



链滴

(转)ClassNotFoundException和NoClassDefFoundError的区别

作者: jaz

原文链接: <https://ld246.com/article/1466238463951>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>正如它们的名字所说明的：NoClassDefFoundError是一个错误(Error)，而ClassNotFountException是一个异常，在Java中错误和异常是有区别的，我们可以从异常中恢复程序但却不应该尝试从错中恢复程序。</p>

<p>ClassNotFountException的产生原因：</p>

<p>Java支持使用Class.forName方法来动态地加载类，任意一个类的类名如果被作为参数传递给这个方法都将导致该类被加载到JVM内存中，如果这个类在类路径中没有被找到，那么此时就会在运行时出ClassNotFountException异常。</p>

<p>要解决这个问题很容易，唯一需要做的就是确保所需的类连同它依赖的包存在于类路径中。当Class.forName被调用的时候，类加载器会查找类路径中的类，如果找到了那么这个类就会被成功加载如果没找到，那么就会抛出ClassNotFountException，除了Class.forName，ClassLoader.loadClass、ClassLoader.findSystemClass在动态加载类到内存中的时候也可能会抛出这个异常。</p>

<p>另外还有一个导致ClassNotFountException的原因就是：当一个类已经某个类加载器加载到内中了，此时另一个类加载器又尝试着动态地从同一个包中加载这个类。</p>

<p>由于类的动态加载在某种程度上是被开发者所控制的，所以他可以选择catch这个异常然后采取应的补救措施。有些程序可能希望忽略这个异常而采取其他方法。还有一些程序则会终止程序然后让户再次尝试前做点事情。</p>

<p>NoClassDefFoundError产生的原因：</p>

<p>如果JVM或者ClassLoader实例尝试加载（可以通过正常的方法调用，也可能是使用new来创建的对象）类的时候却找不到类的定义。要查找的类在编译的时候是存在的，运行的时候却找不到了。个错误往往是你使用new操作符来创建一个新的对象但却找不到该对象对应的类。这个时候就会导致oClassDefFoundError.</p>

<p>由于NoClassDefFoundError是有JVM引起的，所以不应该尝试捕捉这个错误。</p>

<p>解决这个问题的办法就是：查找那些在开发期间存在于类路径下但在运行期间却不在类路径下的。</p>

<p>另：</p>

<p>ClassNotFountException发生在装入阶段。
当应用程序试图通过类的字符串名称，使常规的三种方法装入类，但却找不到指定名称的类定义时就抛出该异常。</p>

<p>NoClassDefFoundError：当目前执行的类已经编译，但是找不到它的定义时</p>

<p>也就是说你如果编译了一个类B，在类A中调用，编译完成以后，你又删除掉B，运行A的时候那就会出现这个错误</p>

<p>加载时从外存储器找不到需要的class就出现ClassNotFountException
连接时从内存找到需要的class就出现NoClassDefFoundError</p>

<p>另：</p>

<p>
NoClassDefFoundError 解决的三种方法：</p>

<p>1. Simple example of NoClassDefFoundError is class belongs to a jar and jar was not add d into classpath or sometime jar’s name has been changed by someone like in my cas one of my colleague has changed tibco.jar into tibco_v3.jar and by program is failing with jav .lang.NoClassDefFoundError and I was wondering what’s wrong.</p>

<p>首先是类在运行的时候依赖于其它的一个jar包，但是该jar包没有加载到classpath中或者是该jar的名字被其他人改了，就像我的一个例子tibo.jar改为了tibco_v3.jar…….</p>

<p>2. Class is not in Classpath, there is no sure shot way of knowing it but many a times you an just have a look to print System.getproperty(”java.classpath“)and it will print the classpath from there you can at least get an idea of your actual runtime classpath.</p>

<p>运行的类不在classpath中，这个问题没有一个确定的方法去知道，但是很多时候你可以通过Syst m.getproperty(”java.classpath“)方法，该方法能让你至少可以领略到实际存在的运期间的classpath。</p>

<p>
3. Just try to run with explicitly -classpath option with the classpath you think will ork and if its working then it’s sure short sign that some one is overriding java classpa h.</p>

<p>试着通过-classpath命令明确指出你认为正确的classpath，如果能够正常执行的话就说明你使的classpath是正确的，而系统中的classpath已经被修该过了。</p>