **standard** jar files to your library.

Hope it helps.. :)

answered May 4 '17 at 8:24



Sarat Chandra

1,9221216

up vote 0 down vote

Well, I have read this post and seems it is not good enough to fix this issue.

Spring based application on the tomcat 8 will not work.

Here is the solution

Step 1 ⇒ Add the following dependency in your pom.xml

```
<!-- JSTL Support -->

    <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
</dependency>
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.1</version>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```

Step 2 ⇒ Add All following two statement in your JSP page, ensure that you are using isELIgnored="false" attribute in <%@ page %> tag

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
isELIgnored="false" pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
```

Step 3 ⇒ Remove all the other configuration that you have done till now in web.xml or anywhere else :)

Step 4 ⇒ Clean the Tomcat and restart Tomcat.

Side Note ⇒ Actually, JSTL will work with only 3.x Servlet specifications on Tomcat 8.

[<http://www.kriblog.com/j2ee/jsp/jstl-with-spring-4-on-tomcat-8.html>]

answered Feb 21 '16 at 15:06



[user1788142](#)

411

[up vote](#) 0 [down vote](#)

Downloading jstl-1.2.jar instead of jstl.jar solves the problem

answered Jun 6 at 5:09



[shikha singh](#)

494

[up vote](#) 0 [down vote](#)

add the maven dependencies:

```
<dependency>
  <groupId>jstl</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>>taglibs</groupId>
  <artifactId>standard</artifactId>
  <version>1.1.2</version>
  <scope>compile</scope>
</dependency>
```

answered Jul 11 '17 at 12:17



[Dev M](#)

51421232

[up vote](#) 0 [down vote](#)

Simply download **javax.servlet.jsp.jstl.jar** and add to your build path and **WEB-INF/lib** if you simply developing dynamic web application.

When you develop dynamic web application using maven then add **javax.servlet.jsp.jstl** dependency in pom file.

Thanks

nirmalrajsanjeev

answered Nov 30 '17 at 4:50



Sanjeev Kumar

294

Your Answer

- [Links](#)
- [Images](#)
- [Styling/Headers](#)
- [Lists](#)
- [Blockquotes](#)
- [Code](#)
- [HTML](#)
- [advanced help »](#)

Not the answer you're looking for? Browse other questions tagged [java](#) [eclipse](#) [tomcat](#) [jstl](#) or ask your own question.
