

**Quality of Service Testing Suite for Cloud Software Services**

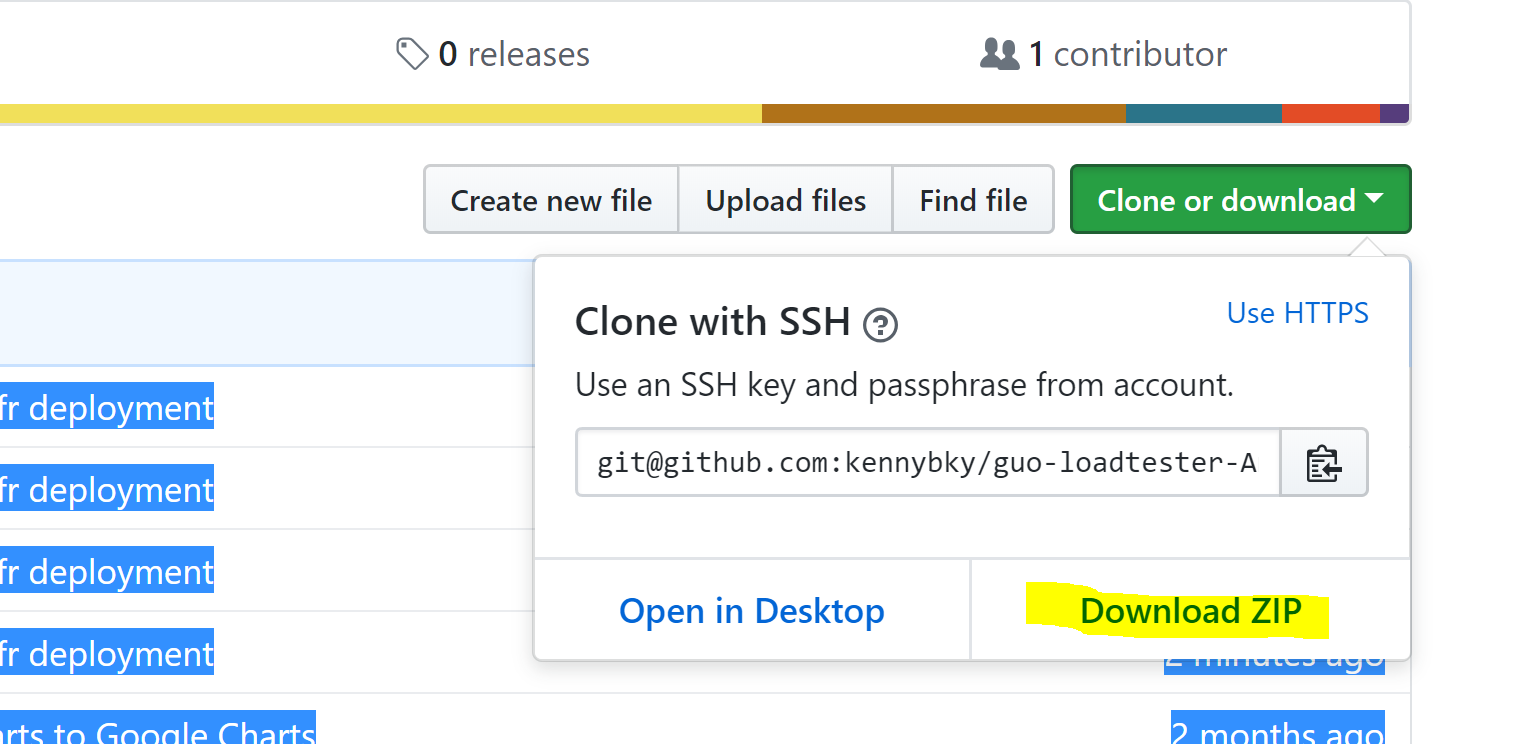
Advisor: Dr. Jiang Guo

**FIRST STEPS**

**DOWNLOAD PROJECT**

The project is hosted on Github. If you know how to use git. Then clone it with this url <https://github.com/kennybky/guo-loadtester-Angular4.git> For instructions on how to use git. Scroll to the very end of this manual. If you are using GitHub, it’s advisable to fork the repository to your own branch then clone it, so you can create pull requests.

If the above sounded like rubbish to you, and you don’t feel like learning git (Even though you probably should) Go to the URL above and download the project as a zip file. Slacker.



**Windows and Mac Instructions**

**Front End Requirements**

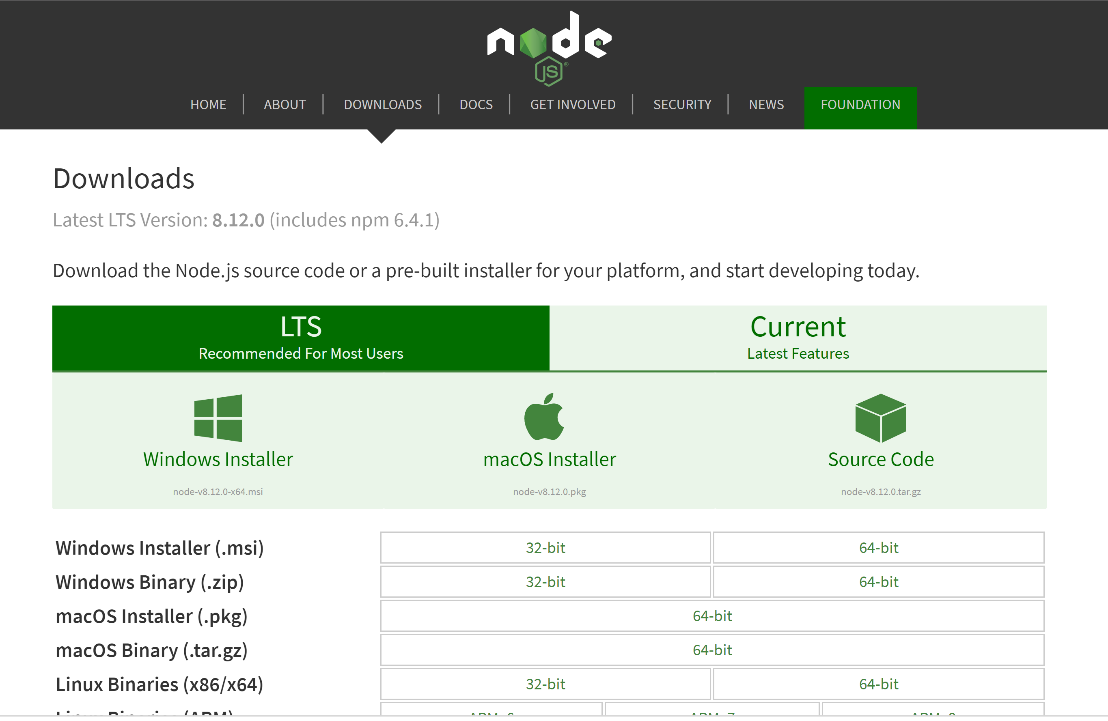
**Node.js and NPM**

Node.js is a platform for JavaScript. It is built on Chrome’s V8 JavaScript engine. Node.js uses an event- driven, non-blocking I/O model to remain lightweight and efficient. Node.js is needed to use Angular CLI which is one of our main tools for this project.

NPM stands for Node Package Manager. NPM lets you install and manage your project’s dependencies. All of our project’s front end dependencies are listed inside our package.json file and running npm install will immediately have all of the dependencies installed.

**Note:** For Mac, if you have Homebrew Installed, just type **brew install npm.** That should cover this whole node tutorial

1. Download the node.js installer from https://nodejs.org/en/download/

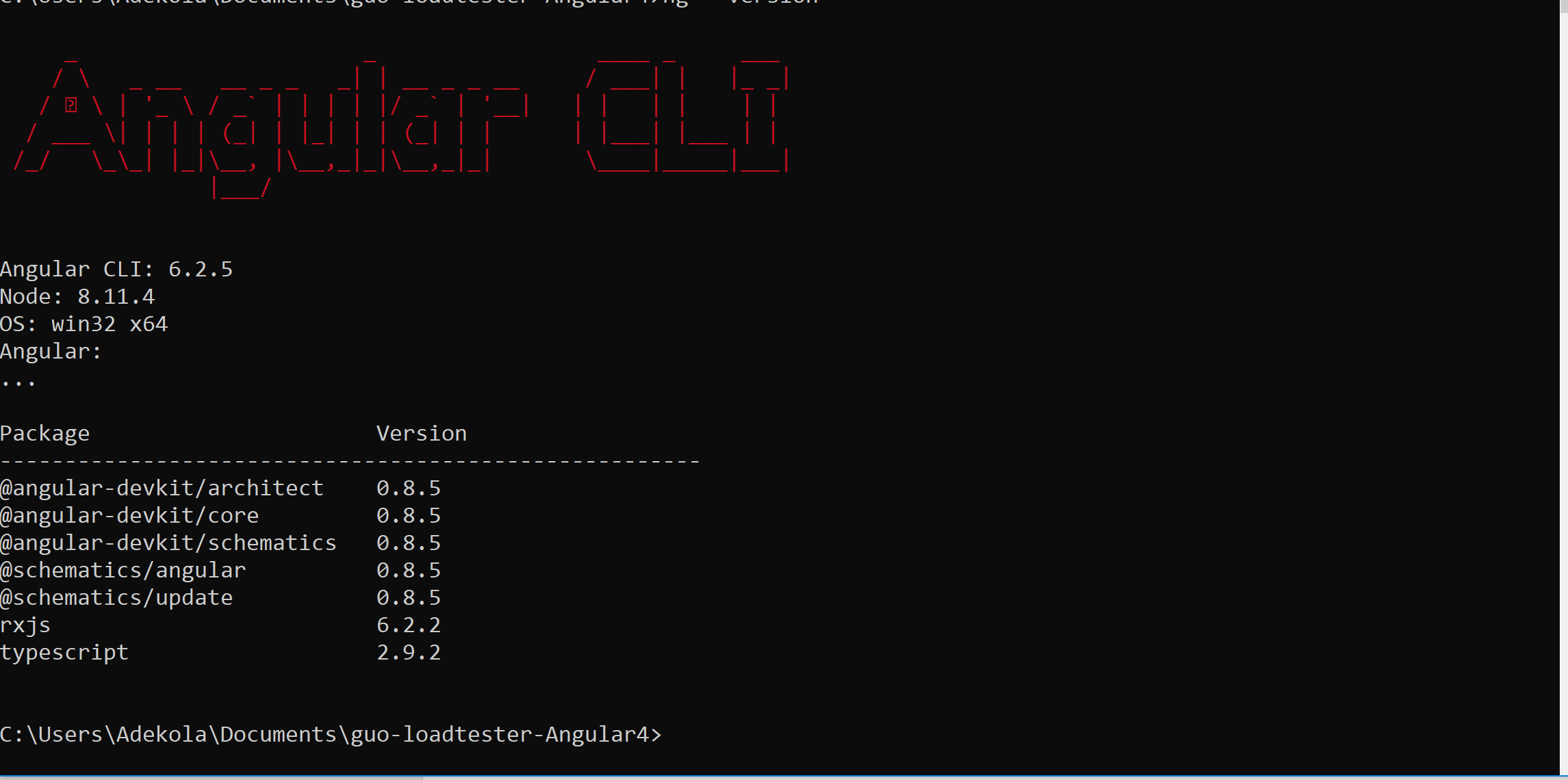


1. Run the installer.
2. Follow the directions of the node installer. Pick the default values for each step, no need to be fancy.
3. When node finishes installing open command prompt and type **node -v** to ensure node is installed. It should display your current version of node e.g. V8.11.4
4. Also run **npm -v** to ensure npm is installed. It should display your current version of npm
5. If you were successful at 4 & 5 congrats, you’ve installed node! You can go to the next section
6. If you were unsuccessful however, you might need to add node.js your path. (see #8 & #9 for windows, #10 & #11 for Mac)
7. To do so, navigate to your computer icon, click properties, Advanced System settings, then environment variables
8. In the System variables section, click on path, then edit. Locate your installation of node.js. It’s most likely C:\Program Files\nodejs. Add the full path to the last line. Click apply
9. For Mac Locate your installation of npm. It’s most likely /usr/local/bin/npm.
10. Run ***export PATH=$PATH:/your\_node\_installation***
11. Run the commands again.

**Angular CLI**

CLI stands for Command Line Interface. Angular CLI is used to create and manage Angular.io applications. It’s a useful tool to use when developing and creating Angular components

1. Open a command Prompt window, them run **npm install -g @angular/cli**
2. After it’s done, confirm it’s running but typing in the command ng –version

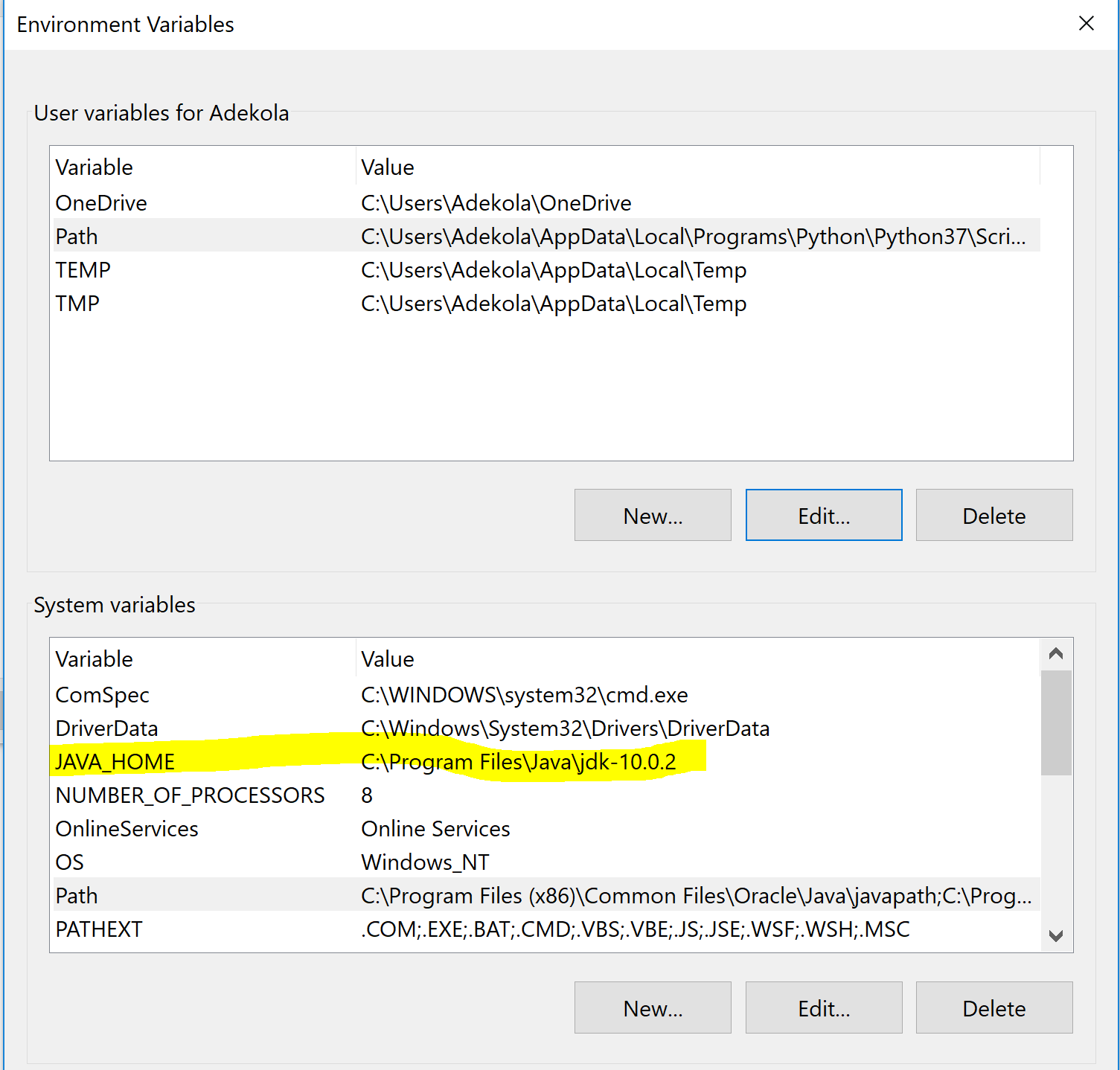


**Back end requirements**

**Java Development Kit (JDK)**

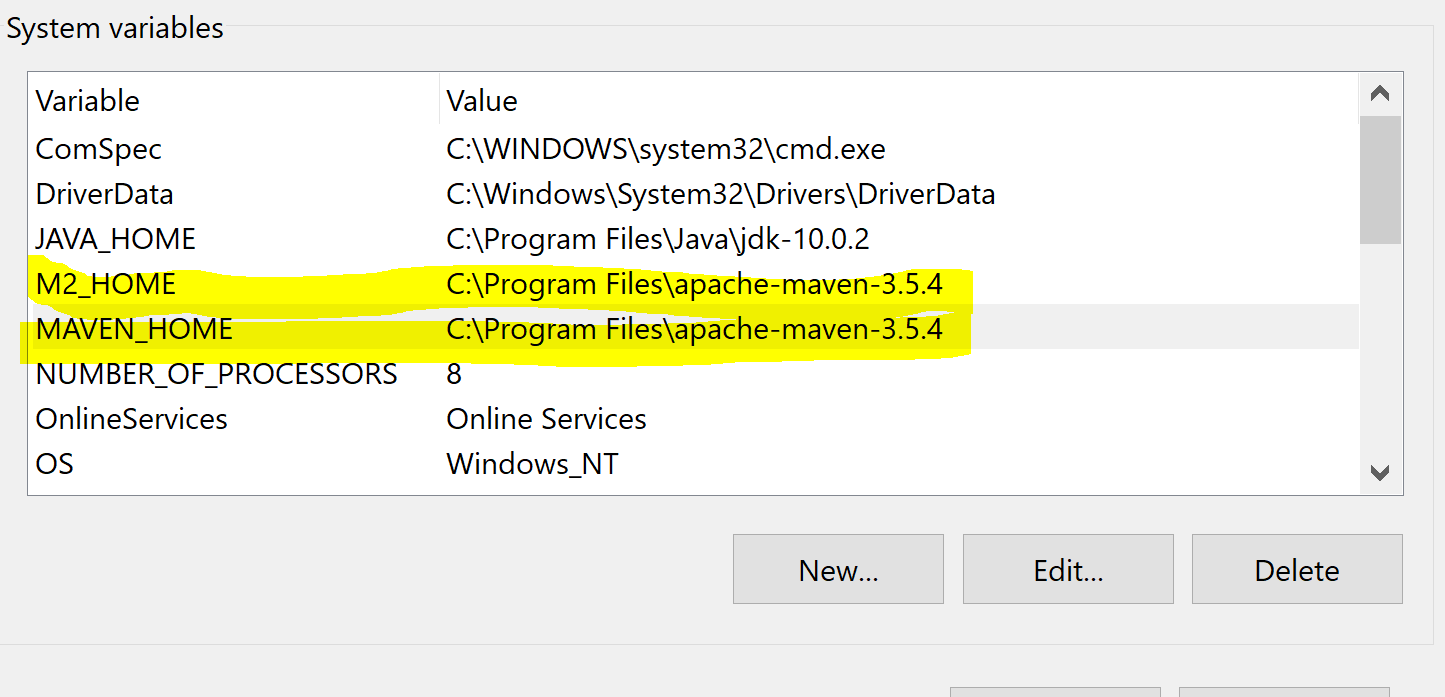
Java development kit a software tool used in creating and running Java Applications. Our Application uses Java as it’s back end so, Java is required to run this Application. This project requires at least JDK 1.8, but it’s advisable to get the latest version

1. Download the latest version of JDK here <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Run the installer.
3. Add JAVA\_HOME to the System Variables. (Check the Node.js section to find out how to get to the system variables. The below example is for windows)



**Maven**

Maven is similar to npm. It is a package manager, and also a project management software tool. It is used in installing dependencies and packaging your software.

1. Download Apache Maven Zip file from the site. <https://maven.apache.org/download.cgi>
2. Extract the zip file to your desired location
3. Add M2\_HOME and MAVEN\_HOME to the system variables. (Check Nodejs section for Mac)

Pro tip- Remember to run **java -version and mvn—version**  to ensure both applications were installed correctly

***Apache Tomcat (Recommended)***

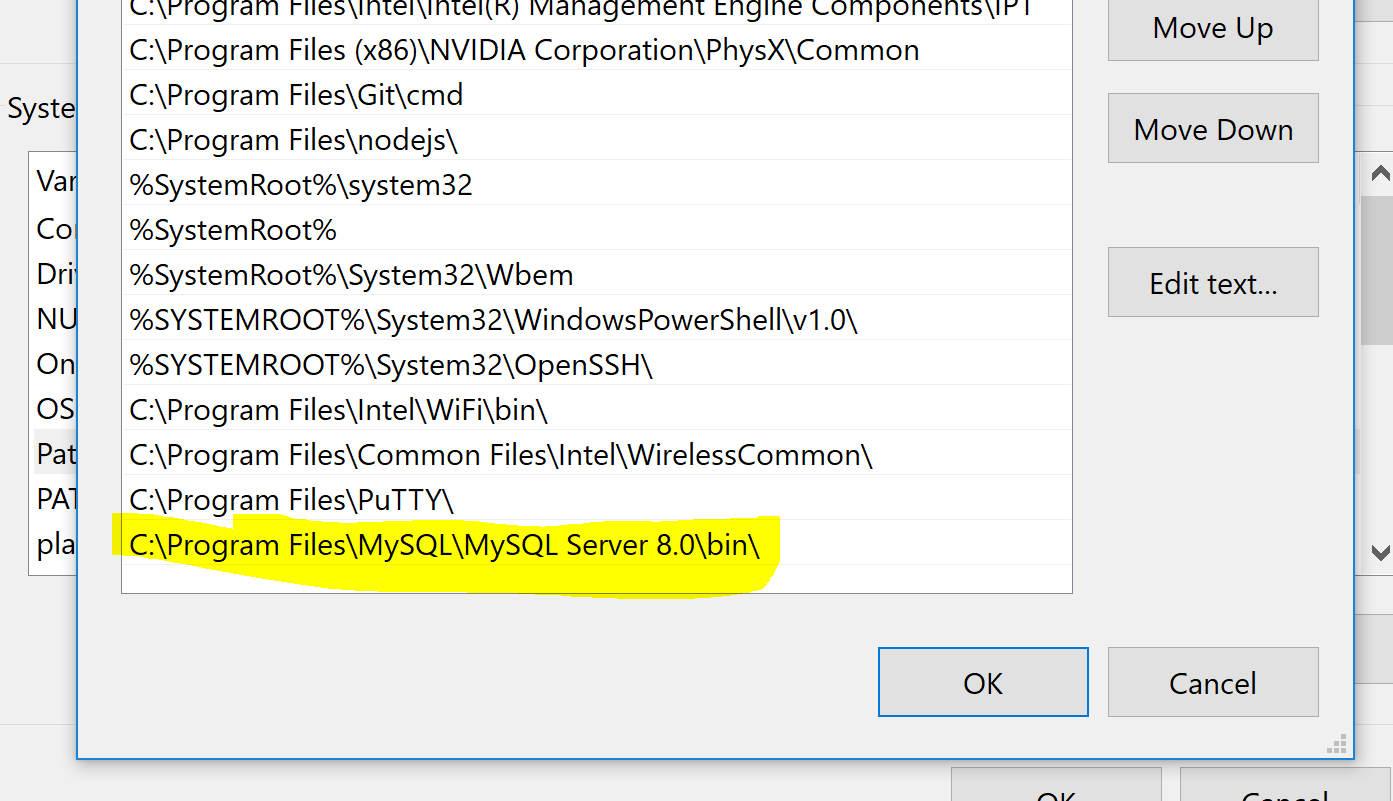
Tomcat is a server that runs on Java. It is used to serve our web application. It is not an absolute requirement for this application but useful nonetheless

1. Download the Zip file from here <https://tomcat.apache.org/download-80.cgi>
2. Extract the zip folder to a destination of your choice.

**MySQL**

MySQL is a database management software tool. It is used to store the application data

1. Download MySQL community server here <https://dev.mysql.com/downloads/mysql/>.
2. Run the installer.
3. Make sure you set a password for the root user during installation
4. When you are done add MySQL to the system variables.



1. Navigate to the project on your computer.
2. Navigate to the folder src/main/webapp/WEB-INF/database
3. Log in to mysql server typing the command mysql -u root -p
4. When it prompts you for your password enter your password.
5. When you successfully log in run the command **source stats.sql** (stats is the name of the database)
6. Run this query **GRANT ALL PRIVILEDGES ON stats.\* To 'loadtester'@'localhost' IDENTIFIED BY 'loadtester';** This creates a new user loadtester and password loadtester, which is the login info for the database in the application. However, If you can’t even, just change the username and password in the source file DatabaseClient.java.
7. You’re all set. Now you need to find out how to run it

**Running the Application**

**Command Line method**

If you didn’t install Apache Tomcat, or you don’t plan on installing other software IDE tools like IntelliJ or eclipse, this version is for you.

**First steps:**

1. Go to the project. Go the the folder, angular-project. Open up proxy.conf.json. Ensure this is set

**{**

**"/v1/\*"**: {  
 **"target"**: **"http://localhost:8080/loadtester"**,  
 **"secure"**: **false**}

}

1. It is. Good! Move on. You’ll earn how to run the front end later in this section.

**No Tomcat installation**

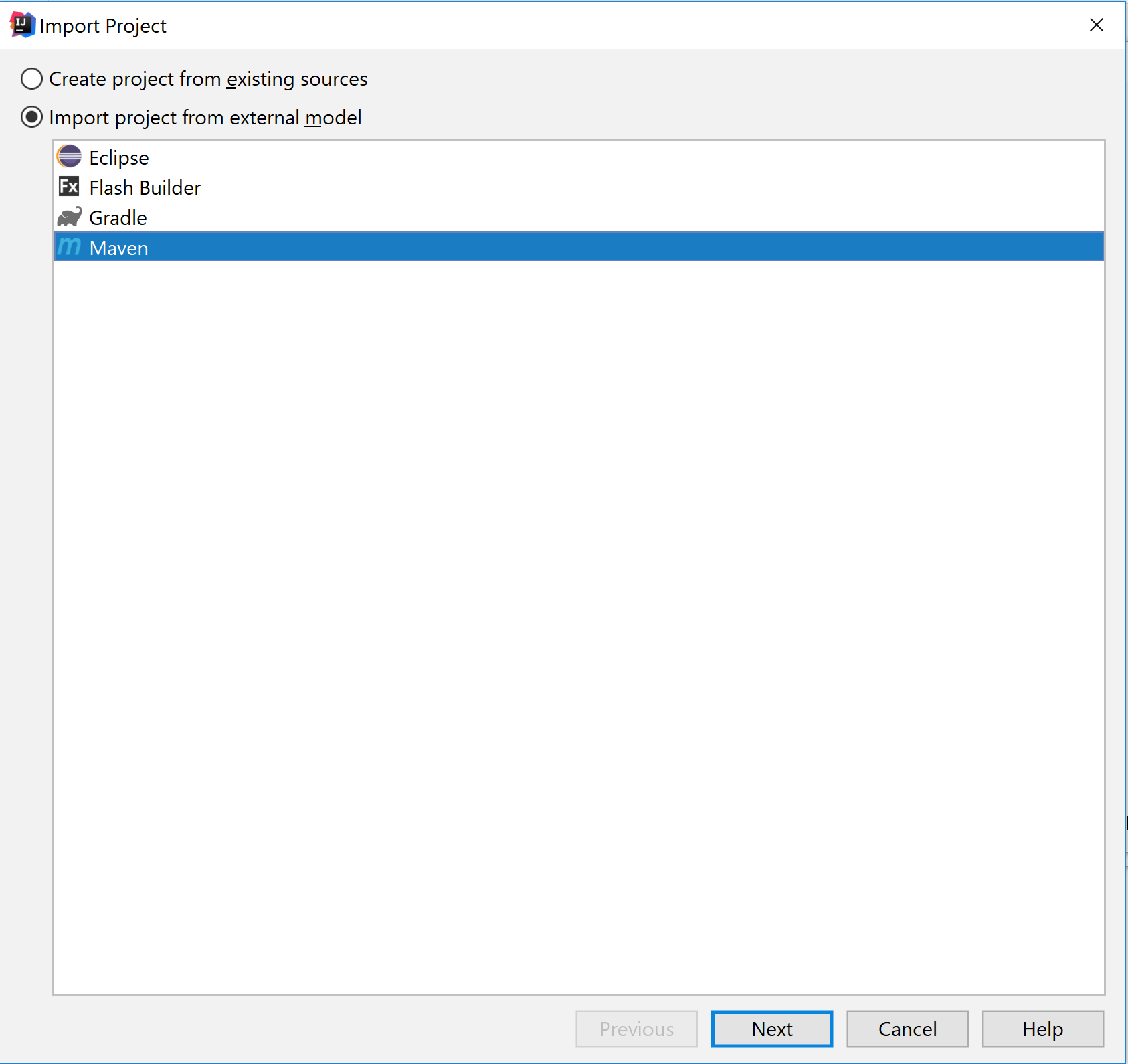
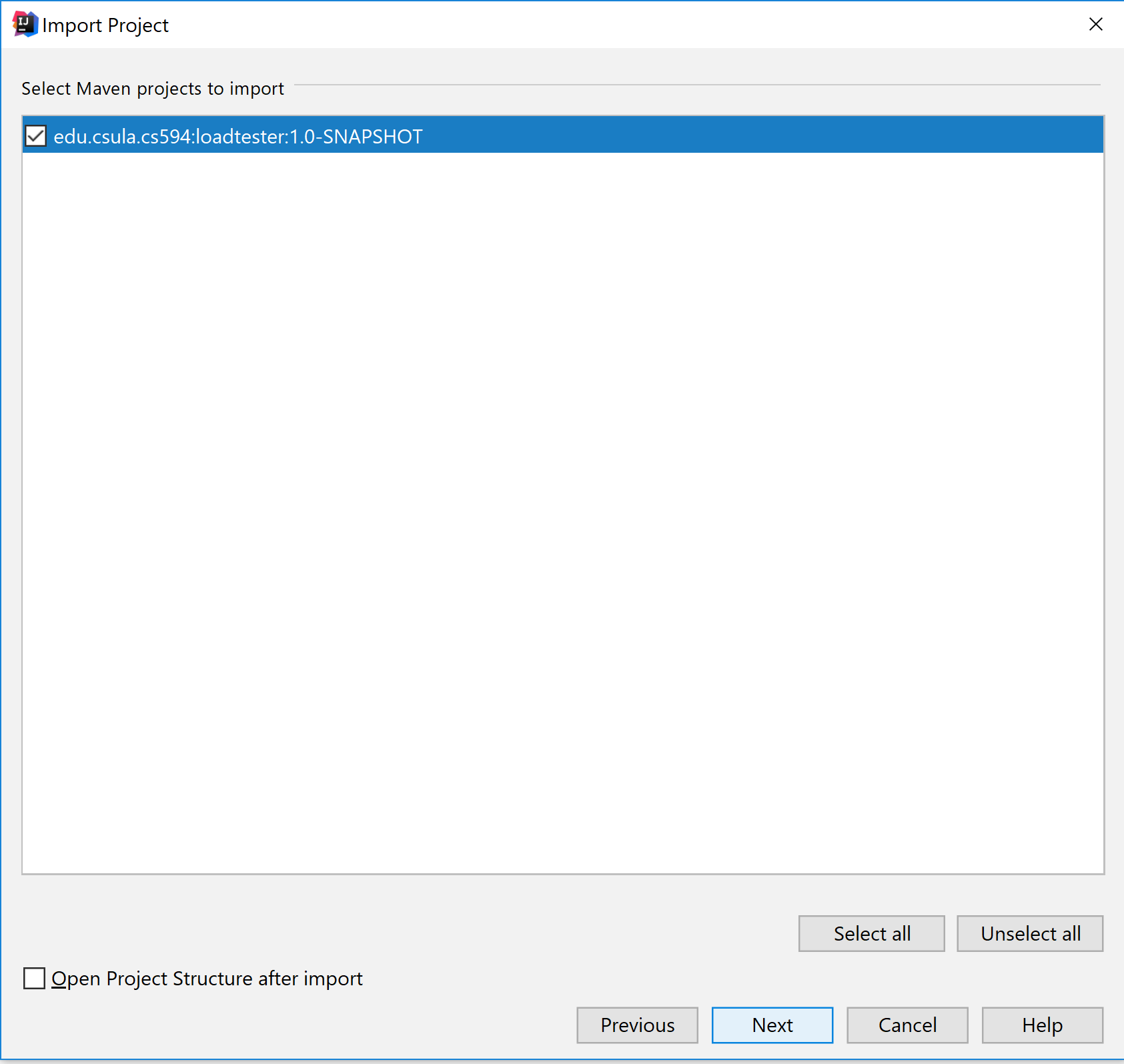
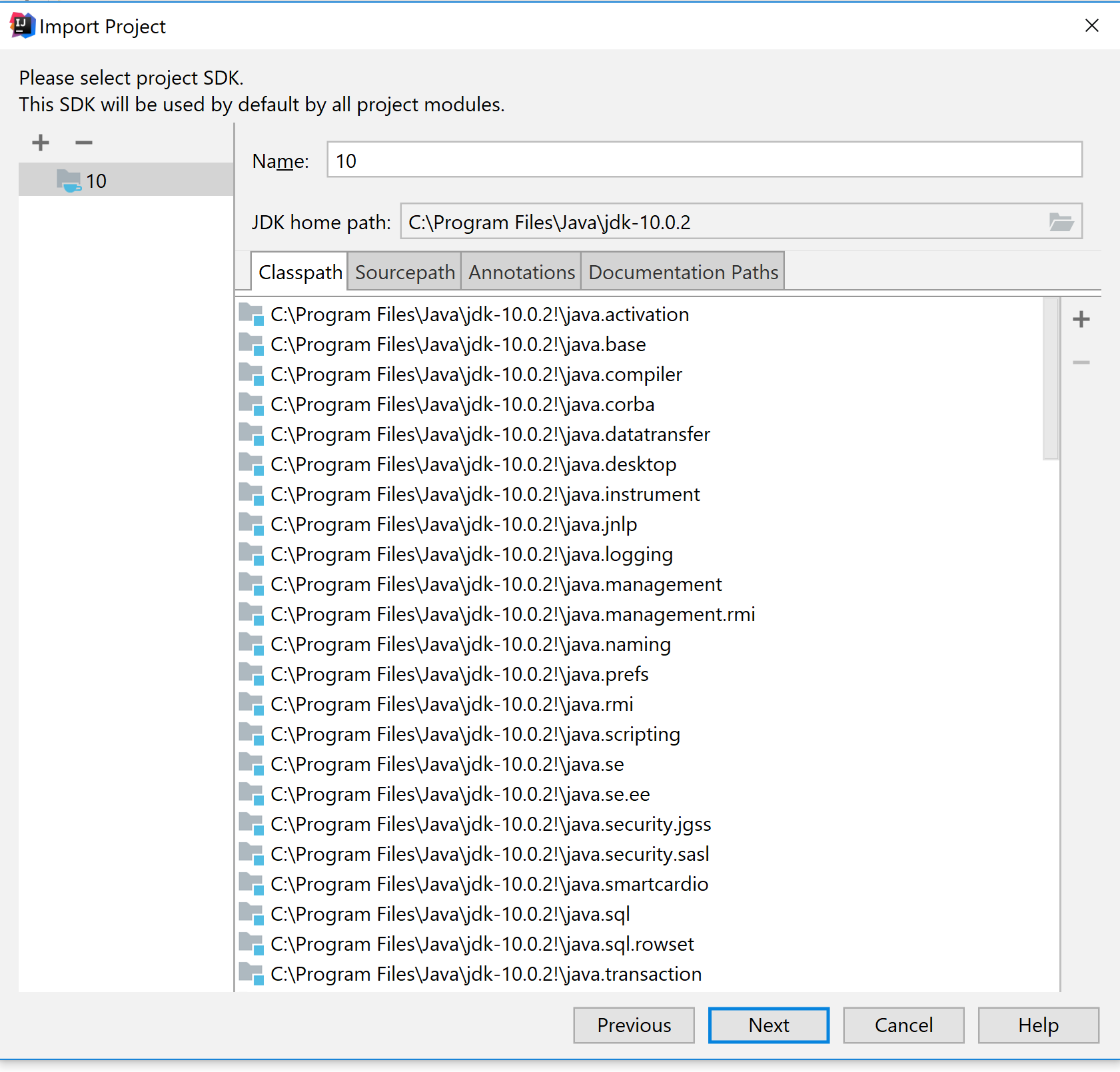
1. Just navigate to the root folder of the project and run **mvn install**, then **mvn tomcat:run**. Finite. The backend is up and running.
2. No really, that’s it. You’ll learn how to install the front end later

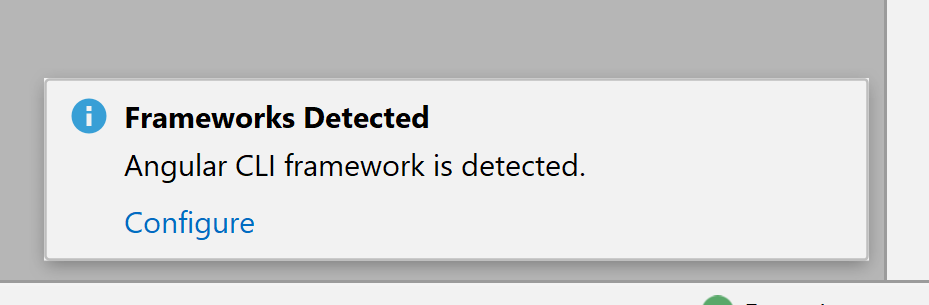
**Tomcat Installation**

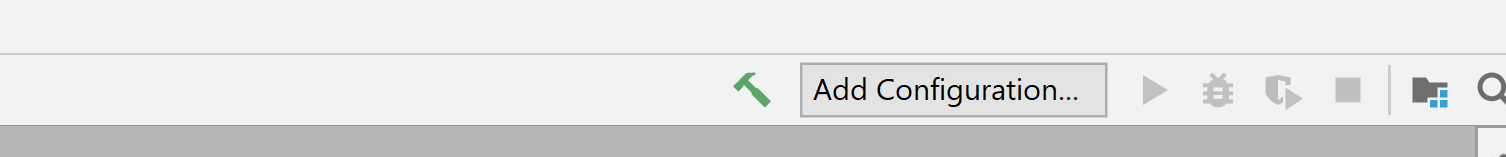
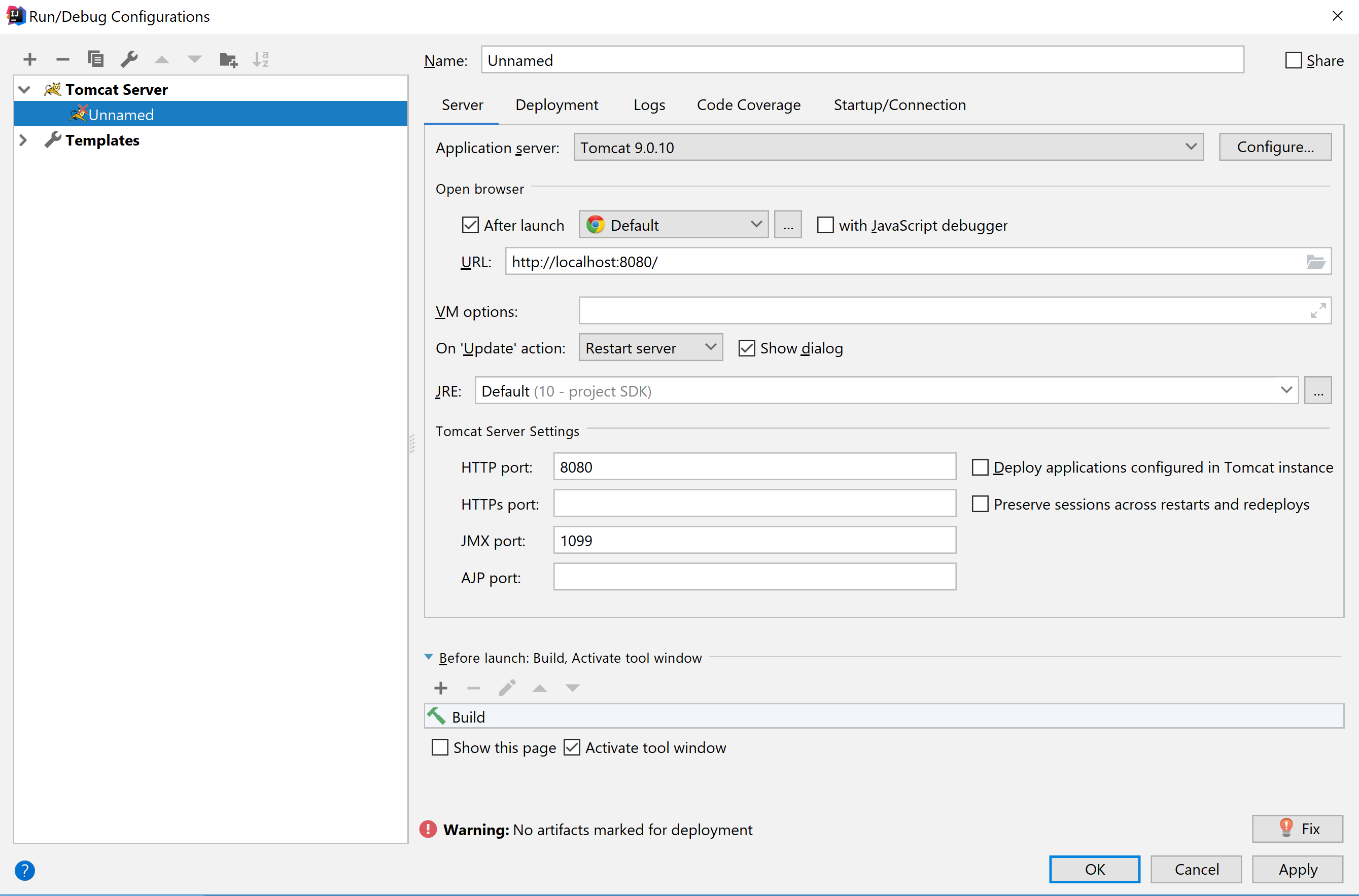
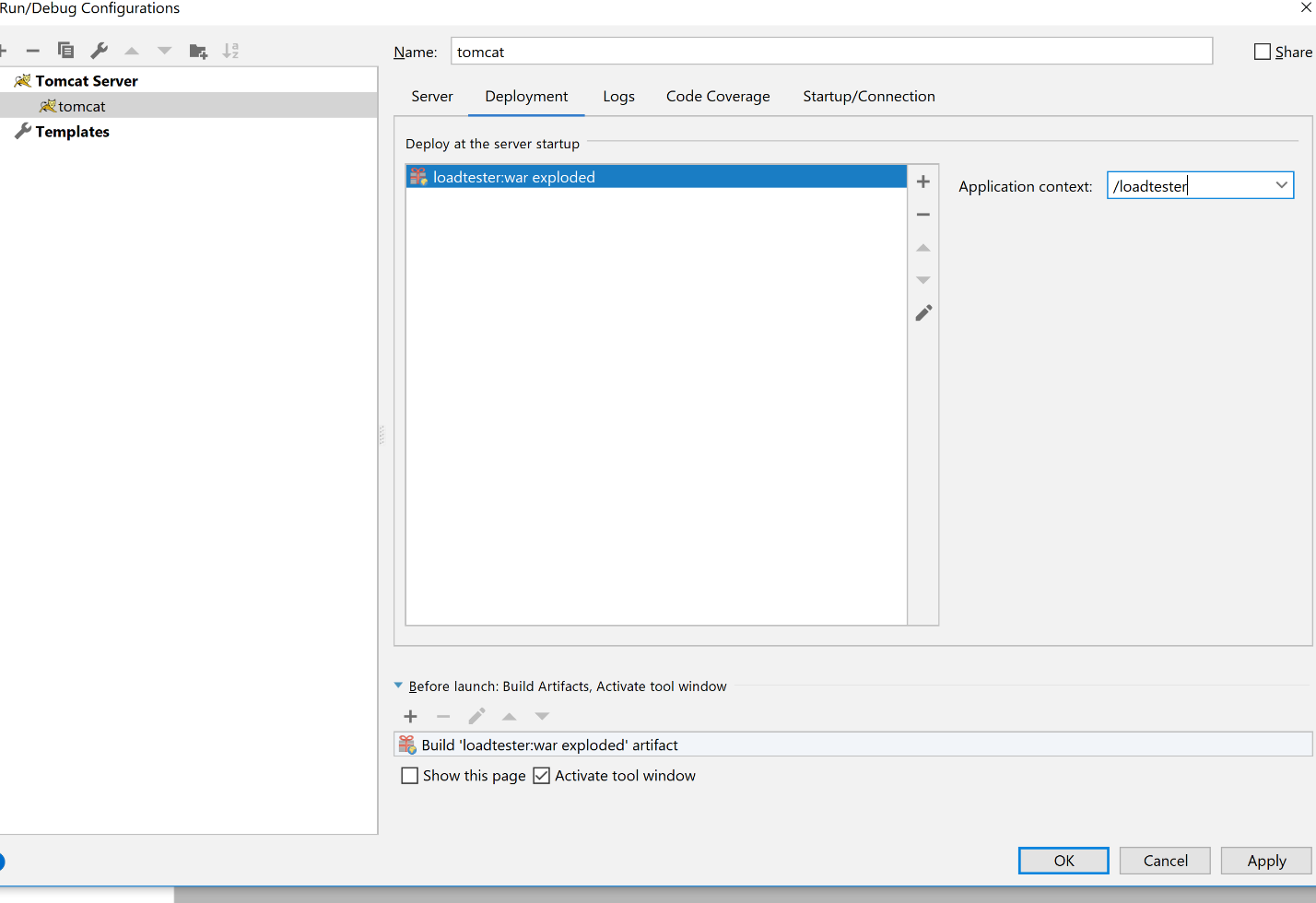
1. Navigate to the root folder of the project and run **mvn package.**  When the command finishes, navigate to target, you’ll see a file called *loadtester.war,* copy that file to the *webapps* folder of your tomcat installation.
2. Navigate to the bin folder of your tomcat installation on the command line. Type **catalina.sh start.** Your application is now running.
3. To stop the application type **catalina.sh stop**

**IntelliJ IDEA Version**

IntelliJ IDEA is a Java integrated development environment for developing computer software. It is developed by JetBrains. It has a free version (Community) and a licensed edition. This manual will be illustrating the licensed edition. If you are a student, you can get the licensed edition for free.

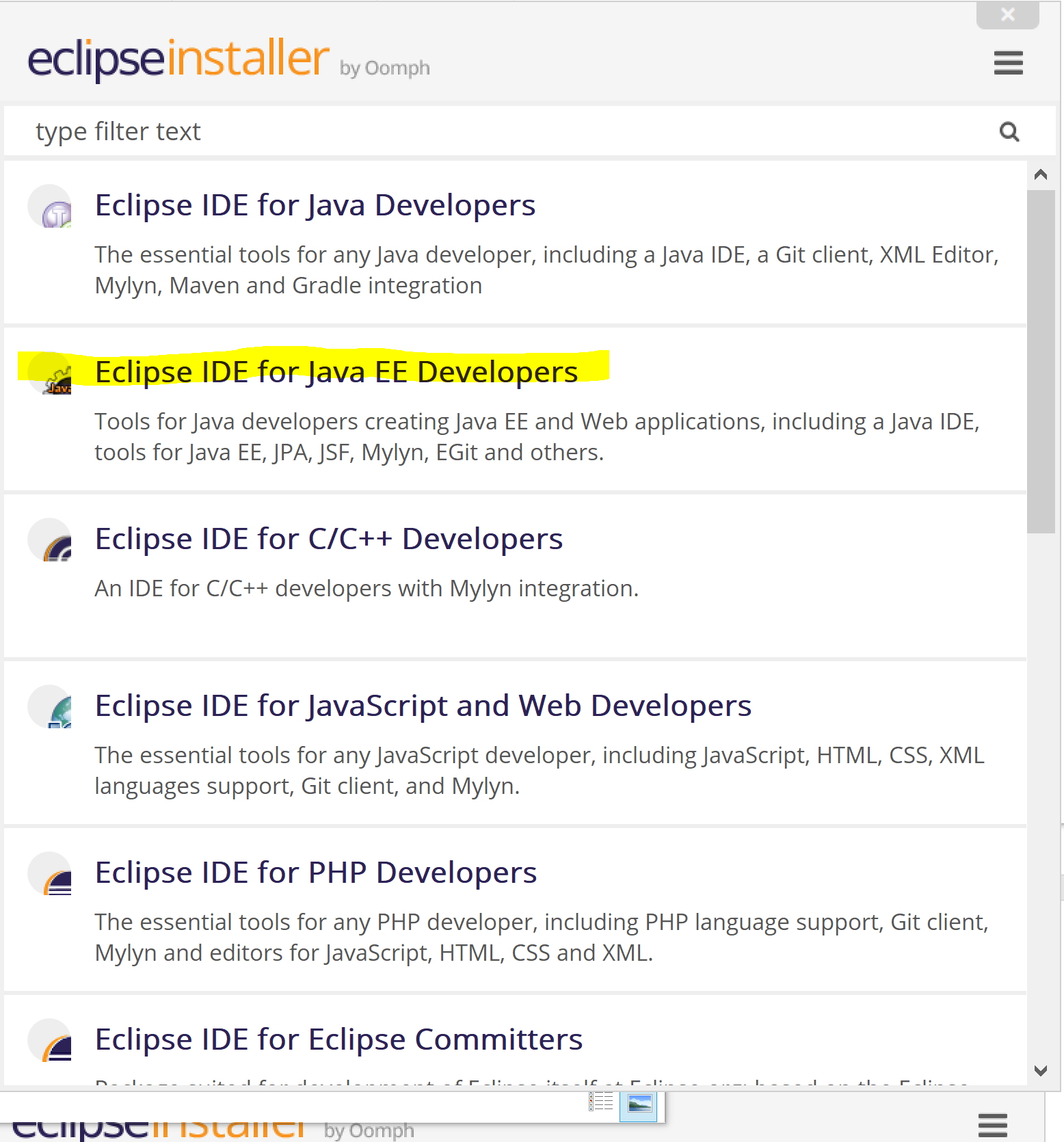
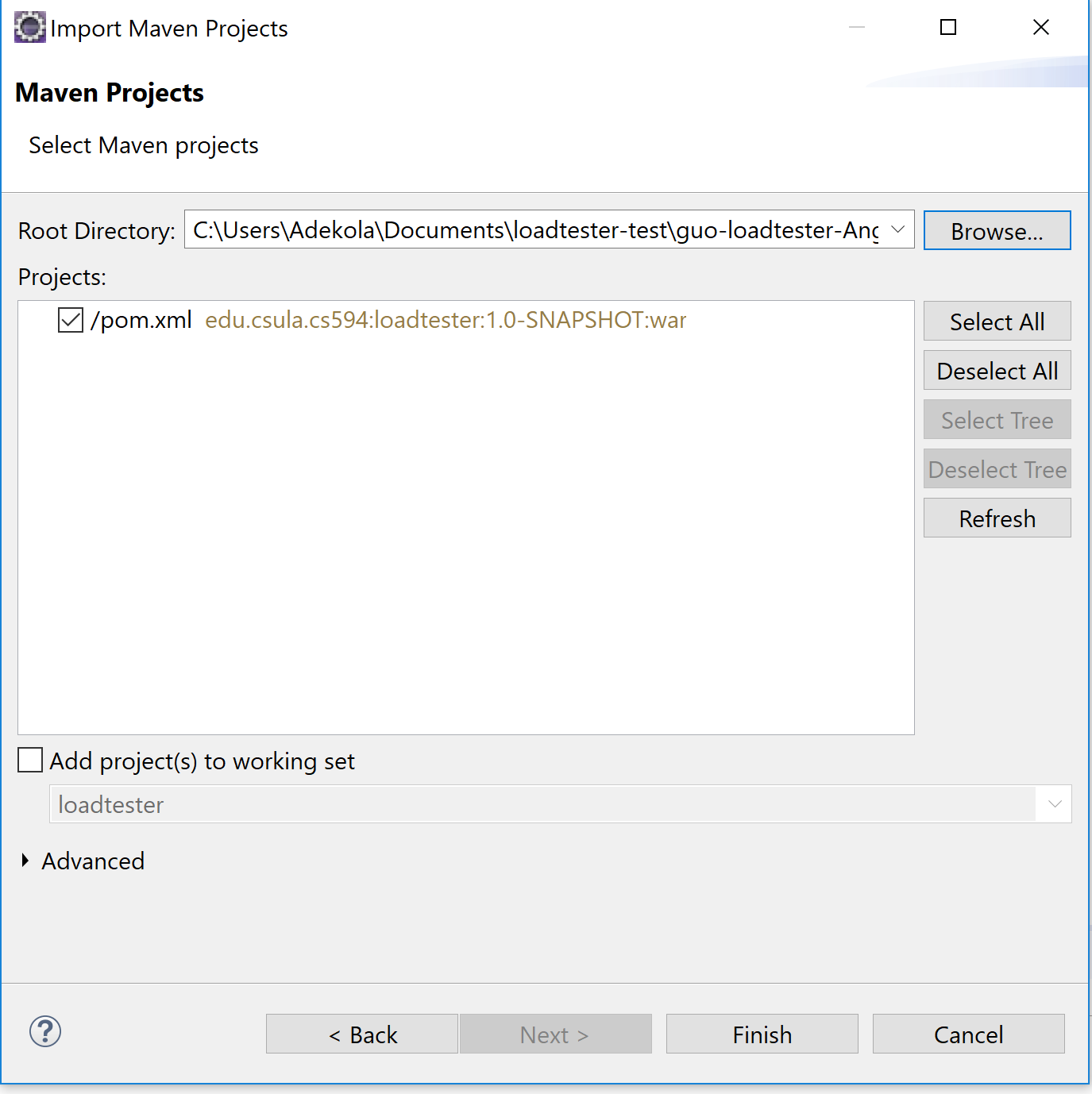
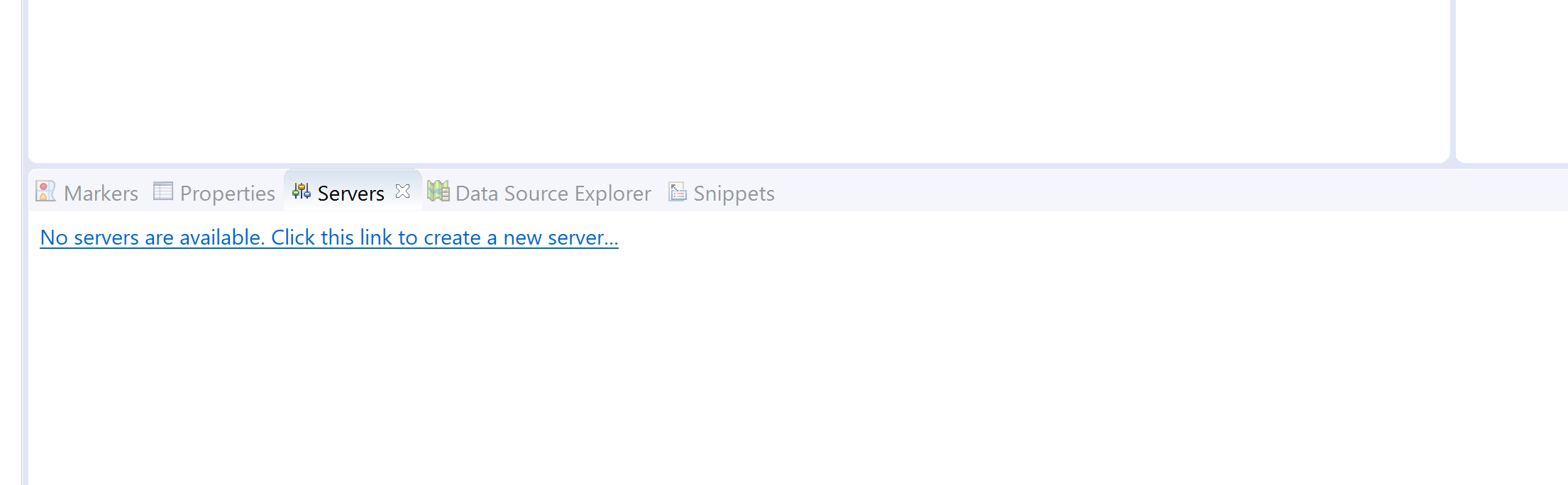
1. Download the installer here <https://www.jetbrains.com/idea/download/>
2. Install the software. When it is done, open Intellij.
3. Choose import project. Select Maven
4. Choose next, select the SNAPSHOT 
5. Click next. Make sure your class path is selected
6. Click next then finish.
7. You might see a message “Angular Frameworks Detected” click import and follow the instructions

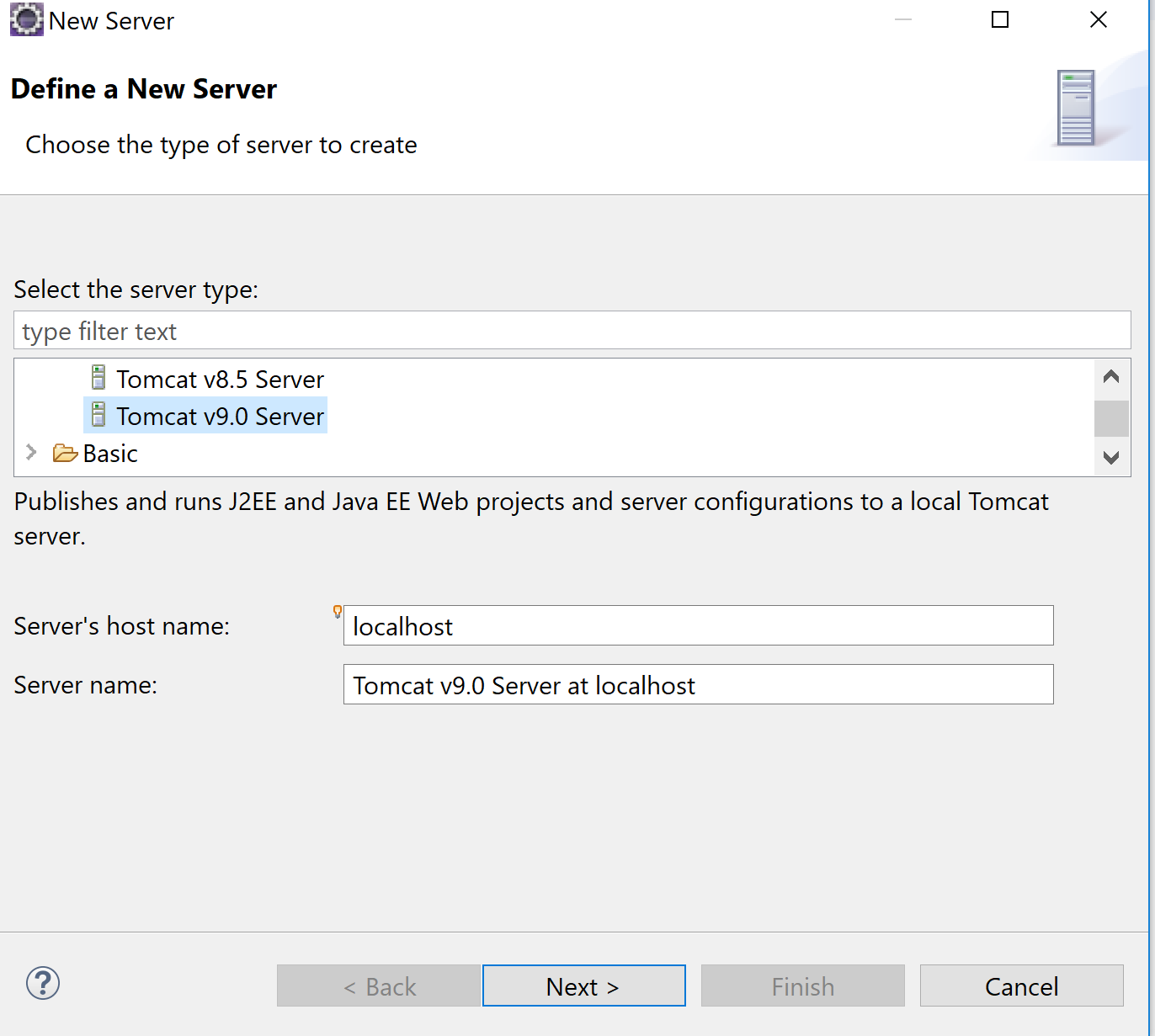


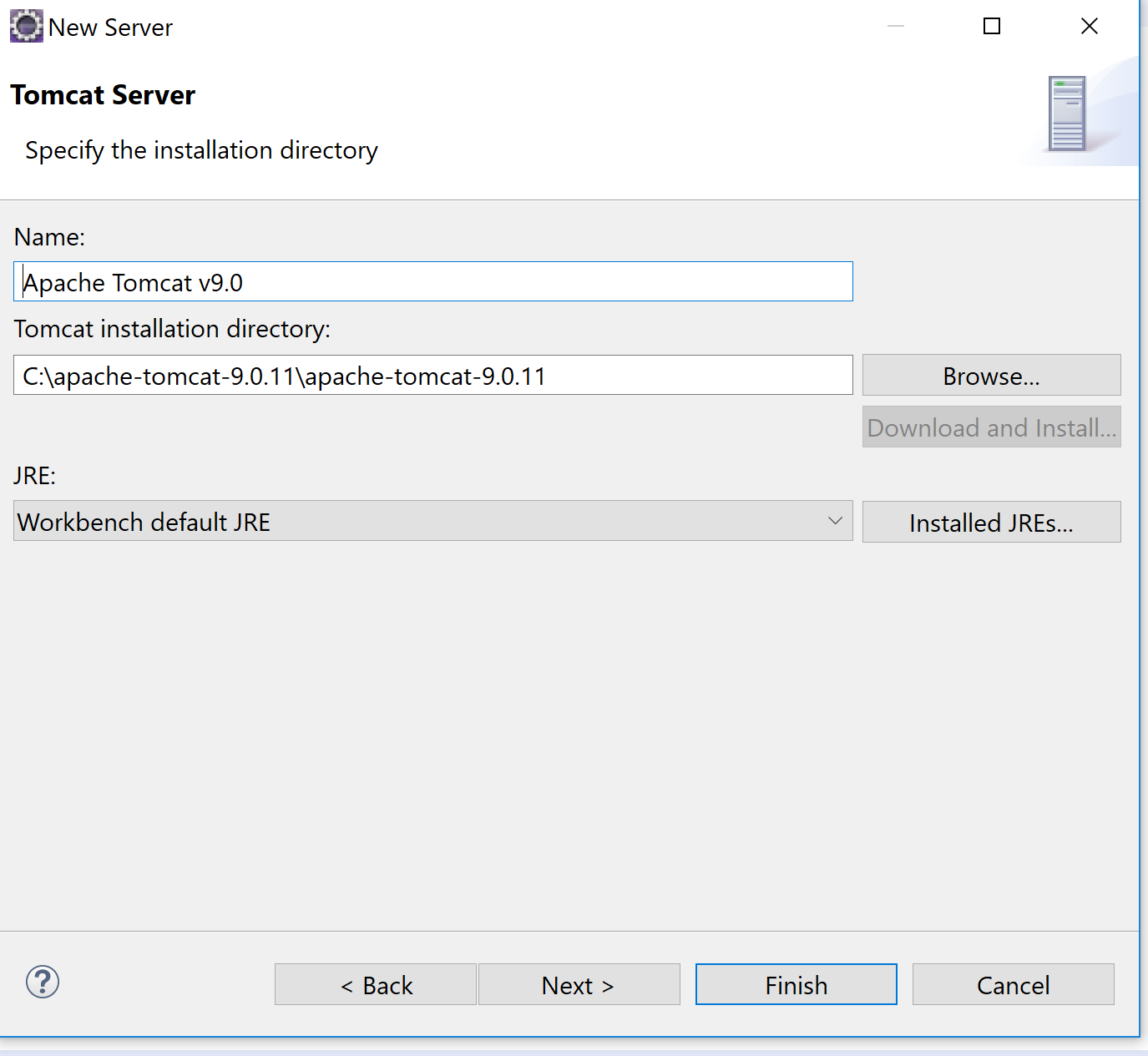
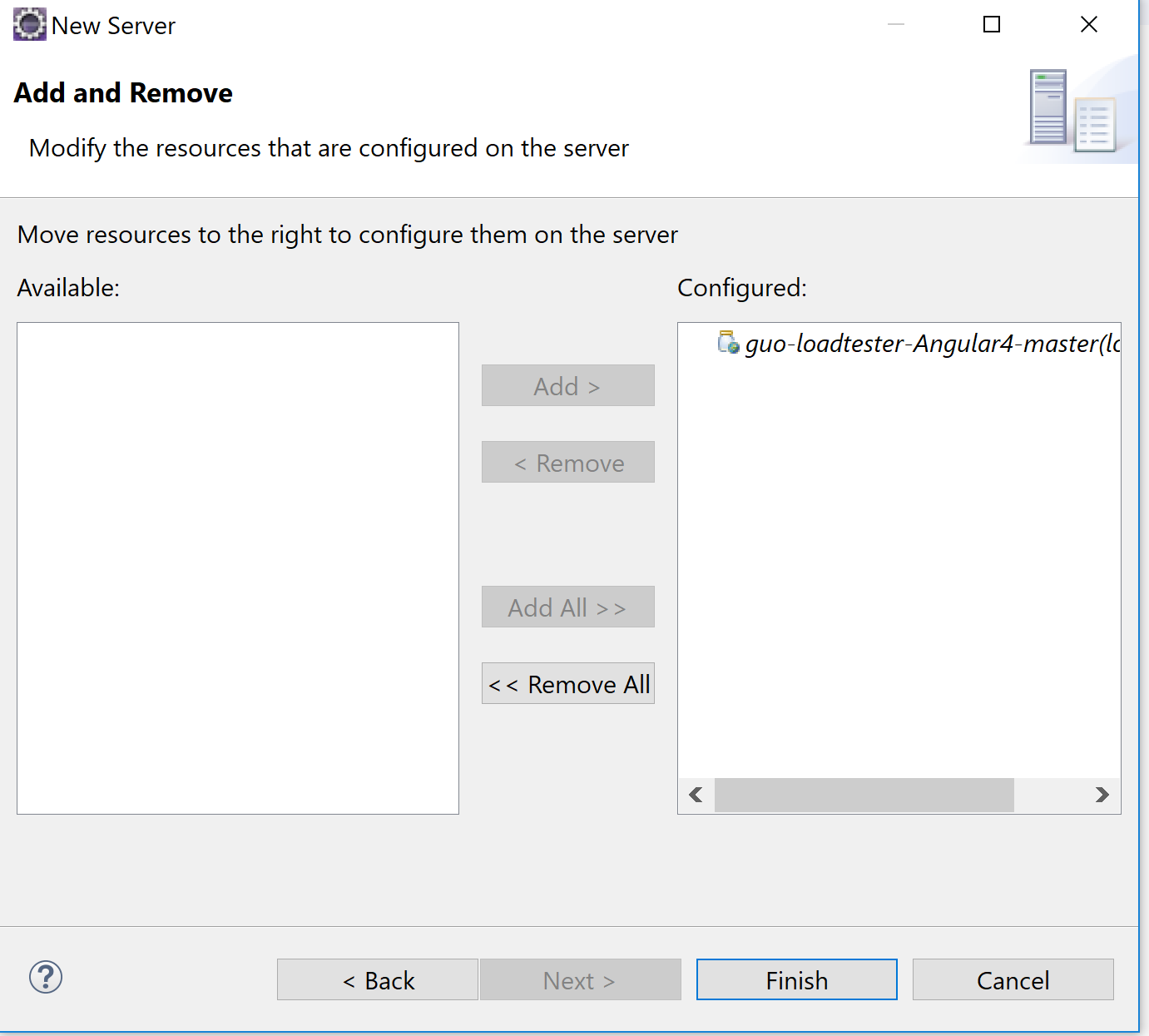
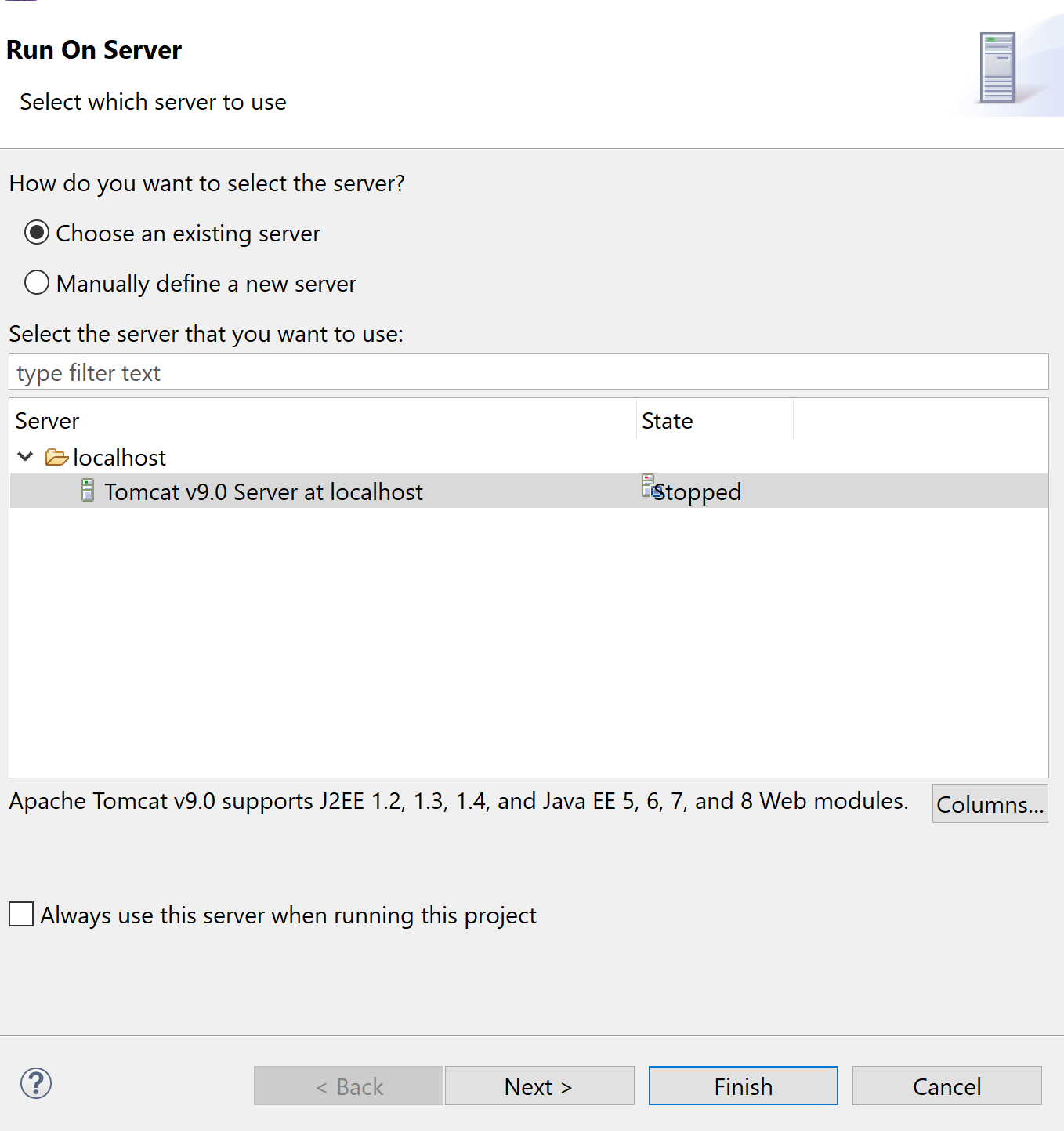
1. Go to add configurations at the top right of the screen. Click the **+** button and select Tomcat, then local
2. Choose the directory of your tomcat installation, make sure JRE directory is set. 
3. On the warning that says ‘No artifacts marked for deployment” choose fix and select loadtester:war exploded.
4. In the Deployment Tab, set the Application Context to loadtester
5. Click apply to save changes. You’re done.

**Eclipse JEE Version**

Eclipse for Java Enterprise Edition is an IDE used in building and developing web applications. It is similar to IntelliJ except that it’s free

1. Download the Eclipse Installer here <http://www.eclipse.org/downloads/packages> choose “Eclipse IDE for Java Developers”.
2. Run the installer. Select the option highlighted in yellow below 
3. It should download some packages, wait for it to download then choose all the defaults during the installation. Make sure to select your workspace when Eclipse launches.
4. After launching eclipse, go to File 🡪 Import 🡪Maven 🡪Existing Maven projects
5. Browse the location of the Project Directory and click on it.
6. You should see the pom.xml file displayed. Click finish.
7. When the project finishes loading, right-click on the project, go to Maven🡪 Update Project. This will install all the dependencies from pom.xml
8. In the bottom navigation tab, got to servers, then add a server
9. In the tab opened go to Apache, then select your Tomcat installation. Mine is 9.0. Click next.



1. Browse to your tomcat installation. Select it. **Don’t** click finish yet. Click next 
2. Move your application into the server configuration. Then click finish. 
3. Run your project. Select the run button, Click on run on server, and select your installed server. Click next 
4. Make sure your project is configured on the server (At the right) Click finish.
5. Your project is available on <http://localhost:8080/loadtester>

**Running the Angular Project**

Setting up the project to run on Angular is straightforward. Ensure you have installed npm and Angular CLI, as directed in the tutorial above.

1. Navigate to the root of the project. The Angular project is located in the folder angular-project. With the command line, navigate there.
2. Run the command **npm install**. This will install all the dependencies the project uses from package.json
3. When the packages finishes installing, go to proxy.conf.json and ensure this value is set exactly as it this

**{**

**"/v1/\*"**: {  
 **"target"**: **"http://localhost:8080/loadtester"**,  
 **"secure"**: **false**}

**}**

1. Run **ng serve** to run the front end.
2. That’s it.

**Git Installation**

It is a useful tool. It is a version control used for tracking changes in files, and aiding collaboration with multiple people for the same project. Github is a web based hosting service that uses git. To sign up for a github account to go <https://github.com/join>

To install git

1. Download the installer from this url <https://git-scm.com/downloads>
2. Run the installer. This should add all the path variables. If it doesn’t, follow the instructions from previous tutorials
3. Ensure git is installed by running **git --version**

**Extra Steps**

To set up your git installation, you need to tell git who you are. Run the commands below. Replace the values of the username and email with yours.

git config --global user.name "John Doe"

git config --global user.email johndoe@example.com

**Common commands**

Common git commands are as follows

1. **git clone <repo> -** Used to download a git repository
2. **git pull <remote> <branch> -** Used to fetch changes from a repository branch to your local git repository
3. **git push <remote> <branch> -** Used to upload changes from your local git to the server

A git cheat sheet can be found here

<https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>