Dependancy Management using Maven Exercise

Links:-
https://maven.apache.org/guides/introduction/introduction-to-repositories.html
https://devcenter.heroku.com/articles/local-maven-dependencies#create-a-local-maven-repository-directory

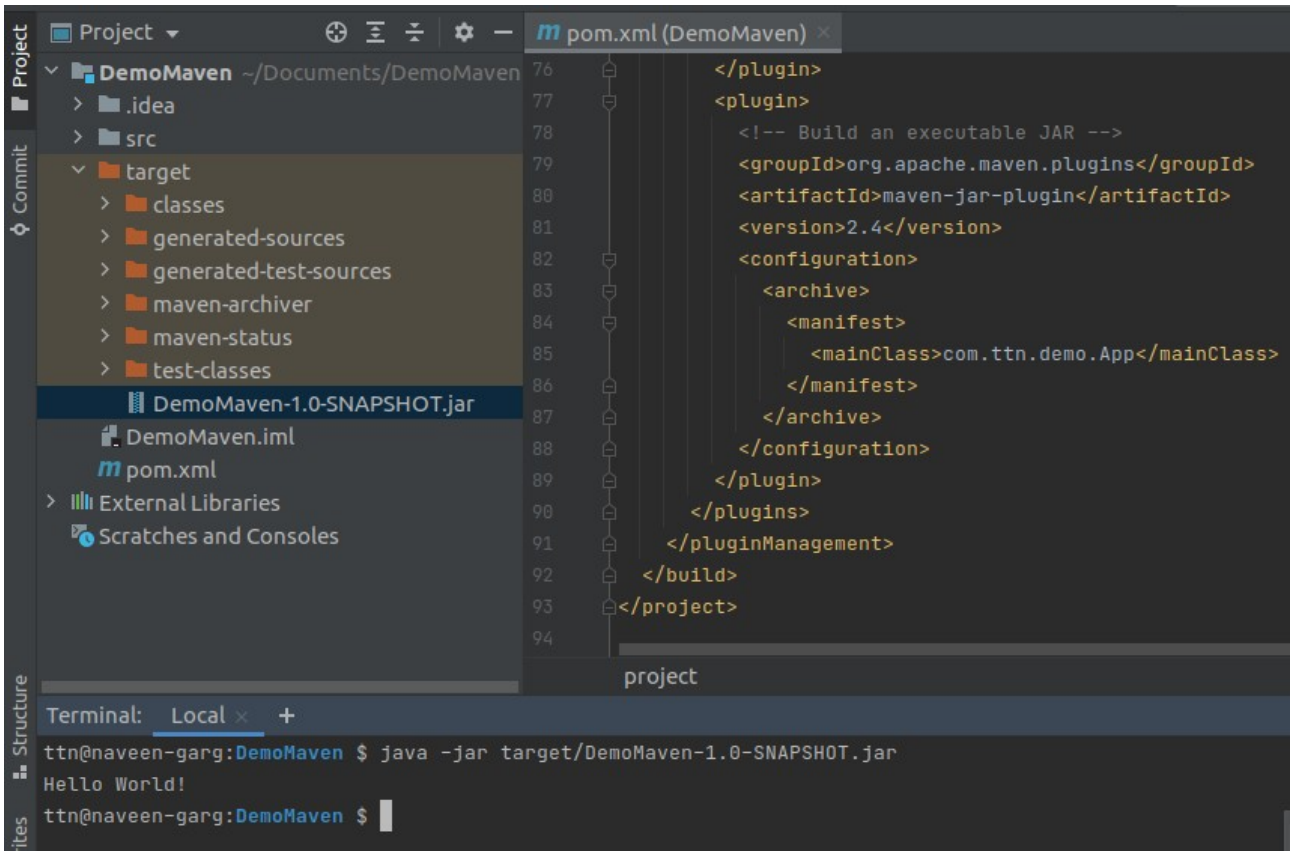
Ques 1. Add a maven dependency and its related repository URL.

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Ques 2 .Add a new repository in the pom.xml and use its dependencies.

```
<repositories>
  <repository>
    <id>my-internal-site</id>
    <url>https://maven.java.net/content/repositories/public/</url>
  </repository>
</repositories>
```

Ques 3. Using JAR plugin, make changes in the pom.xml to make the jar executable. Using java -jar JAR_NAME, the output should be printed as "Hello World"



Ques 4.-Differentiate between the different dependency scopes: compile, runtime, test, provided

Sol-

Type of Scope of dependency as follows:-

Compile:- This is the default scope when no other scope is provided.Dependencies with this scope are available on the classpath of the project in all build tasks and they're propagated to the dependent projects.

More importantly, these dependencies are also transitive:

<dependency>

<groupId>commons-lang</groupId>

<artifactId>commons-lang</artifactId>

<version>2.6</version>

</dependency>

Link:-

https://mvnrepository.com/artifact/log4j/log4j

Runtime:-

The dependencies with this scope are required at runtime, but they're not needed for compilation of the project code. Because of that, dependencies marked with the runtime scope will be present in runtime and test classpath, but they will be missing from compile classpath.

A good example of dependencies that should use the runtime scope is a JDBC driver:

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>6.0.6</version>

<scope>runtime</scope>

</dependency>

Link-

https://mvnrepository.com/artifact/com.thoughtworks.xstream/xstream/1.4.14-jdk7

test- This scope indicates that the dependency is not required for normal use of the application and it is only available for the test compilation and execution phases. The standard use case for this scope is adding test library like JUnit to our application:

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

Link:-

https://mvnrepository.com/artifact/junit/junit/4.12

Provided:-

This scope is used to mark dependencies that should be provided at runtime by JDK or a container, hence the name.
A good use case for this scope would be a web application deployed in some container, where the container already provides some libraries itself.
For example, a web server that already provides the Servlet API at runtime, thus in our project, those dependencies can be defined with the provided scope:

```
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>servlet-api</artifactId>
<version>2.5</version>
<scope>provided</scope>
</dependency>
```

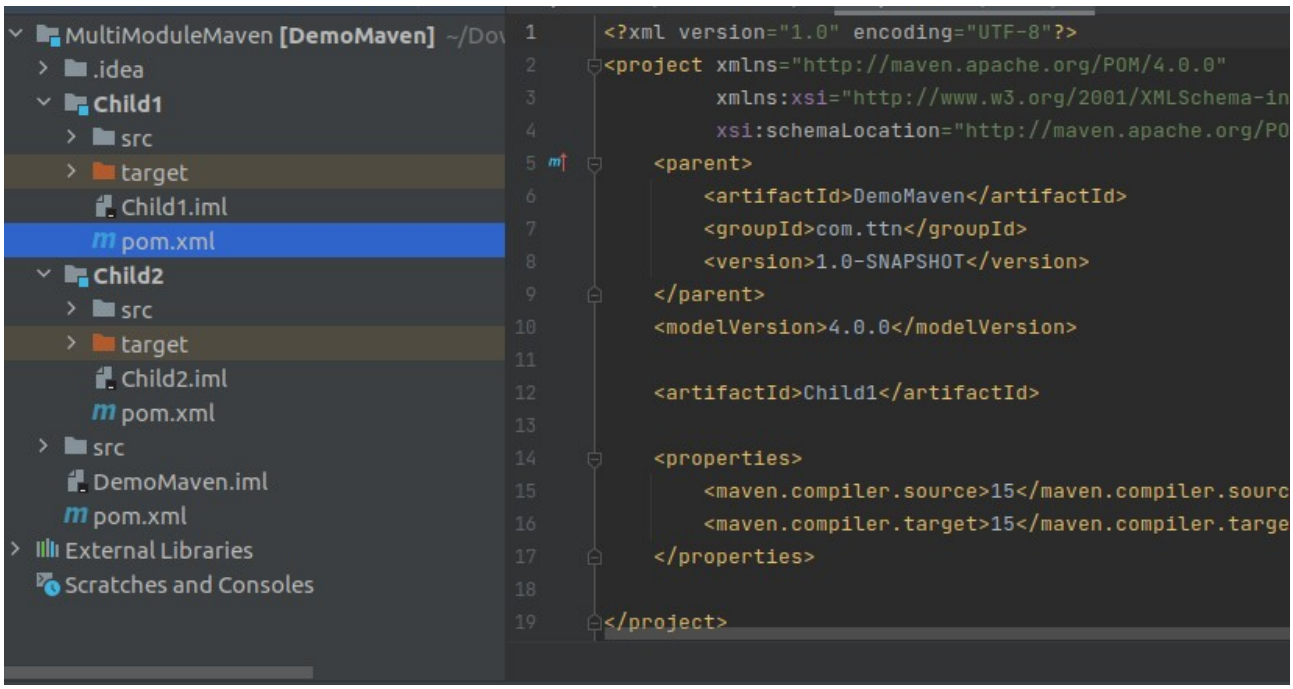
The provided dependencies are available only at compile-time and in the test classpath of the project; what's more, they aren't transitive.
Link-
https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api/3.1.0

Import:-

This scope was added in Maven 2.0.9 and it's only available for the dependency type pom. We're going to speak more about the type of the dependency in future articles.
Import indicates that this dependency should be replaced with all effective dependencies declared in it's POM:

```
<dependency>
<groupId>com.example</groupId>
<artifactId>custom-project</artifactId>
<version>1.3.2</version>
<type>pom</type>
<scope>import</scope>
</dependency>
```

Ques 5. Create a multi-module project. Run package command at the top level to make jar of every module.


MultiModuleMaven [DemoMaven] ~/Dov
.idea
Child1
src
target
Child1.iml
pom.xml
Child2
src
target
Child2.iml
pom.xml
src
DemoMaven.iml
pom.xml
External Libraries
Scratches and Consoles
<?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-in
xsi:schemaLocation="http://maven.apache.org/PO
<parent>
<artifactId>DemoMaven</artifactId>
<groupId>com.ttn</groupId>
<version>1.0-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>
<artifactId>Child1</artifactId>
<properties>
<maven.compiler.source>15</maven.compiler.sourc
<maven.compiler.target>15</maven.compiler.targe
</properties>
</project>