

# Front-end App

# Full-stack-application-with-springboot-and-angular

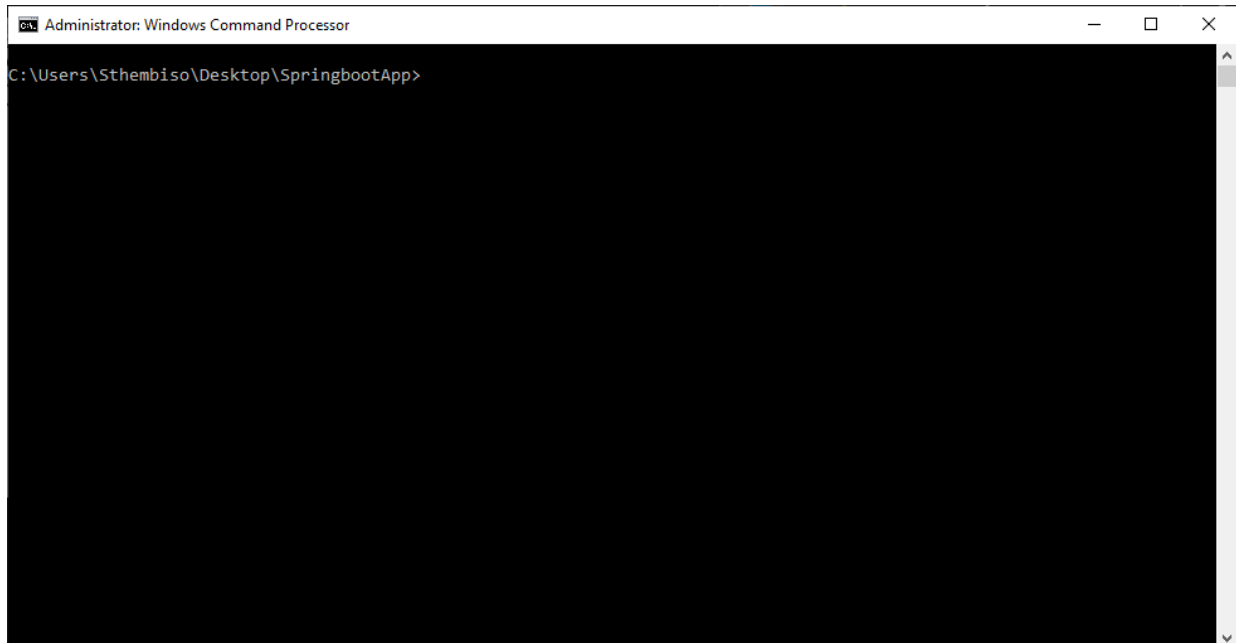
This is a to-do application build with springboot and angular cli

This project was generated with [Angular CLI](https://github.com/angular/angular-cli) version 7.0.3.

# Setting up your local development

Create a directory/folder named "**SpringbootApp**"

Navigate to "**SpringbootApp**" folder with your terminal or command prompt. On windows, ensure that you running your command prompt as an administrator



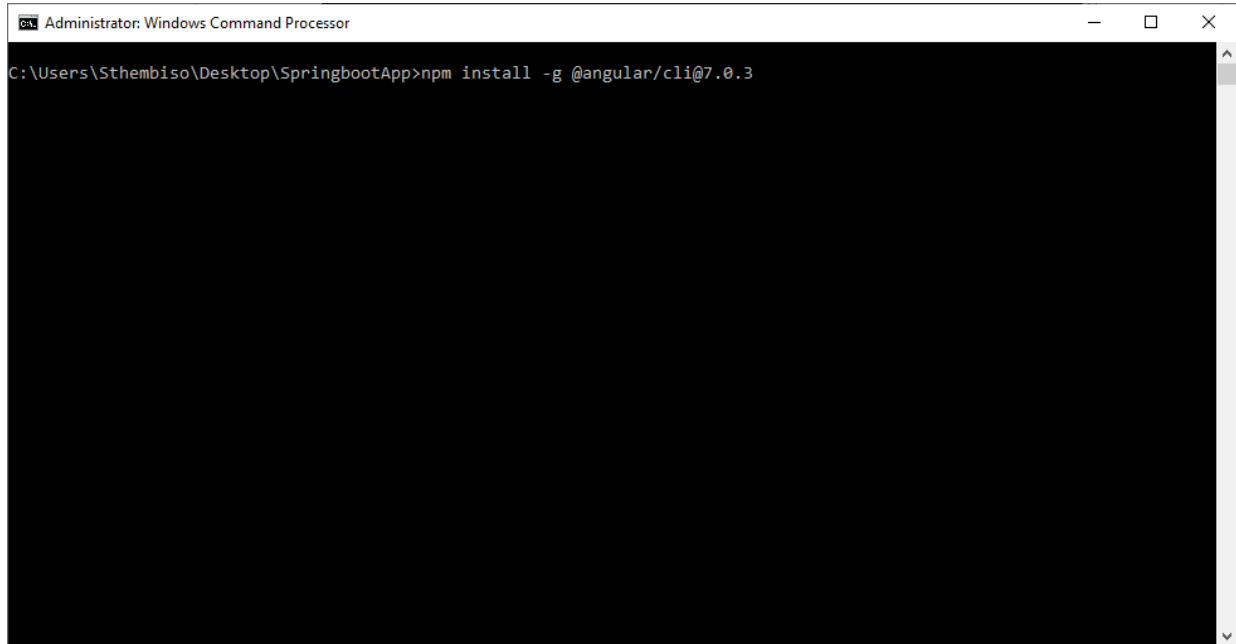
Ensure that you have the following on your local development

- Run "**node -v**" to check your node version. Ensure that you have V8.12.0 or higher
- Run "**npm -v**" to check your node version. Ensure that you have 6.4.1 or higher

Ensure that you are in the "**SpringbootApp**" folder on your terminal or command prompt

Run "**npm install -g @angular/cli@7.0.3**". On windows, ensure that you are running your command prompt as an administrator

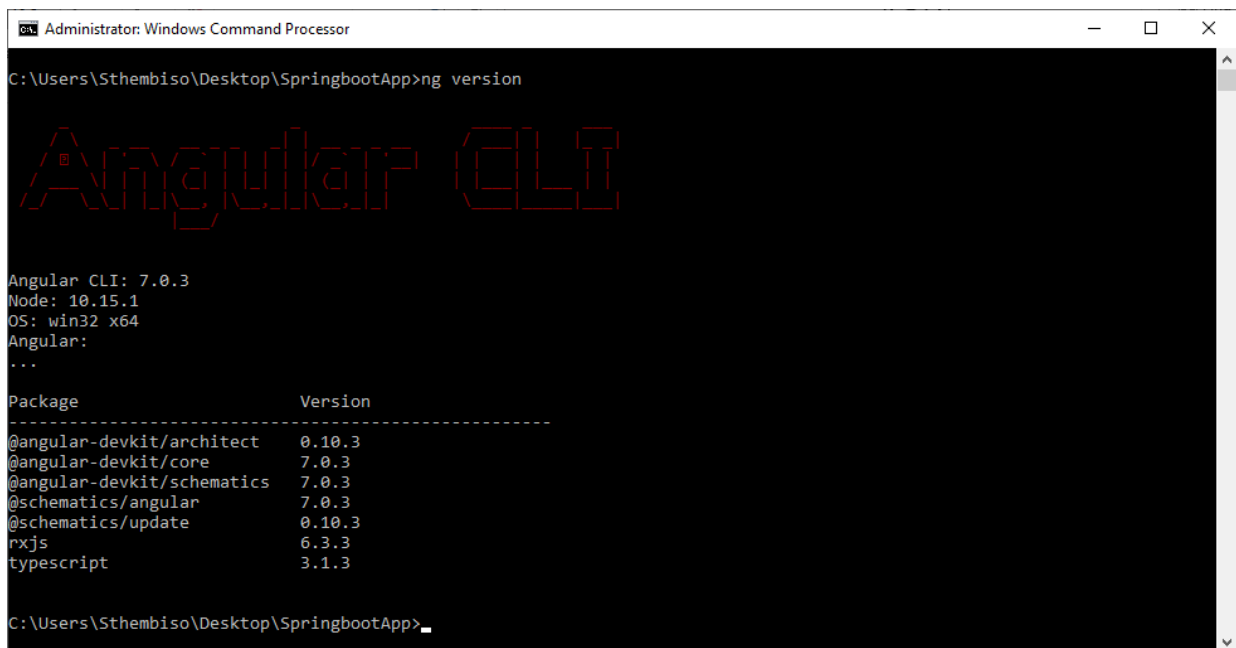
On MacOS or Linux, Run "sudo npm install -g @angular/cli@7.0.3" (Terminal might prompt you for root user password. Enter your root user password to download Angular CLI to your local development)



```
Administrator: Windows Command Processor
C:\Users\Sthembiso\Desktop\SpringbootApp>npm install -g @angular/cli@7.0.3
```

NPM will download the all the npm packages for you

Run "**ng version**" on your terminal to see the Angular CLI version you have installed



```
Administrator: Windows Command Processor
C:\Users\Sthembiso\Desktop\SpringbootApp>ng version

Angular CLI
Angular CLI: 7.0.3
Node: 10.15.1
OS: win32 x64
Angular:
...

Package      Version
-----
@angular-devkit/architect 0.10.3
@angular-devkit/core      7.0.3
@angular-devkit/schematics 7.0.3
@schematics/angular       7.0.3
@schematics/update        0.10.3
rxjs               6.3.3
typescript         3.1.3

C:\Users\Sthembiso\Desktop\SpringbootApp>
```

Run **'ng new "todo"'** on your terminal to see the Angular CLI version you have installed

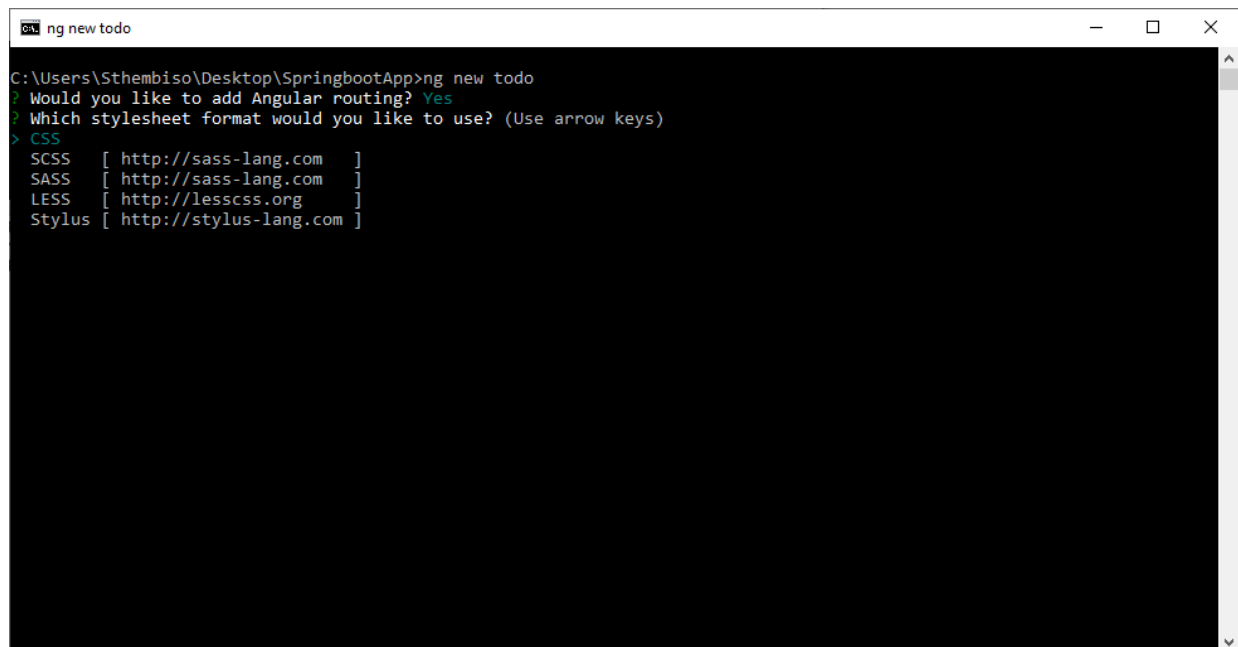
Terminal or Command Prompt will ask you these questions:

For Routing, Choose Yes

? Would you like to add Angular routing? Yes

Choose CSS

? Which stylesheet format would you like to use? CSS

A screenshot of a terminal window titled "ng new todo". The terminal shows the command "ng new todo" being executed at the path "C:\Users\Sthembis\Desktop\SpringbootApp". It then displays two prompts: "Would you like to add Angular routing?" with "Yes" entered, and "Which stylesheet format would you like to use? (Use arrow keys)" with "CSS" entered. Below these, a list of options is shown: SCSS [http://sass-lang.com], SASS [http://sass-lang.com], LESS [http://lesscss.org], and Stylus [http://stylus-lang.com].

```
ng new todo
C:\Users\Sthembis\Desktop\SpringbootApp>ng new todo
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SCSS [ http://sass-lang.com ]
  SASS [ http://sass-lang.com ]
  LESS [ http://lesscss.org ]
  Stylus [ http://stylus-lang.com ]
```

# When you it's done installing

Clone the front-end folder from the github repository to your local development environment and unzip it.

Copy the all the files and folders on the front-end folder and replace the file and folders on your todo folder

After replacing the files and folders, navigate to your todo folder with your terminal or command prompt and run the following command

Run “**ng serve**” for a dev server.

Navigate to `http://localhost:4200/` on your internet browser. The app will automatically reload if you change any of the source files.

# The following commands will help you will developing the app

## Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.

## Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory. Use the `--prod` flag for a production build.

## Running unit tests

Run `ng test` to execute the unit tests via [Karma](https://karma-runner.github.io).

## Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via [Protractor](http://www.protractortest.org/).

## Further help

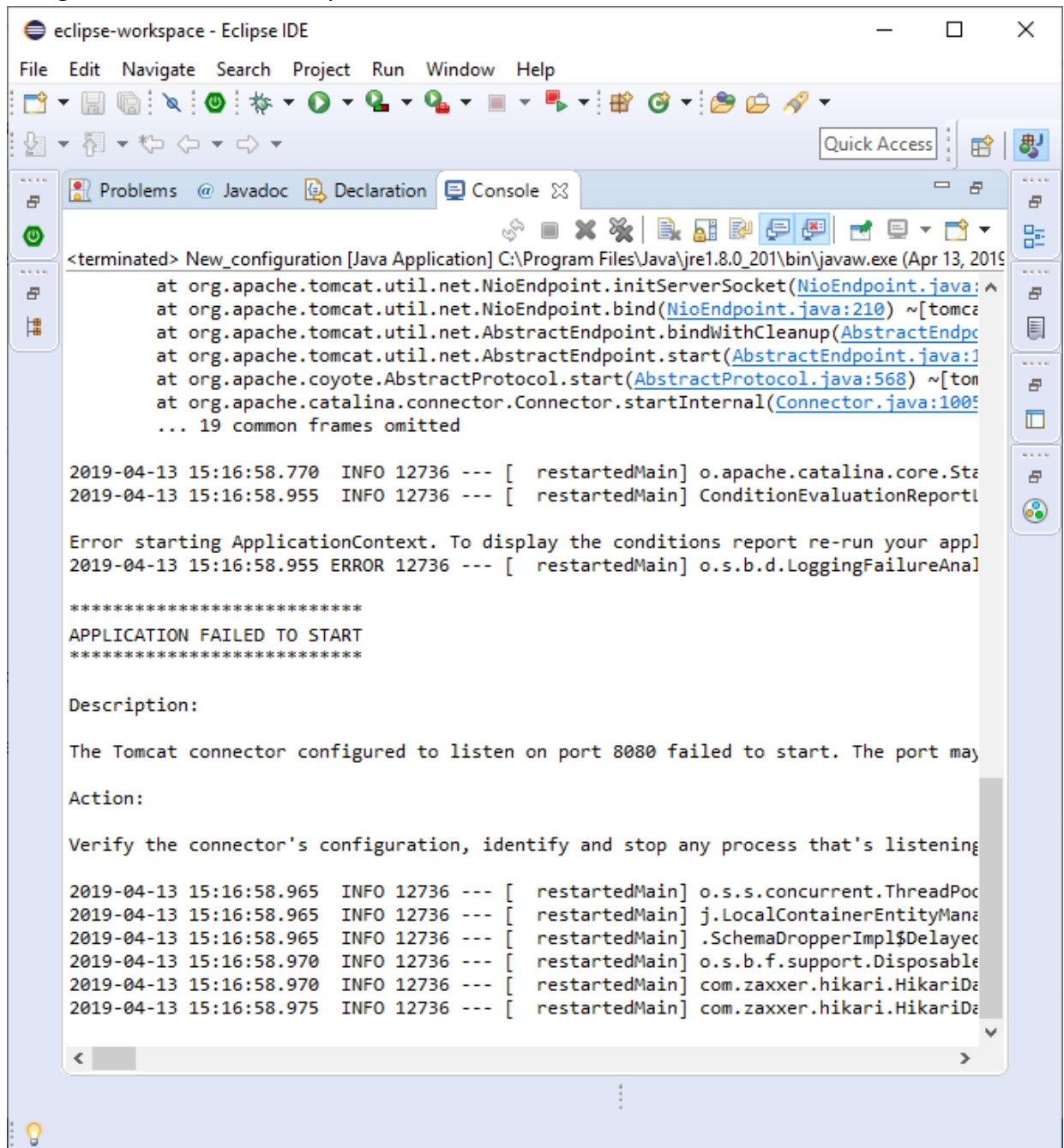
To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI README](https://github.com/angular/angular-cli/blob/master/README.md).

## Restful Web Service

Clone the Restful-service from the repository and unzip it

- Download and install Java JDK(Java Development Kit)  
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Download and Install Eclipse <https://www.eclipse.org/downloads/>

1. Open Eclipse
2. Navigate to File then click Import



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The console output displays a stack trace for a terminated Java application, followed by log messages and an error message indicating that the application failed to start. The error message states: 'Error starting ApplicationContext. To display the conditions report re-run your application in debug mode.' Below this, a detailed description of the error is provided, stating: 'The Tomcat connector configured to listen on port 8080 failed to start. The port may already be in use. Consider using a different port, e.g. 8081. See the manual page for the server (tomcat) for more details.' The action recommended is to 'Verify the connector's configuration, identify and stop any process that's listening on port 8080, or adjust the configuration to listen on a different port.' The console also shows several log messages from the 'restartedMain' thread, including information about the Tomcat connector, the application context, and the Hikari database connection pool.

```
<terminated> New_configuration [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (Apr 13, 2019)
at org.apache.tomcat.util.net.NioEndpoint.initServerSocket(NioEndpoint.java:210) ~[tomcat-util.jar:8.5.21]
at org.apache.tomcat.util.net.NioEndpoint.bind(NioEndpoint.java:210) ~[tomcat-util.jar:8.5.21]
at org.apache.tomcat.util.net.AbstractEndpoint.bindWithCleanup(AbstractEndpoint.java:100) ~[tomcat-util.jar:8.5.21]
at org.apache.tomcat.util.net.AbstractEndpoint.start(AbstractEndpoint.java:100) ~[tomcat-util.jar:8.5.21]
at org.apache.coyote.AbstractProtocol.start(AbstractProtocol.java:568) ~[tomcat-coyote.jar:8.5.21]
at org.apache.catalina.connector.Connector.startInternal(Connector.java:1005) ~[catalina.jar:8.5.21]
... 19 common frames omitted

2019-04-13 15:16:58.770 INFO 12736 --- [ restartedMain] o.apache.catalina.core.StandardEngine.startInternal()
2019-04-13 15:16:58.955 INFO 12736 --- [ restartedMain] ConditionEvaluationReportListener.handleReport()

Error starting ApplicationContext. To display the conditions report re-run your application in debug mode.
2019-04-13 15:16:58.955 ERROR 12736 --- [ restartedMain] o.s.b.d.LoggingFailureAnalysisReporter.report()

*****
APPLICATION FAILED TO START
*****

Description:

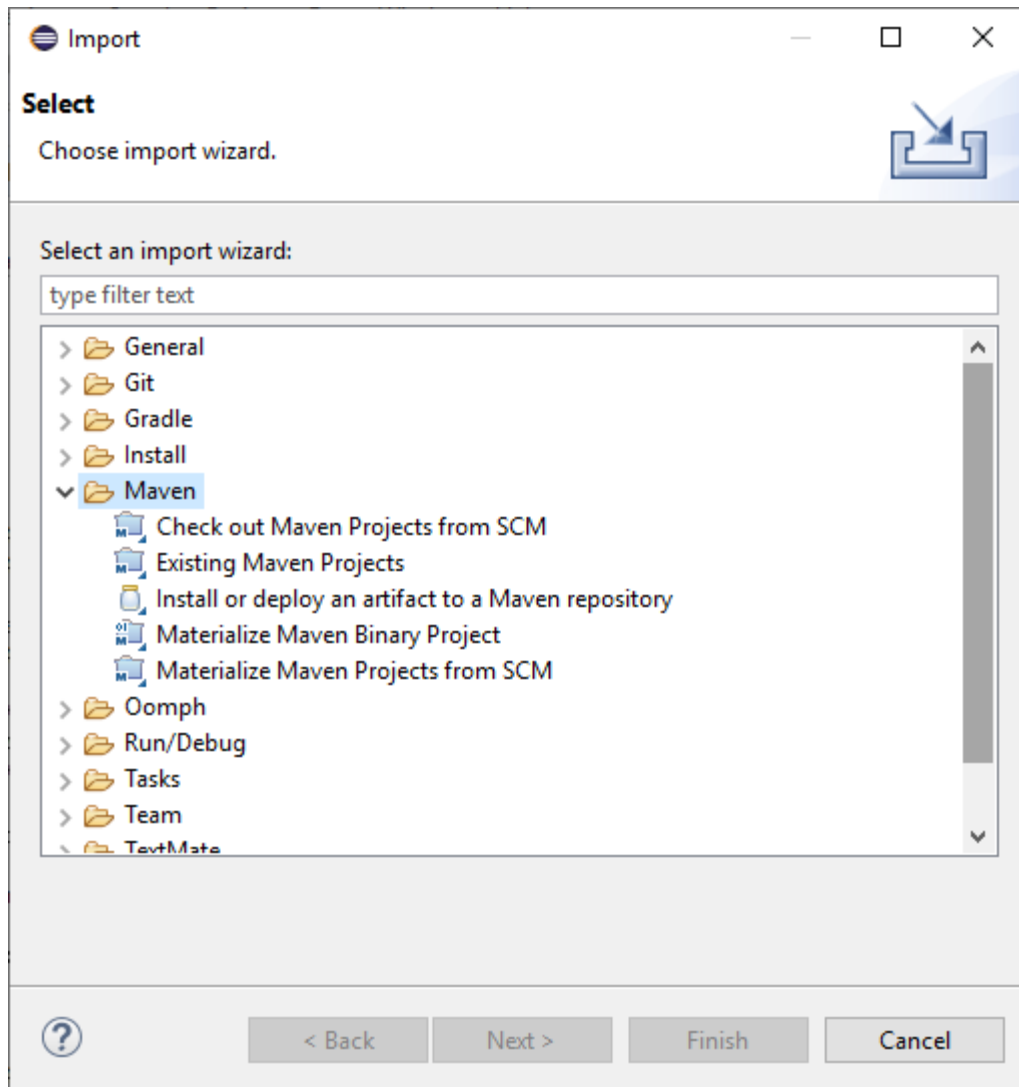
The Tomcat connector configured to listen on port 8080 failed to start. The port may already be in use. Consider using a different port, e.g. 8081. See the manual page for the server (tomcat) for more details.

Action:

Verify the connector's configuration, identify and stop any process that's listening on port 8080, or adjust the configuration to listen on a different port.

2019-04-13 15:16:58.965 INFO 12736 --- [ restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor$RunnableAdapter.run()
2019-04-13 15:16:58.965 INFO 12736 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean.createInitialEntityManagerFactory()
2019-04-13 15:16:58.965 INFO 12736 --- [ restartedMain] .SchemaDropperImpl$DelayedDropTask.doDrop()
2019-04-13 15:16:58.970 INFO 12736 --- [ restartedMain] o.s.b.f.support.DisposableBean.destroy()
2019-04-13 15:16:58.970 INFO 12736 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource.start()
2019-04-13 15:16:58.975 INFO 12736 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource.start()
```

- 3.
4. Navigate to Maven, Click Existing Maven Projects



- 5.
6. In Root Directory, browse to where you unzipped the restful service and import the folder. It will download all the plugins and dependencies needed to run the restful-web services
7. Click Run when it's done downloading the plugins and dependencies