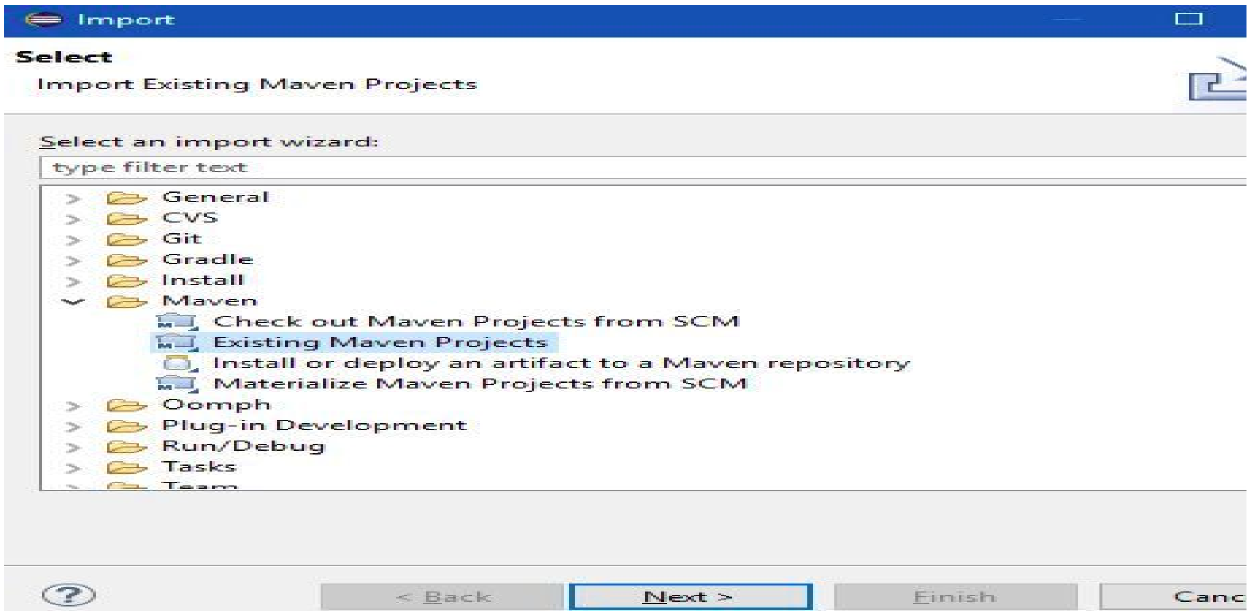
Build & Deploy WebApp using Maven & Eclipse Oxygen

Pre-requisites:

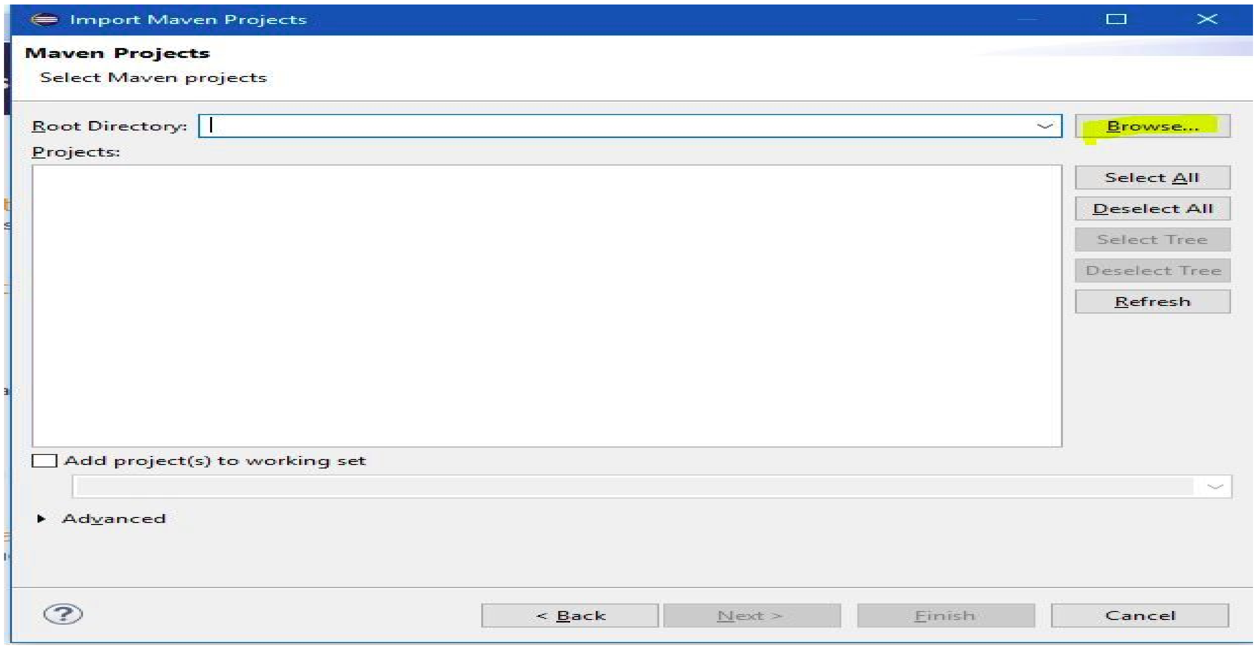
1.) User should have Eclipse Version: 2018-09 (4.9.0) installed.
2.) Download Maven App zip from the below link.
“ https://github.com/rajnikhattarrsinha/java-tomcat-maven-example”
3.) Add “*%JAVA_HOME%\bin*” to the PATH under system variables (Same as done for “M2” for Maven installation)
4.) Java JDK installed and java path set in the environment variables (refer the Maven installation document for how to instructions), confirm java is working properly by running CMD as administrator and executing ***> java -version***

Download the **Eclipse Oxygen** release from the below link https://www.eclipse.org/downloads/packages/release/oxygen/3a”

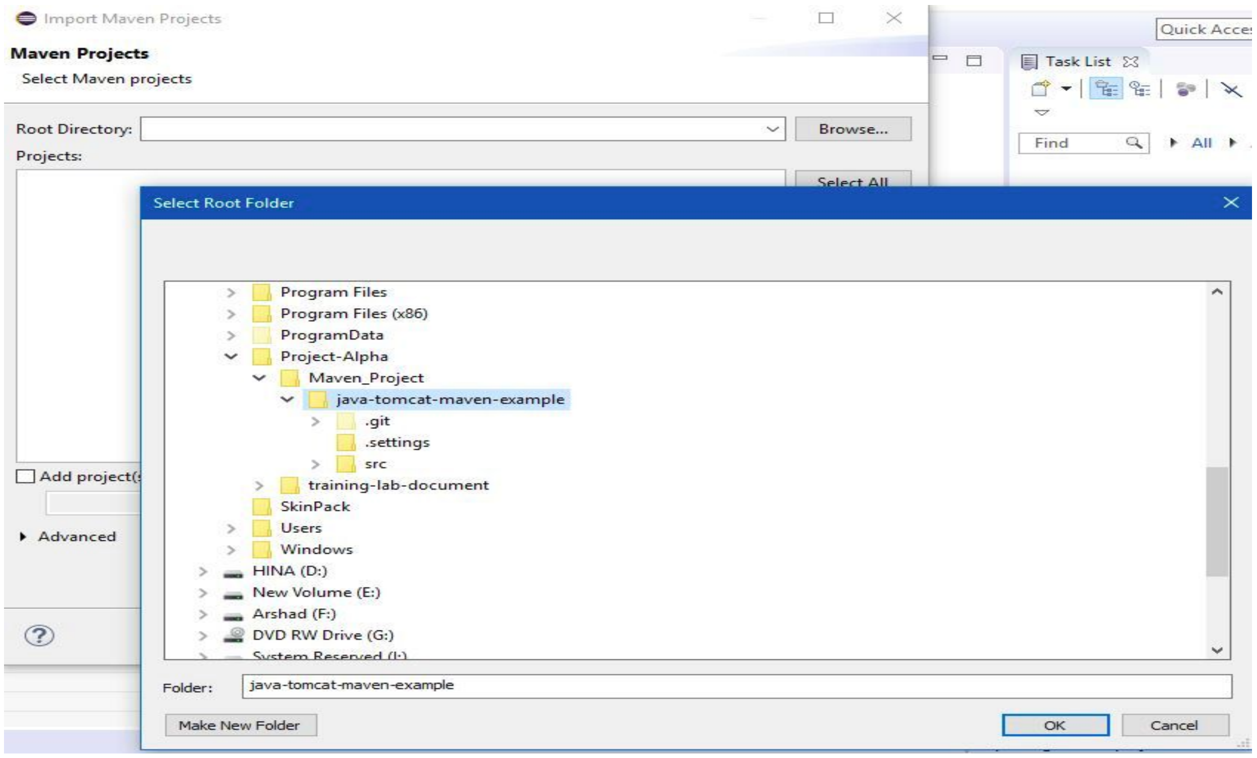
- Unzip the downloaded file in C drive (Create a New folder).
- Launch the Eclipse IDE by running eclipse.exe (as **administrator**) under **C** drive eclipse folder.
- Goto Files > Import.
- Under Import expand the Maven Folder and Click on “Existing Maven Projects”

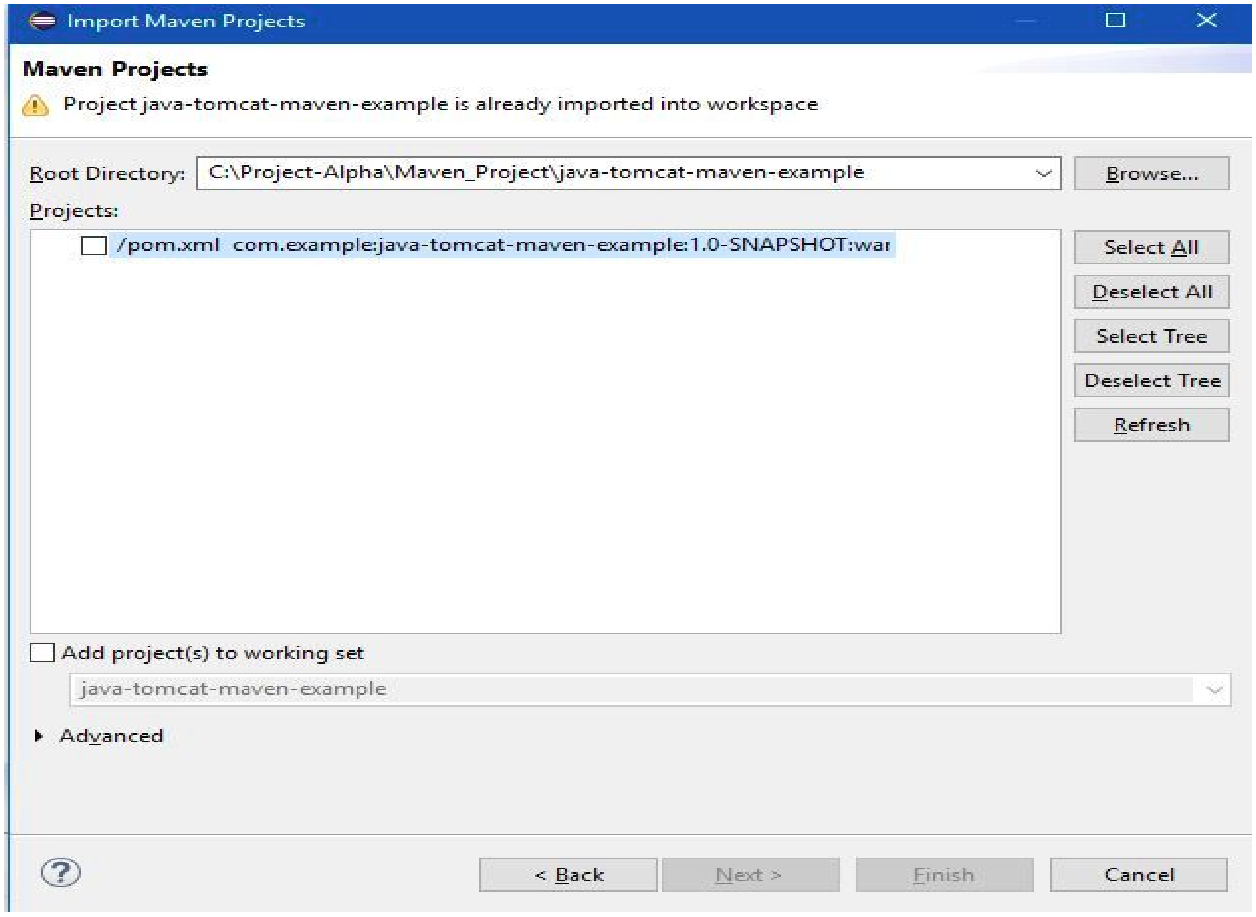


- Click on the Browse tab and browse to the folder where https://github.com/rajnikhattarrsinha/java-tomcat-maven-example.zip was downloaded and extracted.



- Once into the project folder - select the parent project folder and click ok





Project java-tomcat-maven-example is already imported into workspace

Root Directory: C:\Project-Alpha\Maven_Project\java-tomcat-maven-example

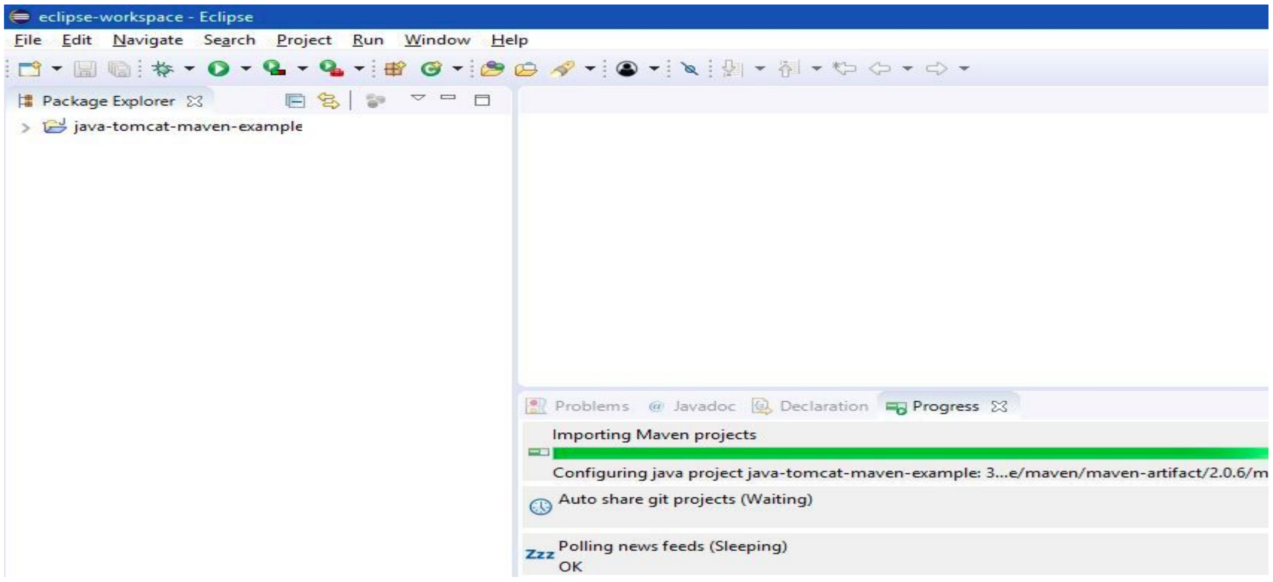
Projects:

- [ ] /pom.xml com.example:java-tomcat-maven-example:1.0-SNAPSHOT:war

- [ ] Add project(s) to working set

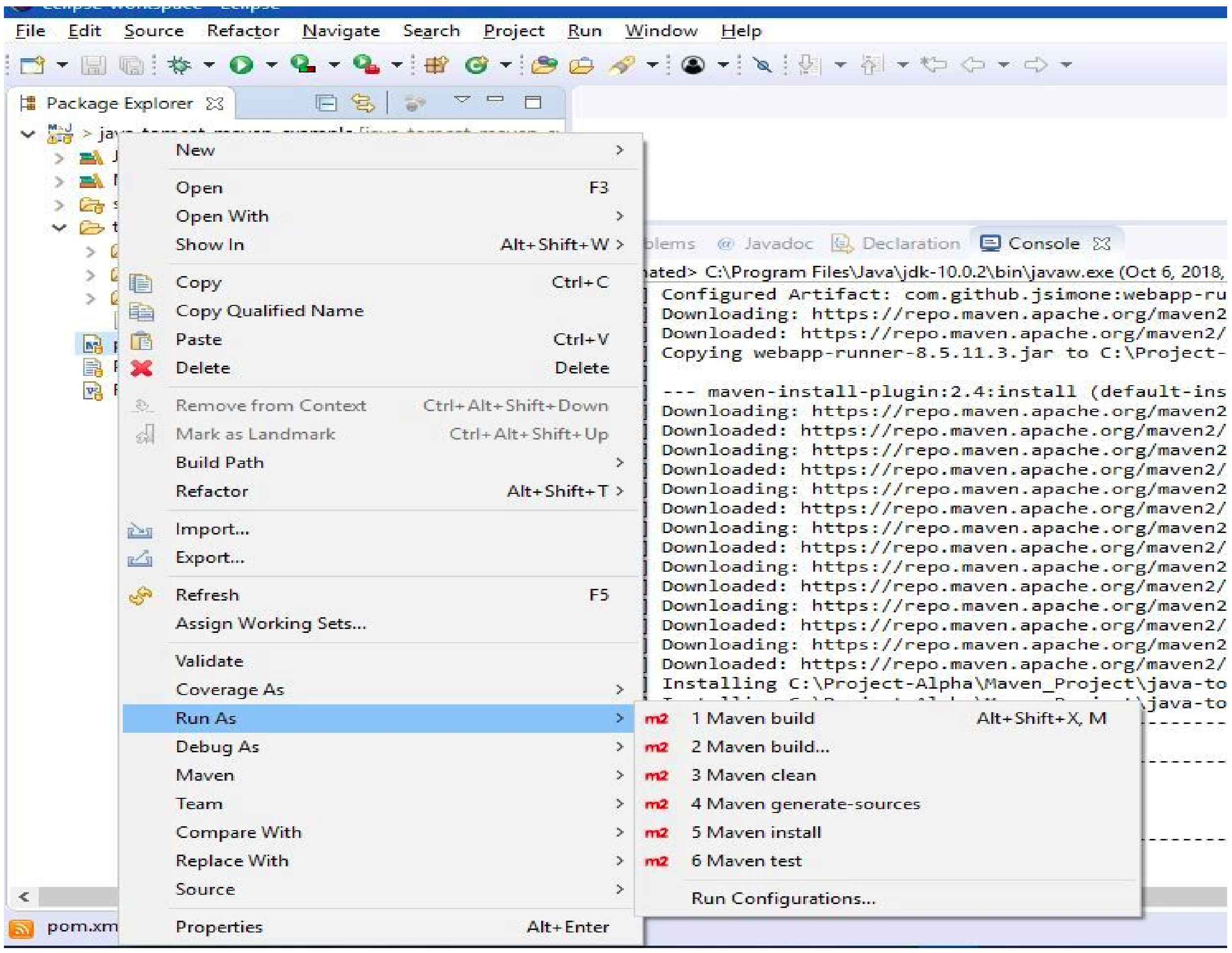
java-tomcat-maven-example

- You can see that the pom.xml file has been automatically scanned and selected. Click Next and the project will start importing. Once all the dependencies have been downloaded you will see the project folder structure on the left hand side of the Eclipse.


eclipse-workspace - Eclipse
File Edit Navigate Search Project Run Window Help
Package Explorer
java-tomcat-maven-example
Problems
Javadoc
Declaration
Progress

```
Importing Maven projects
Configuring java project java-tomcat-maven-example: 3...e/maven/maven-artifact/2.0.6/m
Auto share git projects (Waiting)
Polling news feeds (Sleeping)
OK
```

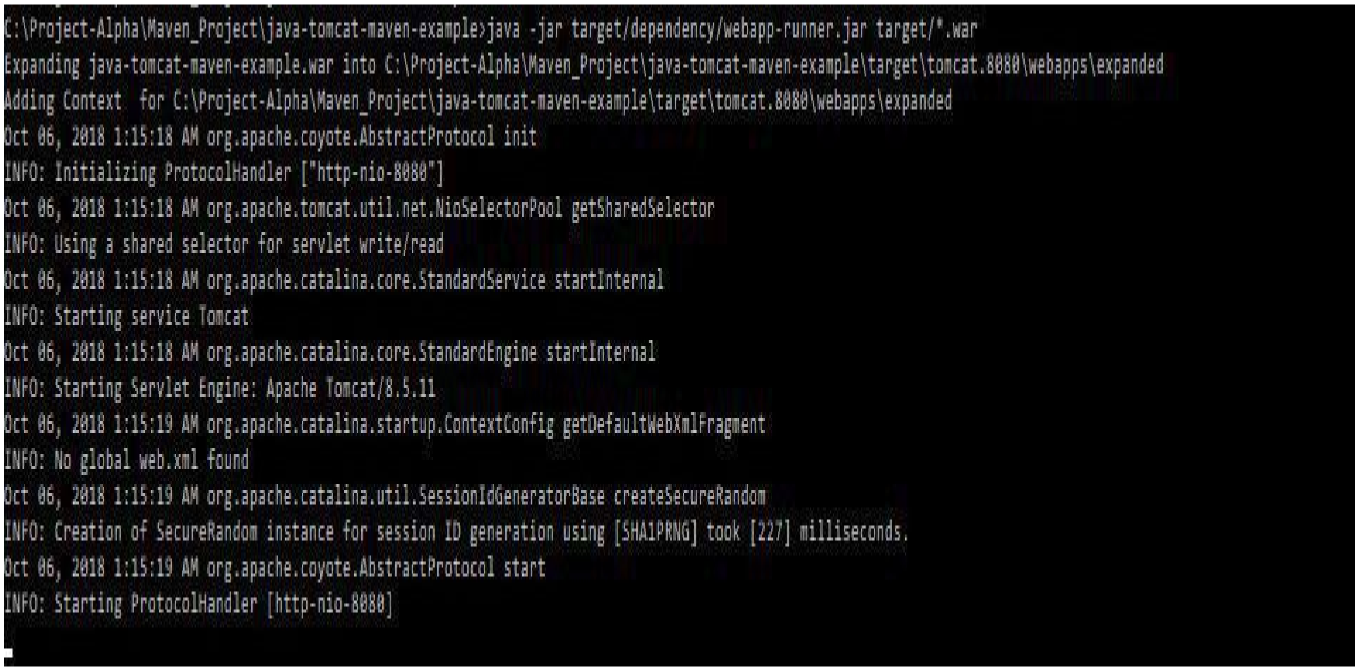
- Expand the project folder, right click on the pom.xml and click on Run As ***3. Maven Clean***.



```
ated> C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (Oct 6, 2018,
] Configured Artifact: com.github.jsimone:webapp-ru
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Copying webapp-runner-8.5.11.3.jar to C:\Project-
]
] --- maven-install-plugin:2.4:install (default-ins
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Downloading: https://repo.maven.apache.org/maven2
] Downloaded: https://repo.maven.apache.org/maven2/
] Installing C:\Project-Alpha\Maven_Project\java-to
```

- You can see on the right hand side in the console window that Eclipse is downloading all the defined dependencies as described in the pom.xml file.
- Once you see ***Build Success*** in the console window, right click on the **pom.xml** file and this time select Run As ***5. Maven Install***.
- After the build is successful you can see that it created a "**target"** sub-folder under the java-tomcat-maven-example folder.
- Now click on the windows button and run CMD as "**Administrator**".
- Cd to the project folder where the **java-tomcat-maven-example folder.zip** was extracted and run the below command

"***java -jar target/dependency/webapp-runner.jar target/*.war***"



```
C:\Project-Alpha\Maven_Project\java-tomcat-maven-example>java -jar target/dependency/webapp-runner.jar target/*.war
Expanding java-tomcat-maven-example.war into C:\Project-Alpha\Maven_Project\java-tomcat-maven-example\target\tomcat.8080\webapps\expanded
Adding Context  for C:\Project-Alpha\Maven_Project\java-tomcat-maven-example\target\tomcat.8080\webapps\expanded
Oct 06, 2018 1:15:18 AM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-nio-8080"]
Oct 06, 2018 1:15:18 AM org.apache.tomcat.util.net.NioSelectorPool getSharedSelector
INFO: Using a shared selector for servlet write/read
Oct 06, 2018 1:15:18 AM org.apache.catalina.core.StandardService startInternal
INFO: Starting service Tomcat
Oct 06, 2018 1:15:18 AM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/8.5.11
Oct 06, 2018 1:15:19 AM org.apache.catalina.startup.ContextConfig getDefaultWebXmlFragment
INFO: No global web.xml found
Oct 06, 2018 1:15:19 AM org.apache.catalina.util.SessionIdGeneratorBase createSecureRandom
INFO: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [227] milliseconds.
Oct 06, 2018 1:15:19 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler [http-nio-8080]
```

- Console output ***- starting ProtocolHandloer [http-nio-8080]*** confirms that the application has been deployed and is running on port **8080** locally. To verify open the browser and browse to http://localhost:8080