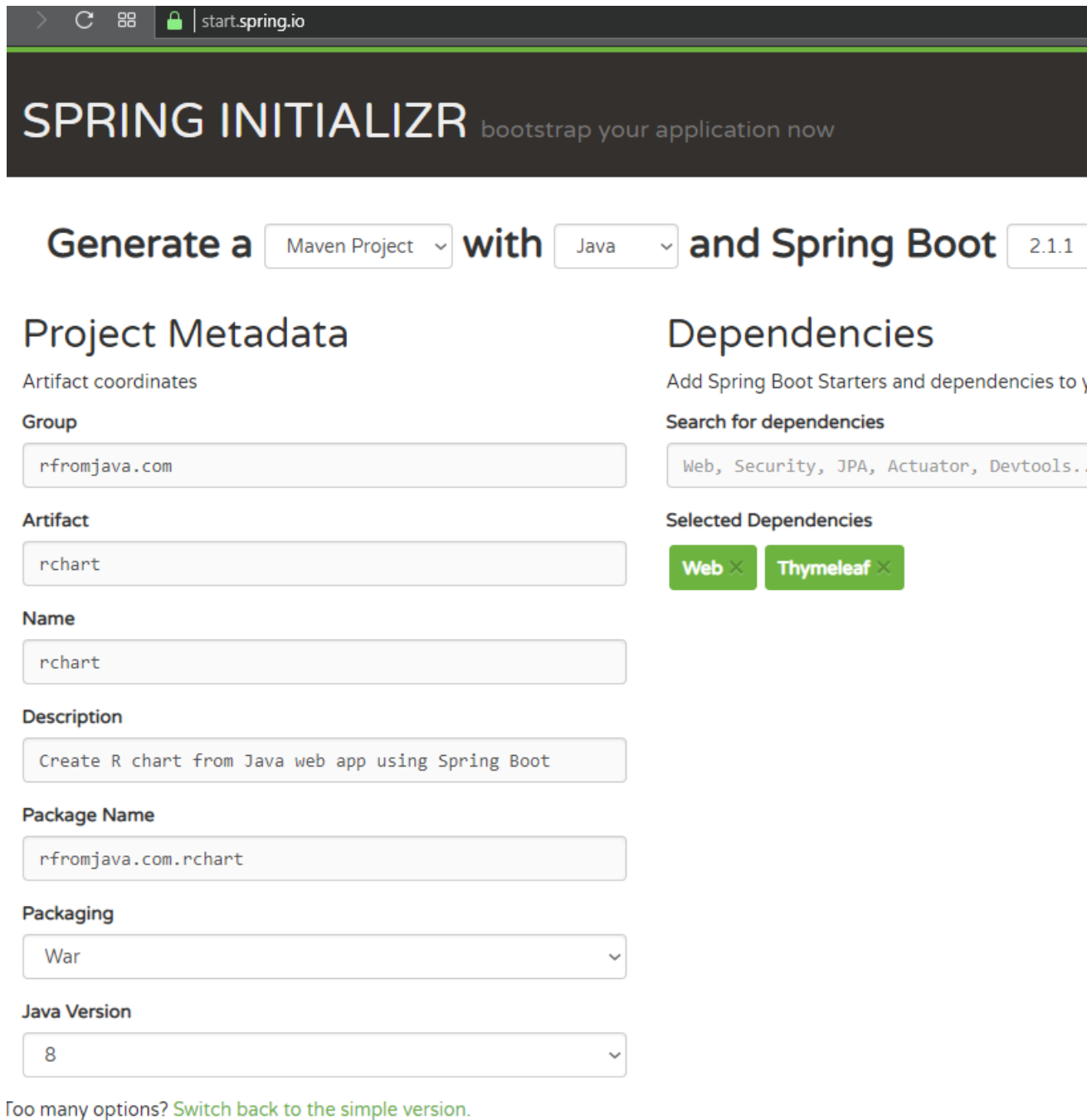


# NOTES AND STEPS ON GENERATING AND DEVELOPING rchart PROJECT

1.- Go to start.spring.io and generate your project, setting it similar to the below screen:



The screenshot shows the Spring Initializr web application interface. At the top, it says "SPRING INITIALIZR bootstrap your application now". Below this, there's a section to "Generate a" with a dropdown menu set to "Maven Project", followed by "with" and a dropdown menu set to "Java", and "and Spring Boot" with a dropdown menu set to "2.1.1".

On the left, under "Project Metadata", there are several input fields:

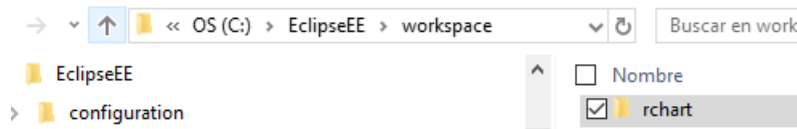
- Artifact coordinates:**
  - Group:** rfromjava.com
  - Artifact:** rchart
  - Name:** rchart
  - Description:** Create R chart from Java web app using Spring Boot
  - Package Name:** rfromjava.com.rchart
- Packaging:** War
- Java Version:** 8

On the right, under "Dependencies", there's a search bar with the text "Web, Security, JPA, Actuator, Devtools..". Below it, under "Selected Dependencies", there are two green buttons: "Web" and "Thymeleaf".

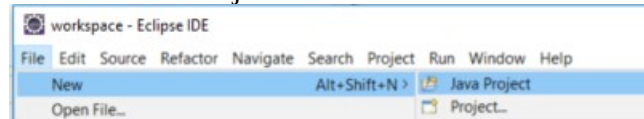
At the bottom left, there's a link: "Too many options? [Switch back to the simple version.](#)"

Generate Project alt + ⌘

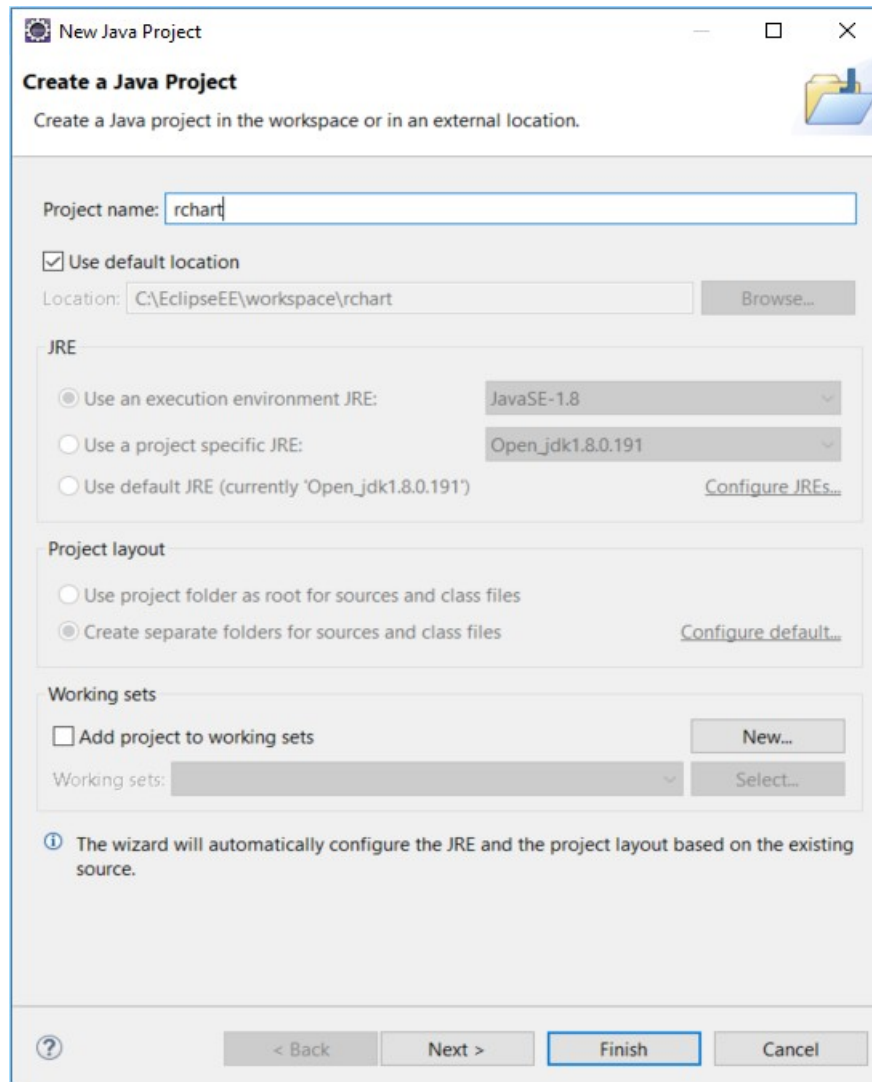
2.- Once the project is generated, open the zip file and extract it to your eclipse workspace:



3.- Start eclipse and go to File → New → Java Project:



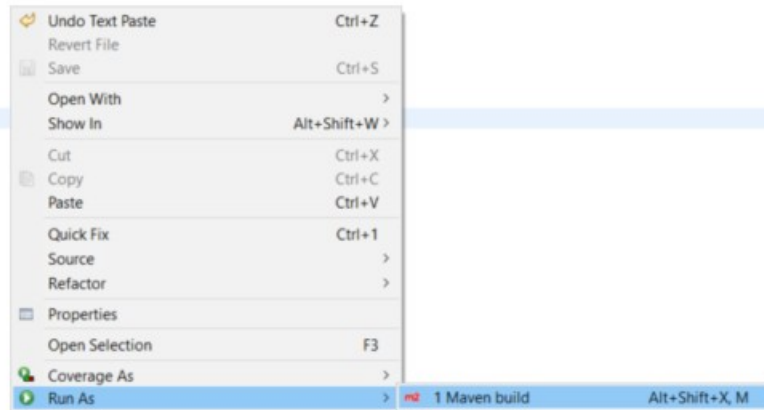
4.- Enter the name of your java project as it was generated by the spring starter website, click on finish:



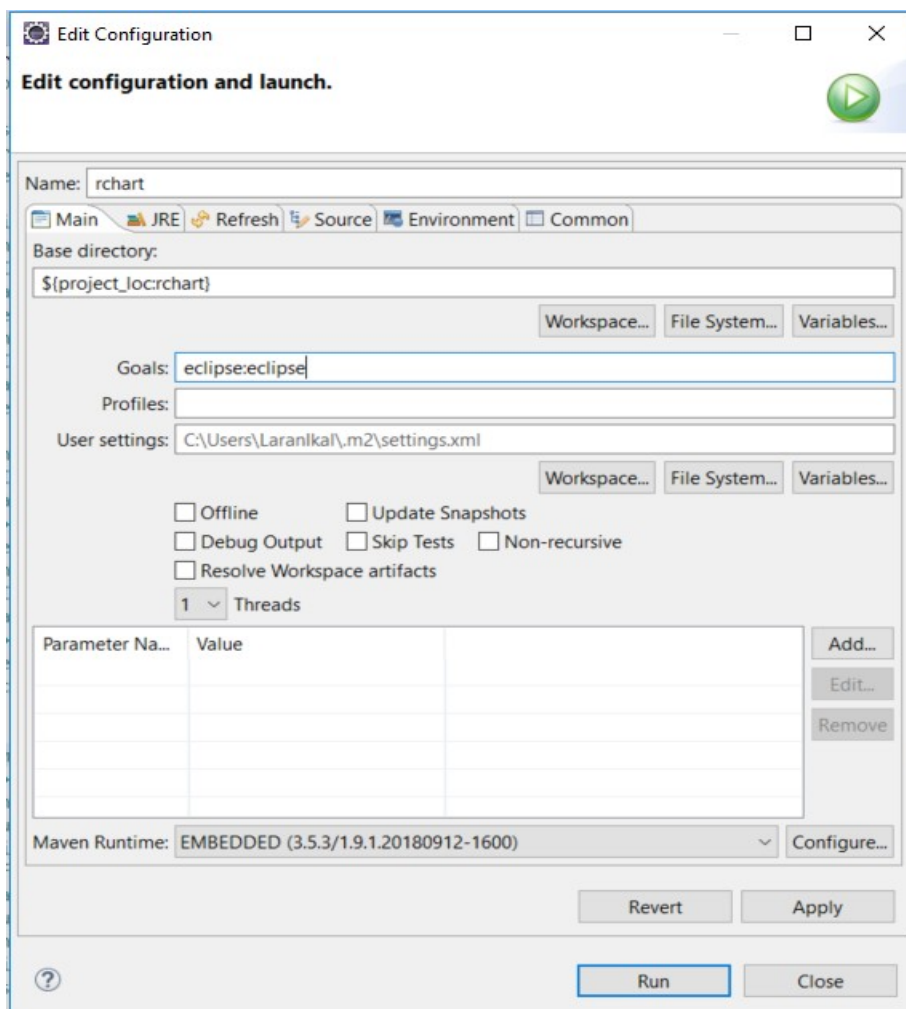
5.- Edit file pom.xml and add the next lines in the build section:

```
44<build>
45  <finalName>RChartFromJava</finalName>
46  <plugins>
47    <plugin>
48      <groupId>org.springframework.boot</groupId>
49      <artifactId>spring-boot-maven-plugin</artifactId>
50      <configuration>
51        <warName>${finalName}</warName>
52        <outputDirectory>C:\apache-tomcat-8.5.30\webapps</outputDirectory>
53      </configuration>
54    </plugin>
55  </plugins>
56</build>
```

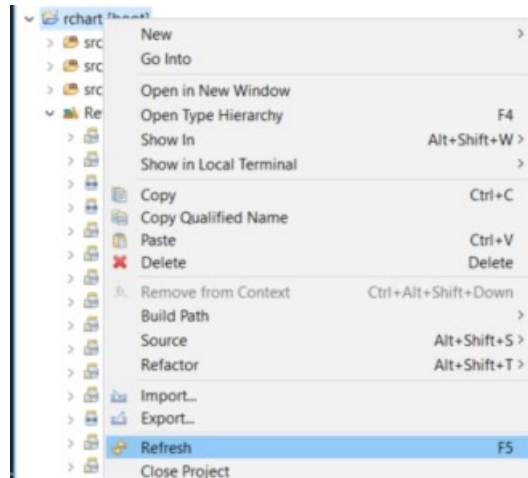
6.- Right-Click with the mouse and select Run-As → Maven build:



7.- On the edit configuration and launch screen type: eclipse:eclipse and click Run:



8.- Right-click with the mouse on the project name and select refresh:



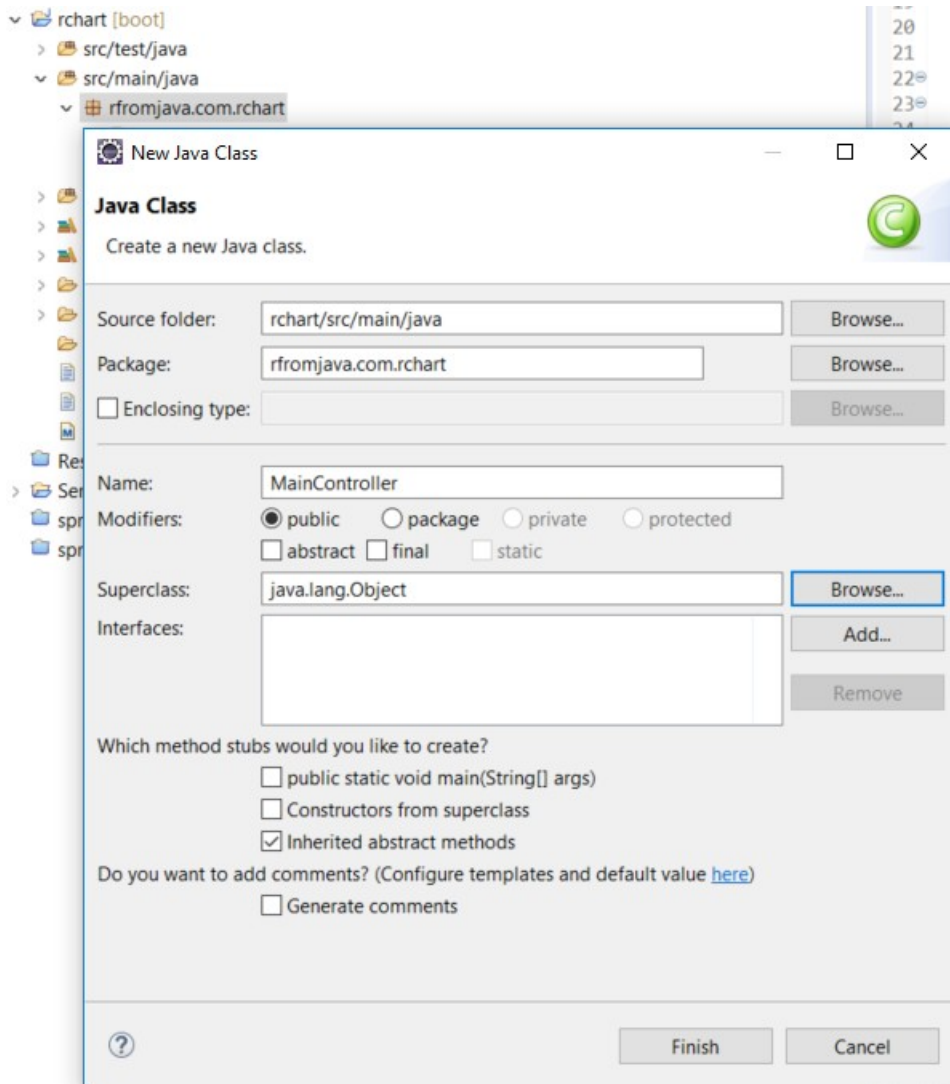
Now you will be able to see all the Referenced libraries updated, here just some of them:

```

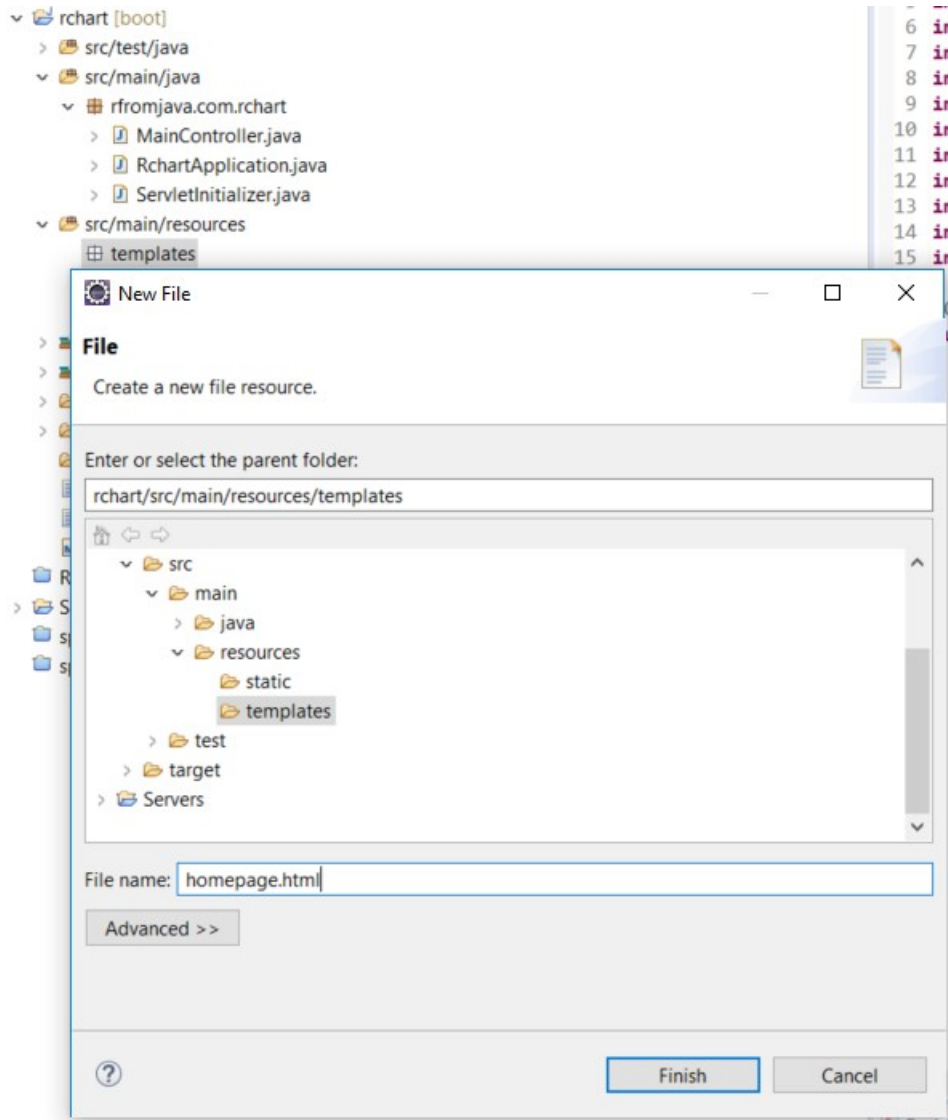
rchart [boot]
├── src/test/java
├── src/main/java
├── src/main/resources
└── Referenced Libraries
    ├── javax.annotation-api-1.3.2.jar - M2_REPO\javax\annotation\jav
    ├── validation-api-2.0.1.Final.jar - M2_REPO\javax\validation\validi
    ├── spring-boot-starter-thymeleaf-2.1.1.RELEASE.jar - M2_REPO\or
    ├── spring-boot-starter-2.1.1.RELEASE.jar - M2_REPO\org\springfra
    ├── spring-boot-2.1.1.RELEASE.jar - M2_REPO\org\springframewor
    ├── spring-core-5.1.3.RELEASE.jar - M2_REPO\org\springframework
    └── spring-jcl-5.1.3.RELEASE.jar - M2_REPO\org\springframework\

```

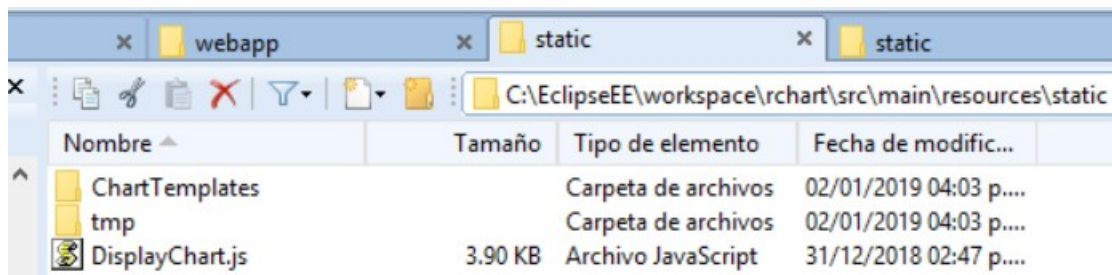
9.- Create the first controller class, just right-click on the package name and select New → class, enter the name, in this case it is called MainController, click finish:



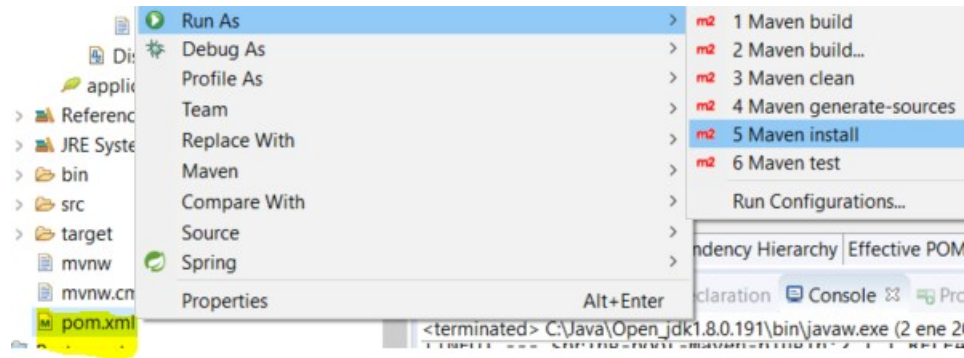
10.- Right-Click on templates to create the Thymeleaf template, select New → File, enter the name, in this case homepage.html, click finish:



11.- Under your **project folder/src/main/resources/static**, put all the public files needed by your web app:



12.- To generate your war file in the folder configured in the pom file, right-click on the pom.xml file, select **Run As → Maven Install:**

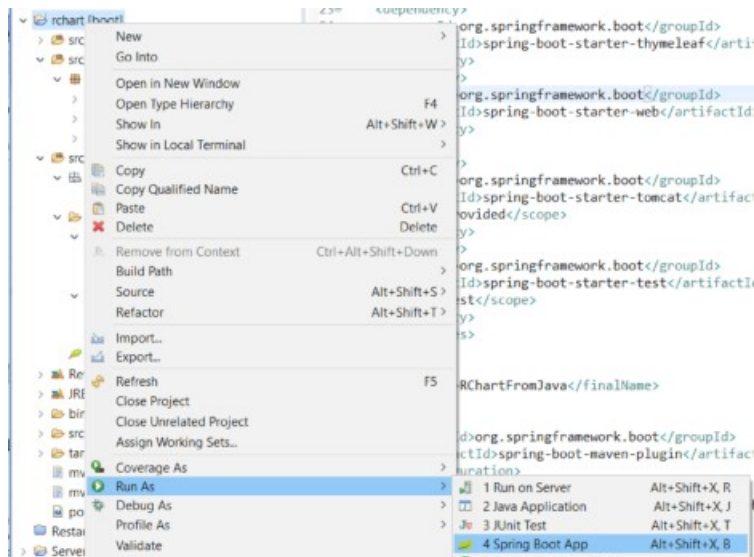


In this case, this create a war file on folder: `C:\apache-tomcat-8.5.30\webapps`

Now just by running startup.bat starts tomcat and expands the war file and the web app is ready to be accessed opening the web browser and going to:

`localhost:8080/RChartFromJava/`

Note. To start the web app inside eclipse, we just need to right-click on the project name, select **Run As → Spring Boot App:**



to view the web app, just point the web browser to: `localhost:8080`