1、什么叫做maven的依赖冲突问题？853769620

大家先自己思考一下自己有没有遇到过，上一讲，我们不是说maven会自动依赖调解，说对已给项目不同的版本选择一个版本来使用

但是如果maven选择了错误的版本呢。。。。。。

某某某class not found

某某某方法() not found

MyClass.doMethod() not found。。。。

oh my god，这个依赖，我有声明啊，可能是我的某个一级依赖引入的二级依赖，或者三级依赖，但是为啥这个坑爹的二级或者三级依赖找不到呢。。。

这个就是典型的工作场景里面的依赖冲突问题

传递性依赖 + 依赖调解，导致了这个问题

2、依赖冲突是如何产生的？

比如你依赖了A和B，此时A依赖了C-1.0，B依赖了D，D依赖了C-2.0

X

-> A

-> C-1.0

-> B

-> D

-> C-2.0

X -> A -> C-1.0

X -> B -> D -> C-2.0

此时就会导致的事情是，由于A -> C1.0是最短路径，所以会用C1.0

但是坑爹的事情发生了，B依赖的是D，D依赖的C结果用了C-1.0版本

D本来用的是C-2.0的一个方法，但是现在给D的时C-1.0的一个类

比如C-1.0的类CClass.sayHello()

C-2.0给类CClass加了一些方法，比如CClass.printHello()，同时也有CClass.sayHello()方法

D调用了C-2.0里面的printHello()这个方法

如果用的时C-1.0，把C-1.0的CClass提供给D去用，D去调用printHello()方法的时候，就会报错。。。

C这个项目的CClass这个类的printHello()这个方法没有找到，not found的异常

3、如何解决这样的依赖冲突？

很简单，一般来说，用最新的版本，因为新版本一般可以兼容旧版本，但是旧版本一定不会提供新版本的功能。一个开源项目是一定要提供向前兼容性的，否则就是太垃圾了，那你可以考虑换一个开源项目了。

你不要让maven自动去选择C-1.0来使用，而是手动强制使用C-2.0，不就可以了么。。。。

4、mvn depedency:tree这个命令

mvn dependency:tree，用dependency插件的tree goal，执行，进行依赖链条分析

我这边告诉大家，比如你发现C这个项目下报出了类似的问题，然后就用mvn dependency:tree这个命令看一下整个项目的依赖路径树，看看所有的依赖路径里，有哪几个版本的C项目，看一下，自己看看要用哪个版本的C，然后将其他的C版本的依赖全部手动排除掉

A

--C-1.0

B

--D

--C-2.0

定位原因，因为A-C1.0最短，所以用了C-1.0，也就是旧的版本

而大多数项目都是最新版本兼容旧版本，也就是说，用C-2.0的话，A依赖的C-1.0的代码是可以运行的

所以此时要做的事情就是。。。

在A下面

<dependency>

<groupId>A</groupId>

<artifactId>A</artifactId>

<version>1.0</version>

<exclusions>

<exclusion>

<groupId>C</groupId>

<artifactId>C</artifactId>

</exclusion>

</exclusions>

</dependency>

通过这个以后，你如果再次运行mvn dependency:tre

就会发现依赖树变成下面这样了：

A

B

--D

--C-2.0

这样的话，就只剩下了一个C-2.0，那肯定是用C-2.0楼，无论是A还是D，都是用C-2.0了

（1）maven工作中最常见的依赖冲突问题的现象？

（2）产生的原因是什么？

（3）解决的思路

（4）具体用什么命令和配置去解决