𝐌𝐚𝐯𝐞𝐧 𝐃𝐞𝐩𝐞𝐧𝐝𝐞𝐧𝐜𝐲 𝐒𝐜𝐨𝐩𝐞𝐬  
-------------------------------------------  
Maven is a widely used build management tool and Dependency management is a core feature of Maven.  
  
It's very common to struggle with issues related to managing dependencies. I am sure at times you must have faced it. It gets even worse when you have a multi-module setup and also when the transitive dependencies come into the picture 🤣  
  
But, when you read, learn, and debug, things start getting easier. 😊  
  
Direct dependencies are the ones that you add explicitly but the transitive ones are included in turn by the direct dependencies.  
  
One good thing to learn, if you don't already know is Dependency Scopes - Which helps in determining when a dependency is included in a classpath. And trust me, many don't know about it, even though they have used maven for years.  
  
So there are basically 6 scopes (you can see them in the image below), but here I will mainly talk about the 4 important scopes.  
  
- Compile - It is the default one. If you don't explicitly mention any scope, it's compile.  
  
<dependency>  
<groupId>[some.group.id](http://some.group.id/)</groupId>  
<artifactId>artifact-id</artifactId>  
<version>1.0.4</version>  
</dependency>  
  
So this dependency is available to you in all phases/classpaths.  
  
- Provided - It's like telling maven that please just provide me the dependency during compile time but at runtime, you don't need to worry, it will be provided to me by the container or JDK and need not be included in the final artifact/jar/war.  
  
<dependency>  
<groupId>[some.group.id](http://some.group.id/)</groupId>  
<artifactId>artifact-id</artifactId>  
<version>1.0.4</version>  
<scope>provided</scope>  
</dependency>  
  
- Runtime - When you don't need the library when your code is getting compiled but is needed in all other phases (test, runtime, etc).  
  
<scope>runtime</scope>  
  
- Test - When the library is only needed to run unit tests (like JUnit etc) or needed by/within unit tests only, use this scope.  
  
<scope>test</scope>  
  
  
Also, Test and Provided are not transitive in nature, which simply means if your library which can be of any scope has a transitive dependency on a library that is either test or provided, they won't be pulled in your project.  
If you want to use those libraries explicitly add them as direct dependencies. But the compile and runtime transitive libs are propagated to your project build as well.  
  
  
I hope it was helpful!  
  
  
𝐁𝐭𝐰 𝐰𝐡𝐢𝐜𝐡 𝐛𝐮𝐢𝐥𝐝 𝐭𝐨𝐨𝐥 𝐚𝐫𝐞 𝐲𝐨𝐮 𝐮𝐬𝐢𝐧𝐠? 𝐌𝐚𝐯𝐞𝐧 𝐨𝐫 𝐆𝐫𝐚𝐝𝐥𝐞?  
  
If you like, you can connect/follow me - [Mayank Ahuja](https://www.linkedin.com/in/ACoAABqrgQcB72kLYkO2NPQ5F4SS-6pPeg2XODQ)  
  
[#maven](https://www.linkedin.com/feed/hashtag/?keywords=maven&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6956534673279700992) [#java](https://www.linkedin.com/feed/hashtag/?keywords=java&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6956534673279700992) [#learningdaily](https://www.linkedin.com/feed/hashtag/?keywords=learningdaily&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6956534673279700992) [#linkedin](https://www.linkedin.com/feed/hashtag/?keywords=linkedin&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6956534673279700992) [#programminglife](https://www.linkedin.com/feed/hashtag/?keywords=programminglife&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6956534673279700992)

