

How to add local project (not jar) as a dependency to a Maven project

Asked 6 years, 6 months ago Modified 1 year, 3 months ago Viewed 32k times



33



I have two Maven projects that I've added to one IntelliJ Idea's project as two modules. Project B depends on Project A.

Here are simplified versions of their pom.xml files.

Project A:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.group</groupId>
  <artifactId>a</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <dependencies>
  </dependencies>
</project>
```

Project B:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.group</groupId>
  <artifactId>b</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
```

```
</build>

<dependencies>
  <dependency>
    <groupId>com.group</groupId>
    <artifactId>a</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
</dependencies>
</project>
```

Project A compiles easily since it has no dependency. But Project B depends on Project A and since I'm not telling maven how to find it, it can not be compiled with `mvn package`. But if I'm not mistaken, this was possible using IntelliJ Idea's "Maven Projects" menu since to IntelliJ Idea both projects are known.

But right now, for some unknown reason, I can not compile project B even in IntelliJ Idea. When I do, it prompts:

```
[ERROR] Failed to execute goal on project b: Could not resolve dependencies for
project com.group:b:jar:1.0-SNAPSHOT: Could not find artifact
com.group:a:jar:1.0-SNAPSHOT -> [Help 1]
```

My question is, how can I include one project into other as its dependency? Please note that I'm not looking for a local dependency injection in maven since I want this to work in the future when I upload my packages into some repository.

[java](#) [maven](#) [intellij-idea](#)

Share Improve this question

edited Aug 22, 2018 at 14:51

asked May 19, 2017 at 15:46

Follow



Mehran

15.8k 27 124 228

- 3 Did you do mvn clean install on your project a? If not do mvn install your project 'a' to local repo from where it can find 'a' while trying to build b. Basically mvn looks for 'a' in local repo, so until and unless you install it in local repo using mvn install, it cannot be found – [user3487063](#) May 19, 2017 at 15:51

But this causes Eclipse features to not work. For example, if I right click on a class name and select "References > Workspace", it will open up the class file from the jar and not the java file from the neighboring project in the workspace (which is the source of the class file). How to make maven work properly? i.e. when I do the above, the java source file should open and not the class file in jar – [BATMAN_2008](#) Jun 9, 2021 at 4:53

2 Answers

Sorted by:

Highest score (default)





40



Your project B depends on project A and you have already specified this as a dependency in your pom file. The only reason project B is not able to find project A may be due because of project A artifact is not installed in your local .m2 repository. So you have to first do a mvn clean install of your project A which will create the artifact in your maven repository. You should be then able to use artifact A in project B.

Share Improve this answer Follow

answered May 19, 2017 at 15:59



Nitin Prabhu

624 6 10

6 And why maven doesn't do this automatically? Thank God for gradle – smac89 Oct 30, 2019 at 3:25

1 But this causes Eclipse features to not work. For example, if I right click on a class name and select "References > Workspace", it will open up the class file from the jar and not the java file from the neighboring project in the workspace (which is the source of the class file). How to make maven / eclipse work properly? i.e. when I do the above, the java source file should open and not the class file in jar – Dojo Apr 5, 2021 at 12:47

@Dojo Even I am facing the exact same issue. I would like my Project B to use Project A directly so that I can make few changes to Project A based on output during the debug and test it right out. If I do mvn clean install then I need to do the same again and again when I make the changes. Were you able to find the solution for this? If so please post here. Thanks in advance. – BATMAN_2008 Jun 8, 2021 at 17:23

@BATMAN_2008 Not yet – Dojo Jun 8, 2021 at 17:37

@BATMAN_2008 By the way in gradle, if you configure everything correctly, you can make gradle generate the eclipse .project file to correctly setup inter project dependencies. Everything is eclipse aware so you can even set breakpoints and do hot code change while debugging. Gradle has a steep learning curve though. – Dojo Jun 8, 2021 at 17:40



0



Since I came here searching for the same problem with **Eclipse** and it is mentioned in some comments (where it seems the IntelliJ focus is not obvious) I want to point to this solution for Eclipse that may be similar to IntelliJ as well: <https://stackoverflow.com/a/10696646/1915920>

The crucial point for us was the **dependent projects** `pom.xml` **version match (in the dependent and depending `pom.xml`)** in the workspace, so the **workspace resolution** works.

Share Improve this answer Follow

answered Aug 5, 2022 at 11:37



Andreas Covidiot

4,386 5 51 97