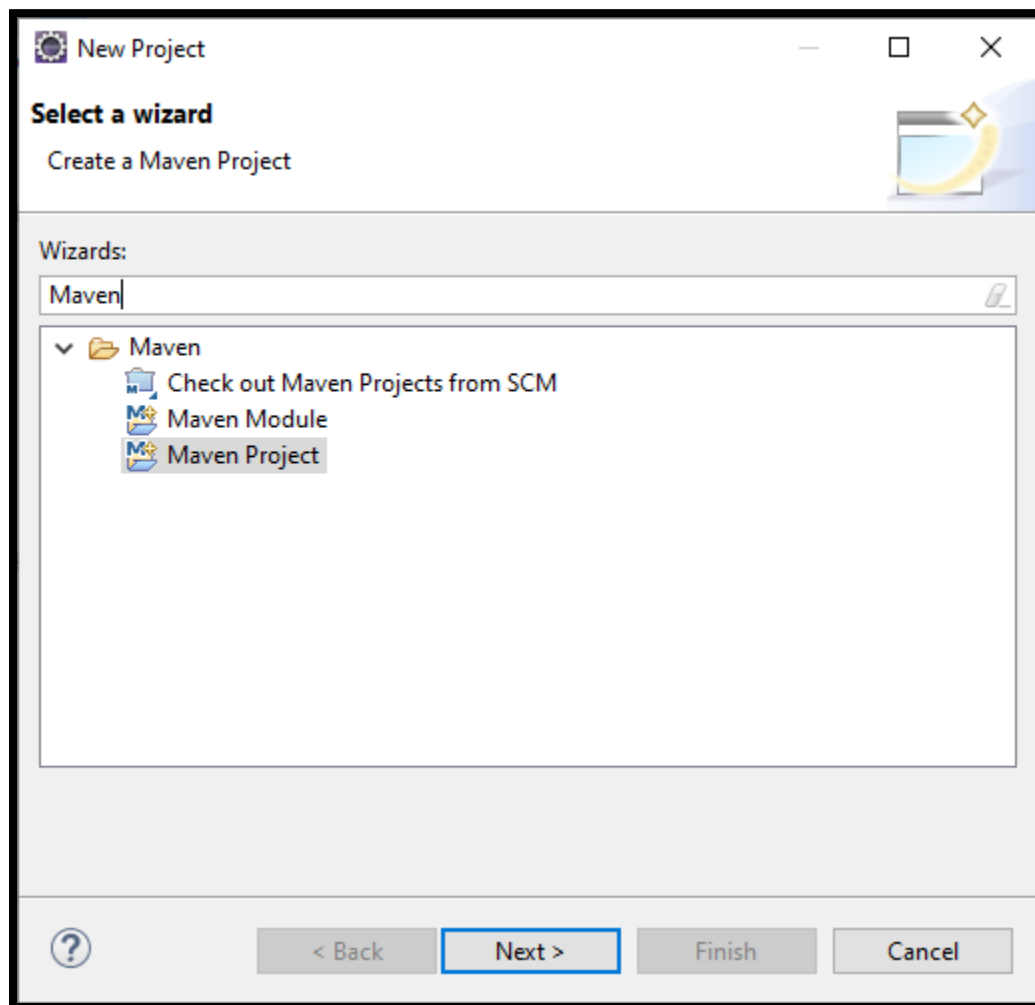


# (Transcript)

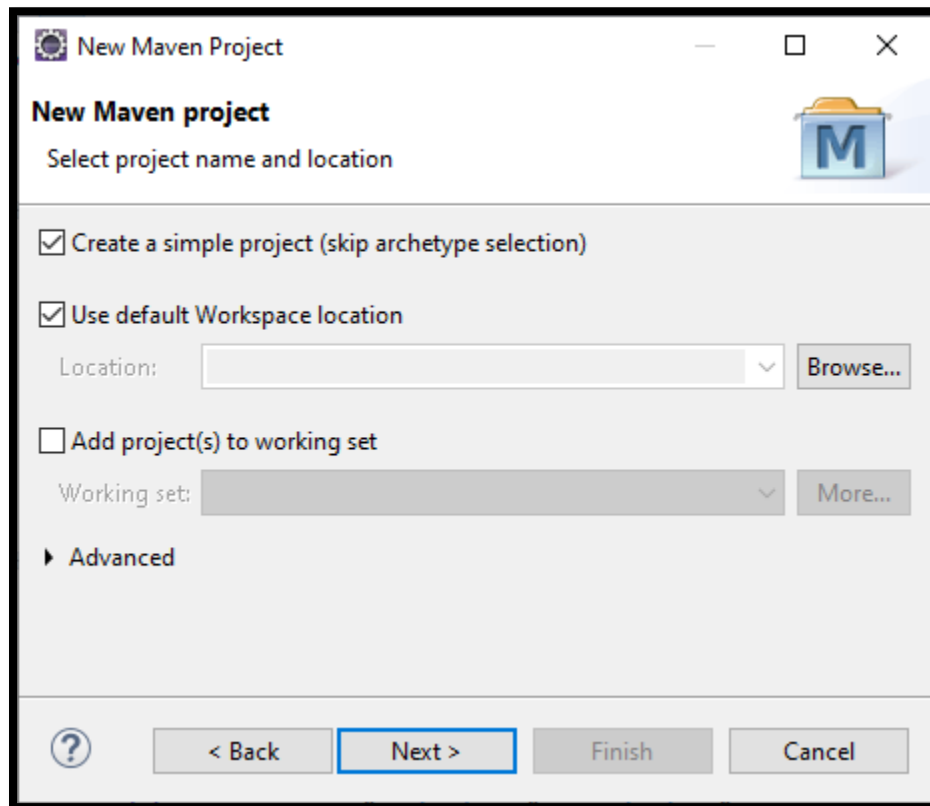
## Create Maven Project

### Build The Project

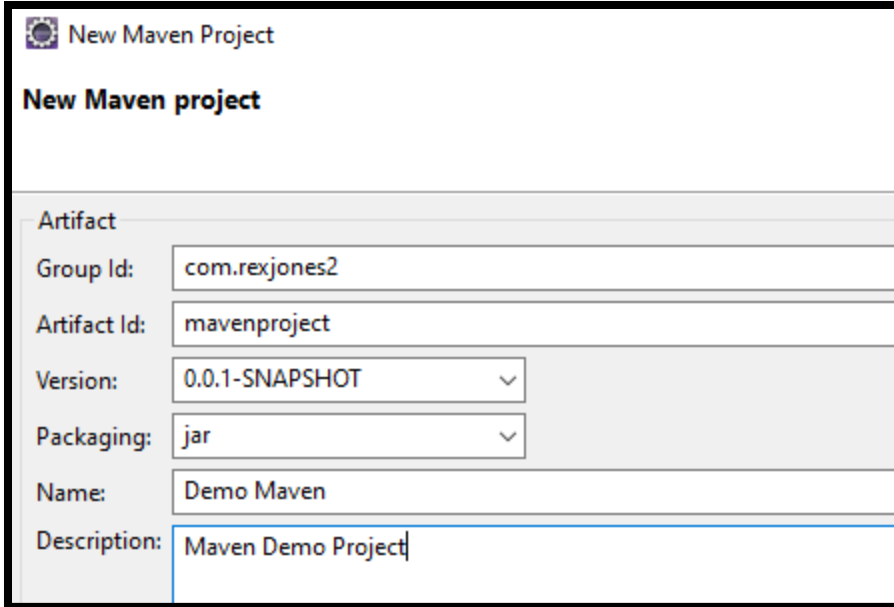
In this session, we are going to create a Maven Project. Start by going to File > New > Project. At this point, we see Maven Project but if we did not see Maven Project. We could search for Maven. Select Maven Project then Next.



In this dialog, we see this checkbox for “Create a simple project (skip the archetype selection). Archetype is a project template generated by Maven. Click Next and here’s a list of archetypes. We are going to skip and not let Maven generate our project. Click Back then click the checkbox and Next.



Group ID, Artifact ID, and Version are required fields. Group ID is a unique ID that identifies our project. The naming convention is similar to our package. It starts with a reversed domain com.rexjones2. Maven is not strict on the naming convention so the reversed domain is not required. Artifact ID is used as the subdirectory under Group ID. The convention is all lowercase letters. It’s the name of the project’s library artifact: mavenproject. Version is the version number of the project that will be packaged as a jar. Name is Demo Maven and Description is Maven Demo Project and Click Finish and now the Maven project has been created in Eclipse.



The screenshot shows the 'New Maven Project' dialog box. It has a title bar with the Eclipse logo and the text 'New Maven Project'. Below the title bar, the text 'New Maven project' is displayed. The dialog contains several input fields and dropdown menus:

Artifact	
Group Id:	com.rexjones2
Artifact Id:	mavenproject
Version:	0.0.1-SNAPSHOT
Packaging:	jar
Name:	Demo Maven
Description:	Maven Demo Project

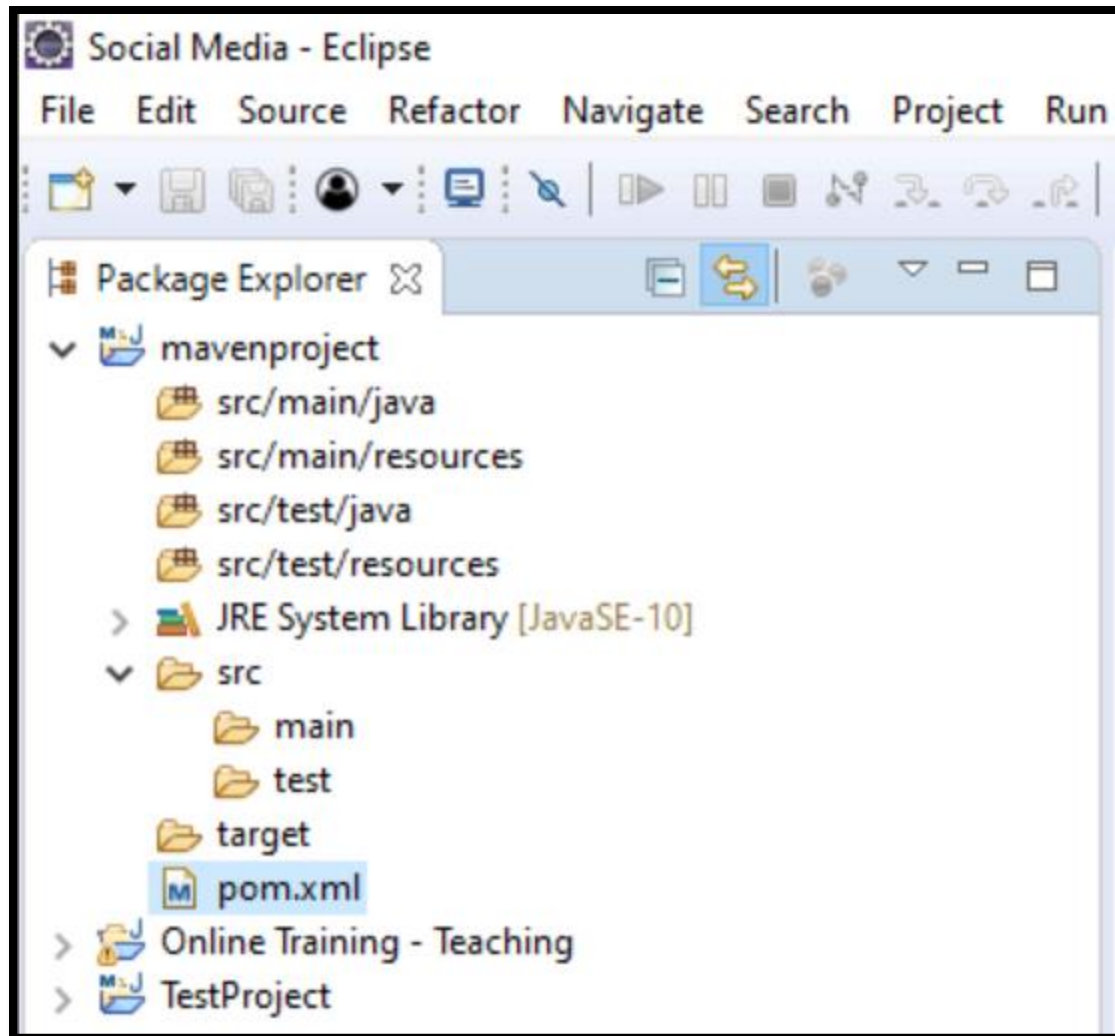
## Describe & Configure Project Structure

Restore and here's the Maven Project. The difference between a Maven Project and Java Project is Maven Project has an M with a J. Java Project only has a J. Do you see the pre-defined folder structure that includes 2 folders for src/main and 2 folders for src/test. src/main is the Library sources and resources. We add our framework code in the src/main/java directory.

src/test is the Test sources and resources. All of our Test Scripts are placed in src/test/java directory.

We have the JRE System Library which contains Java's jar files. You can update the Java version by right clicking the library > Build Path then Configure Build Path. Modulepath, JRE System Library then Click Edit. Select the Execution Environment drop down and select your version.

The src directory contains all of the source material for building the project. target is Maven's default output folder. It maintains all output of the build. The pom.xml file defines our Project Object Model. This is where we add our dependencies.



Next, we are going to add the Selenium and TestNG dependencies then execute our first test.

Social Media Contact

- ✓ YouTube <https://www.youtube.com/c/RexJonesII/videos>
- ✓ Facebook <http://facebook.com/JonesRexII>
- ✓ Twitter <https://twitter.com/RexJonesII>
- ✓ GitHub <https://github.com/RexJonesII/Free-Videos>
- ✓ LinkedIn <https://www.linkedin.com/in/rexjones34/>