

Title: How to fix JAVAX runtime error on JDK11 Tomcat9 Spring Application

Post Body:

The specific error I am facing when starting my app on Tomcat9. Spring version: 5.1.5.RELEASE:

```
SEVERE: Error configuring application listener of class  
[org.springframework.web.context.request.RequestContextListener] java.lang.NoClassDefFoundError:  
javax.xml.ws.WebServiceRef
```

There are multiple answers on this error that all suggest adding maven dependencies. I have added these dependencies:

My build path:

The `WebServiceRef` class is found in the package explorer:

This error shows in the console when starting the app on Tomcat9. Here is more of the stack trace:

```
SEVERE: Error configuring application listener of class  
[org.springframework.web.context.request.RequestContextListener] java.lang.NoClassDefFoundError:  
javax.xml.ws.WebServiceRef at  
org.apache.catalina.core.DefaultInstanceManager.populateAnnotationsCache(DefaultInstanceManager.java:303) ...more  
stuff... java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at  
java.base/java.lang.reflect.Method.invoke(Method.java:566) at  
org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:355) at  
org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:495)
```

Any help or suggestions would be much appreciated!

Update 1: Here are some .jar files that get copied into the final .war build under WEB-INF/lib. The 4 new maven dependencies and their versions are here, but there is also possible duplicate .jar files.

Update 2: Copying the `jaws-api` jar directly into my Tomcat Classpath resolved the runtime error... So the app is launching now, but how would I overcome this issue when actually deploying the .war file?

Accepted Answer:

This issue came down to Docker Desktop (Windows) interfering with Tomcat and specifically Tomcat's admin port `8006`, which forced me to change the port number to allow Tomcat to launch (from `8006` to `8007` for example). I had an old Tomcat container at one point in time, and this may have been fired up when my computer started via Docker Desktop without me realizing it.

Steps completed to fix the issue assuming your `pom.xml` has the necessary dependencies.

1. Stop Docker Desktop
2. Use the latest [JDK11 build](#) and update `JAVA_HOME` as needed
3. Restore default Tomcat admin server port to `8006`
4. Republish/Restart the Tomcat server

I think in most cases this issue will be resolved as explained [here](#) from updating the `pom.xml` with the needed dependencies. In my case it was entirely environmental and for some reason Docker Desktop prevented Tomcat from accessing the `jaws-api-2.3.1.jar` file at runtime. Tomcat's inability to access this .jar file was confirmed via my 'Update 2' in the question above.

My `pom.xml` ended up only needing the following items:

Highest Rated Answer:

Try by using this dependency in your `pom.xml`:

```
<dependency>      <groupId>javax.annotation</groupId>      <artifactId>javax.annotation-api</artifactId>      <version>1.3.2</version>
```

I just tried and it works with tomcat 9 and spring 5.1.5