链滴

使用 jdk9 运行报错：Caused by: java.lang.ClassNotFoundException: javax.xml.bind.JAXBException

作者：shixiaoxiang

原文链接：https://ld246.com/article/1511838497949

来源网站：链滴

做了一段时间的jdk 8 项目后，jdk 9 发布了，跟随时代的潮流，立马将jdk切换为 9 ，然后。。。。。

运行什么的都没有问题，在进行mvn clean install后，吧啦吧啦的一大堆test error，单独运行个子项目后发现，运行测试用例的时候程序抛异常了：

```
java.lang.IllegalStateException: Failed to load ApplicationContext


at org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate.loadCo
text(DefaultCacheAwareContextLoaderDelegate.java:124)
at org.springframework.test.context.support.DefaultTestContext.getApplicationContext(Defaul
TestContext.java:83)
at org.springframework.test.context.support.DependencyInjectionTestExecutionListener.inject
ependencies(DependencyInjectionTestExecutionListener.java:117)
at org.springframework.test.context.support.DependencyInjectionTestExecutionListener.prepa
eTestInstance(DependencyInjectionTestExecutionListener.java:83)
at org.springframework.boot.test.autoconfigure.SpringBootDependencyInjectionTestExecutio
Listener.prepareTestInstance(SpringBootDependencyInjectionTestExecutionListener.java:44)
at org.springframework.test.context.TestContextManager.prepareTestInstance(TestContextM
nager.java:230)
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.createTest(SpringJUnit4Cl
ssRunner.java:228)
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner$1.runReflectiveCall(Sprin
JUnit4ClassRunner.java:287)
at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.methodBlock(SpringJUnit
ClassRunner.java:289)
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runChild(SpringJUnit4Cla
sRunner.java:247)
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runChild(SpringJUnit4Cla
sRunner.java:94)
at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)
at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)
at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)
at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)
at org.springframework.test.context.junit4.statements.RunBeforeTestClassCallbacks.evaluate(
unBeforeTestClassCallbacks.java:61)
at org.springframework.test.context.junit4.statements.RunAfterTestClassCallbacks.evaluate(Ru
AfterTestClassCallbacks.java:70)
at org.junit.runners.ParentRunner.run(ParentRunner.java:363)
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.run(SpringJUnit4ClassRu
ner.java:191)
at org.junit.runner.JUnitCore.run(JUnitCore.java:137)
at com.intellij.junit4.JUnit4IdeaTestRunner.startRunnerWithArgs(JUnit4IdeaTestRunner.java:68

at com.intellij.rt.execution.junit.IdeaTestRunner$Repeater.startRunnerWithArgs(IdeaTestRunne
.java:47)
at com.intellij.rt.execution.junit.JUnitStarter.prepareStreamsAndStart(JUnitStarter.java:242)
at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitStarter.java:70)

Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean wi
h name 'entityManagerFactory' defined in class path resource [org/springframework/boot/au
oconfigure/orm/jpa/HibernateJpaAutoConfiguration.class]: Invocation of init method failed; n
```

sted exception is java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initialize
ean(AbstractAutowireCapableBeanFactory.java:1628)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreat
Bean(AbstractAutowireCapableBeanFactory.java:555)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBe
n(AbstractAutowireCapableBeanFactory.java:483)
at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBean
actory.java:306)
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(Def
ultSingletonBeanRegistry.java:230)
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanF
ctory.java:302)
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFac
ory.java:197)
at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicat
onContext.java:1078)
at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitializ
tion(AbstractApplicationContext.java:857)
at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicati
nContext.java:543)
at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:693)
at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:360)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:303)
at org.springframework.boot.test.context.SpringBootContextLoader.loadContext(SpringBootC
ntextLoader.java:120)
at org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate.loadCo
textInternal(DefaultCacheAwareContextLoaderDelegate.java:98)
at org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate.loadCo
text(DefaultCacheAwareContextLoaderDelegate.java:116)
... 25 more

Caused by: java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException

at org.hibernate.boot.spi.XmlMappingBinderAccess.<init>(XmlMappingBinderAccess.java:43)
at org.hibernate.boot.MetadataSources.<init>(MetadataSources.java:87)
at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.<init>(EntityManagerFact
ryBuilderImpl.java:179)
at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.<init>(EntityManagerFact
ryBuilderImpl.java:149)
at org.springframework.orm.jpa.vendor.SpringHibernateJpaPersistenceProvider.createContain
rEntityManagerFactory(SpringHibernateJpaPersistenceProvider.java:54)
at org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean.createNativeEntity
anagerFactory(LocalContainerEntityManagerFactoryBean.java:353)
at org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.buildNativeEntityManage
Factory(AbstractEntityManagerFactoryBean.java:370)
at org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.afterPropertiesSet(Abstra
tEntityManagerFactoryBean.java:359)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeIni
Methods(AbstractAutowireCapableBeanFactory.java:1687)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initialize
ean(AbstractAutowireCapableBeanFactory.java:1624)

... 40 more

Caused by: java.lang.ClassNotFoundException: javax.xml.bind.JAXBException

at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:582)
at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:185)
at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:496)
... 50 more

直接绕过百度、必应，在ｇｏｏｇｌｅ上一搜，情况立马知晓了。ｓｔａｃｋｏｖｅｒｆｌｏｗ面这么解释：

The JAXB APIs are considered to be Java EE APIs, and therefore are no longer contained on th default class path in Java SE 9.

Java 9 introduces the concepts of modules, and by default the java.se aggregate module is av ilable on the class path (or rather, module path). As the name implies, the java.se aggregate odule does not include the Java EE APIs that have been traditionally bundled with Java 6/7/8.

Fortunately, these Java EE APIs that were provided in JDK 6/7/8 are still in the JDK, but they ju t aren't on the class path by default. The extra Java EE APIs are provided in the following mod les:

java.activation
java.corba
java.transaction
java.xml.bind << This one contains the JAXB APIs
java.xml.ws
java.xml.ws.annotation

Quick Solution:

To make the JAXB APIs available at runtime, specify the following command-line option:

--add-modules java.xml.bind

But I still need this to work with Java 8!!!

If you try specifying --add-modules with an older JDK, it will blow up becuase it's an unrecogn zed option. I suggest one of two options:

1. You can conditionally apply the argument in a launch script (if you have one) by inspecting he JDK version by inspecting $JAVA_HOME/release for the JAVA_VERSION property.

2. You can add the -XX:+IgnoreUnrecognizedVMOptions to make the JVM silently ignore u recongized options, instead of blowing up. But beware! Any other command line args you use will no longer be validated for you by the JVM. This option works with Oracle/OpenJDK as wel as IBM JDK (as of JDK 8sr4)

怎么样？看出来什么了么？简单来说，就是ｊｄｋ９的模块化功能造成的，java.xml.bind其实是属于ａｖａ　ｅｅ的模块，在ｊｄｋ６／７／８版本中都方在了默认的ｐａｔｈ下面，但是到了ｊｄｋ９就把这些模块

都移出来了，就是没有放在默认的ｐａｔｈ下面，如果想要使用，那么你需要在运行时加上ｏｐｔｉｎｓ，命令--add-modules java.xml.bind。

那么坑又来了，我一开始的时候以为是需要通过命令 java --add-modules java.xml.bind去添加模块ｊａｖａ命令中也有这个命令，但就是提示命令可能有误，一直没反应。然后就去把ｇｏｏｇｌｅ各

翻，最终，结果找到了：

Test Failure
The next step sees the failure of unit tests to fail with `mvn test`.

The cause is the same, but it’ s a bit harder to find. It requires checking the Surefire reports. ome contain exceptions with the following line:

Caused by: java.lang.ClassNotFoundException: javax.xml.bind.JAXBException

Again, test code cannot access the module. This time, however, the _maven-surefire-plugin_ eeds to be configured:

<plugin>

<artifactId>maven-surefire-pluginartifactId>

<version>2.20.1version>

<configuration>

<argLine>--add-modules java.xml.bindargLine>

configuration>

plugin>

ok,在ｐｏｍ文件里面的插件配置里面加上这句话，加一个配置就搞定了。另外，在这个网站中也说了，在编译、mvn　spring-boot:run、打包时出现这类异常该怎么解决，网址

https://dzone.com/articles/migrating-a-spring-boot-application-to-java-9-comp(可能需要翻墙)