Jersey Web API Development (REST)

Table of Contents

1 Create New Project 1

2 Add Jersey Dependency 2

3 Add a new Java Class in Source Package 2

4 Create WEB-INF Folder and edit auto generated web.xml file 2

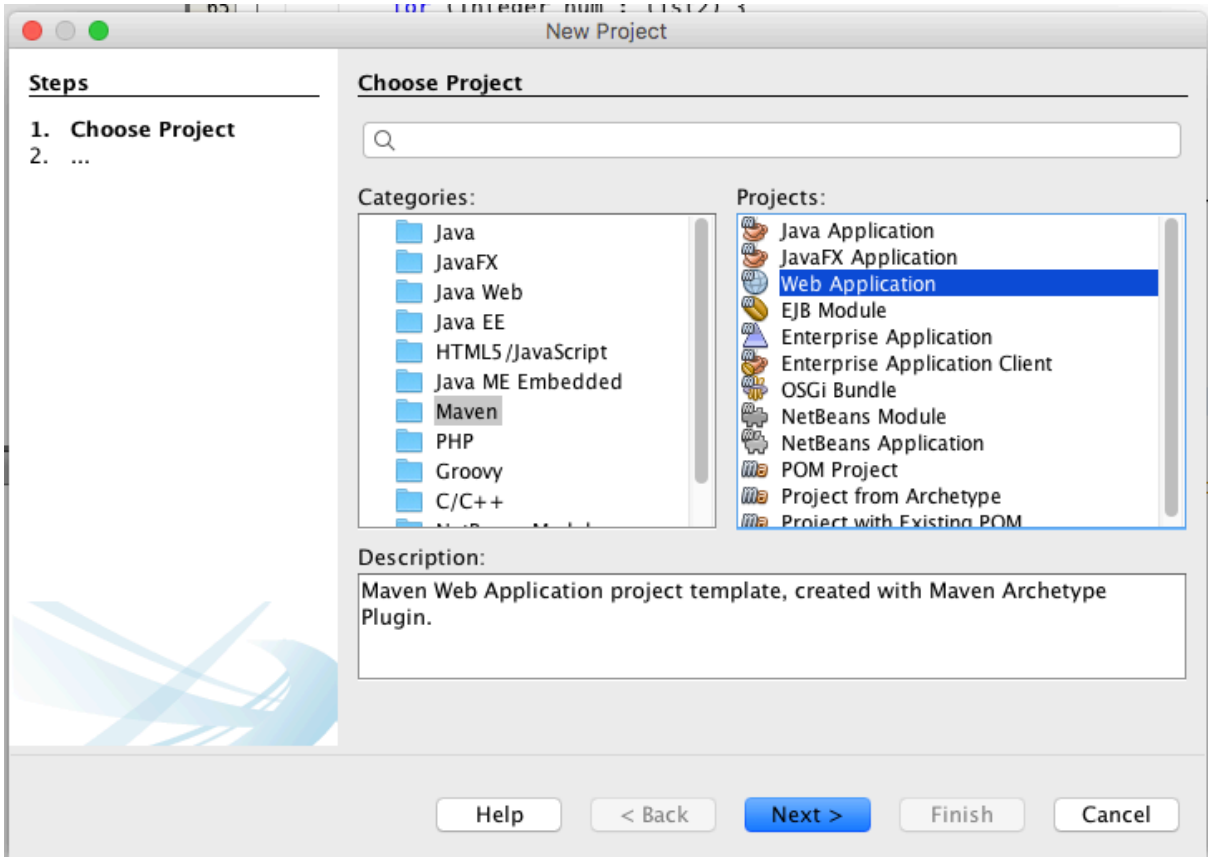
5 Servlet Specification (web.xml) file 5

6 Adding Plugin Servlet Specification pom.xml 6

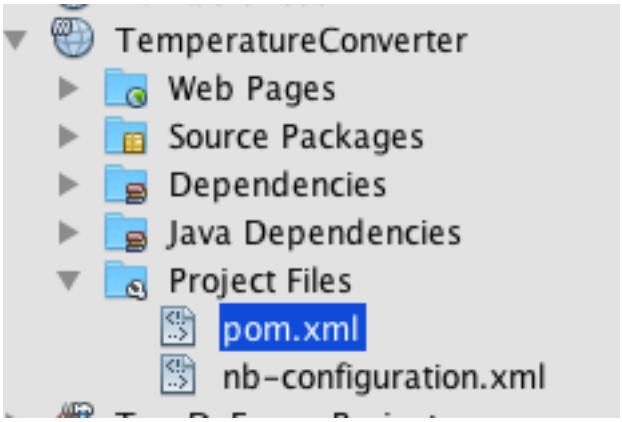
7 Run Project 7

8 Add http method (GET,POST, DELETE OR MORE) to your Java file created in step 3. .. 9

1 Create New Project



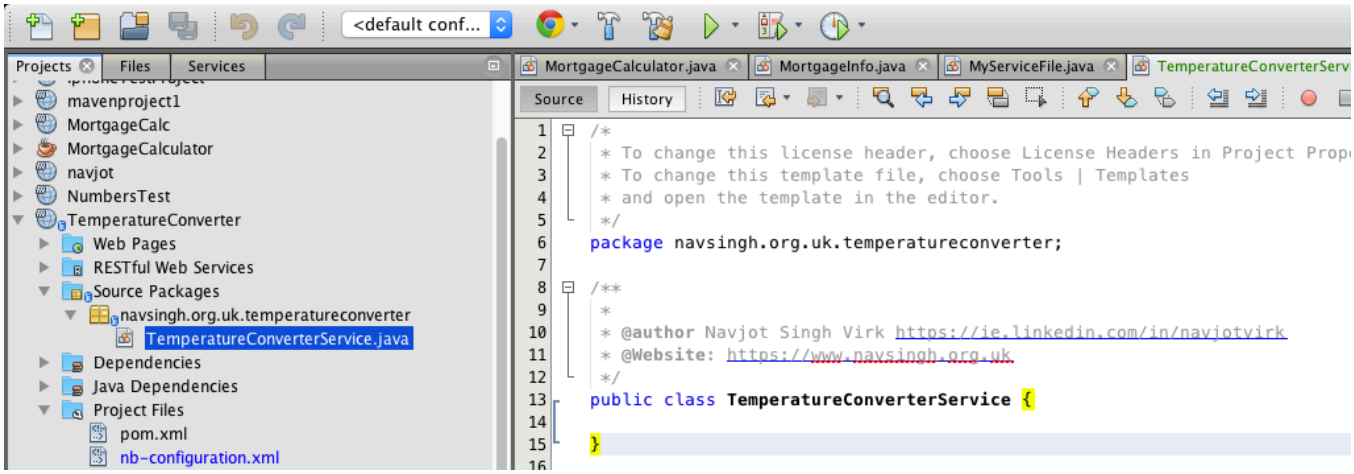
2 Add Jersey Dependency



Code:

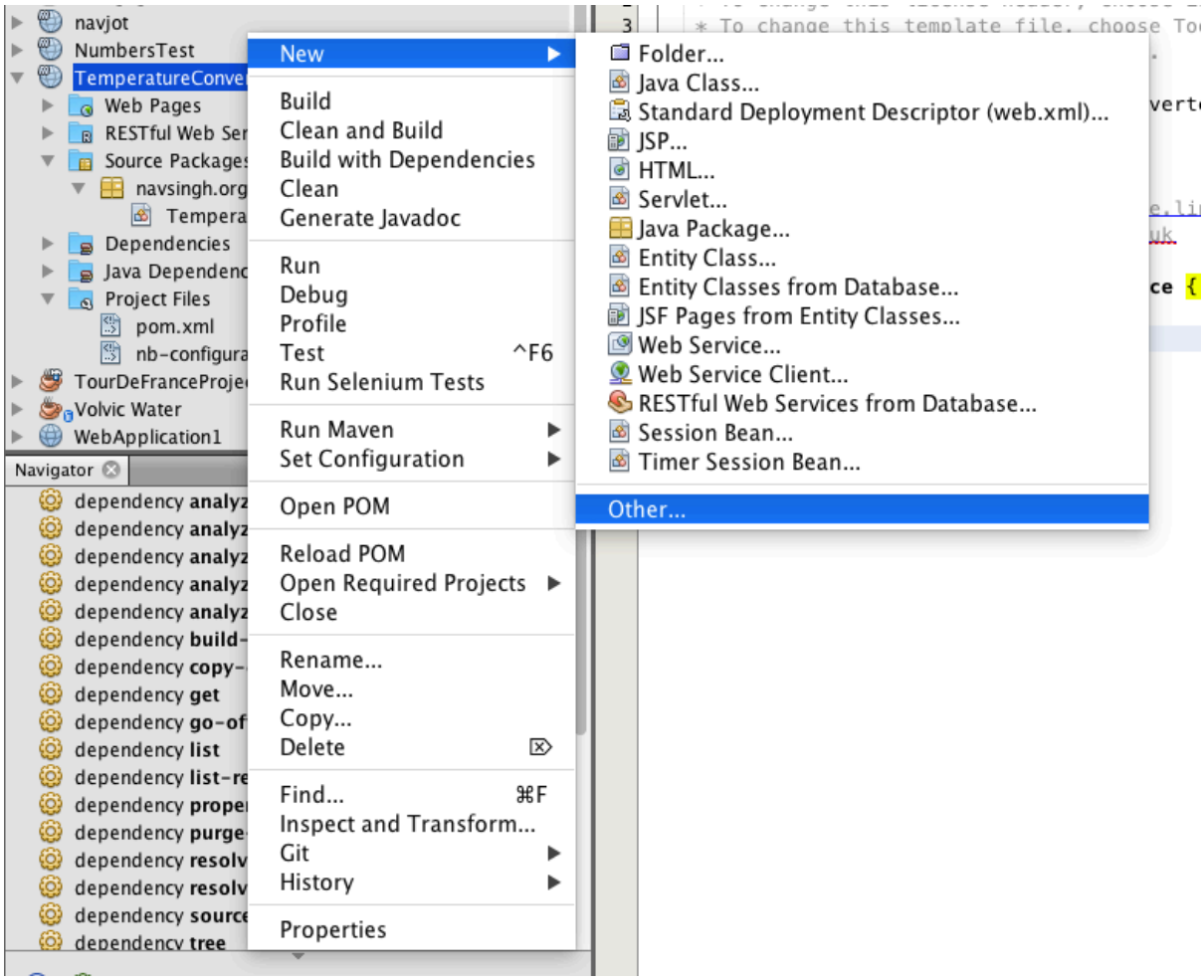
```
<dependency>
        <groupId>org.glassfish.jersey.containers</groupId>
        <artifactId>jersey-container-servlet</artifactId>
        <version>2.0</version>
</dependency>
```

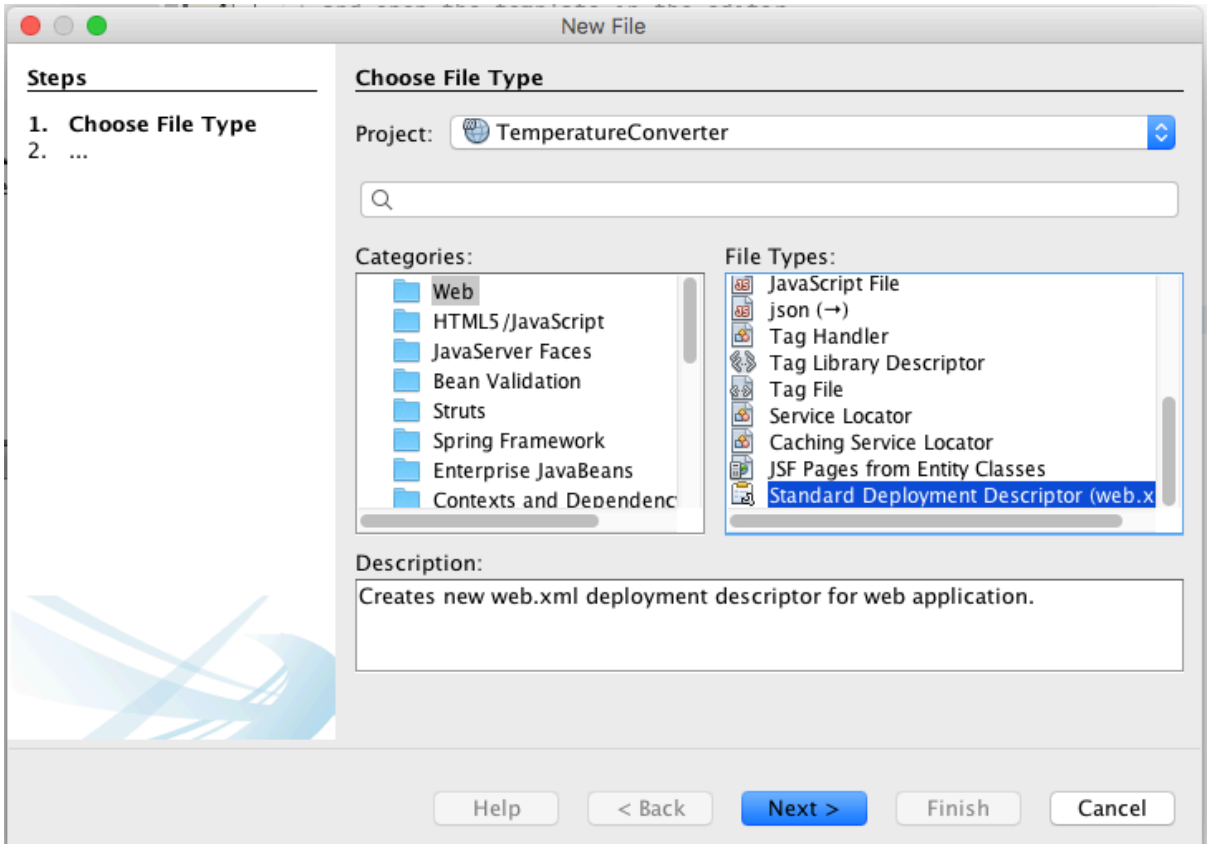
3 Add a new Java Class in Source Package

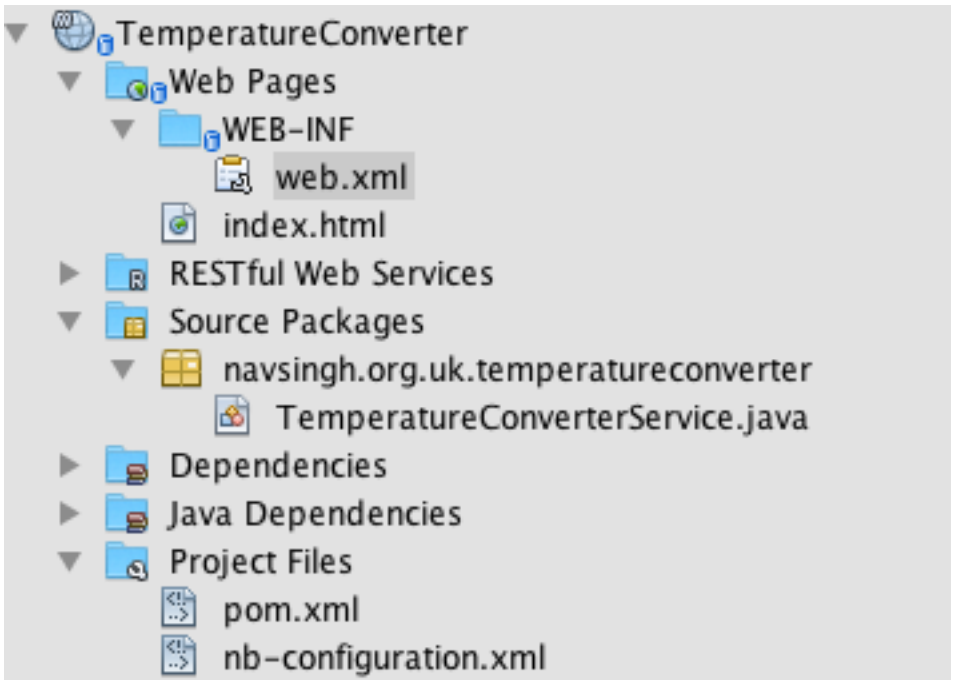


4 Create WEB-INF Folder and edit auto generated web.xml file

Select Project > New > Other


navjot
NumbersTest
TemperatureConve
Web Pages
RESTful Web Ser
Source Packages
navsingh.org
Tempera
Dependencies
Java Dependenc
Project Files
pom.xml
nb-configura
TourDeFranceProje
Volvic Water
WebApplication1
Navigator
dependency analyz
dependency analyz
dependency analyz
dependency analyz
dependency analyz
dependency build-
dependency copy-
dependency get
dependency go-of
dependency list
dependency list-re
dependency prope
dependency purge
dependency resolv
dependency resolv
dependency source
dependency tree
New
Build
Clean and Build
Build with Dependencies
Clean
Generate Javadoc
Run
Debug
Profile
Test ^F6
Run Selenium Tests
Run Maven
Set Configuration
Open POM
Reload POM
Open Required Projects
Close
Rename...
Move...
Copy...
Delete
Find... ⌘F
Inspect and Transform...
Git
History
Properties
Folder...
Java Class...
Standard Deployment Descriptor (web.xml)...
JSP...
HTML...
Servlet...
Java Package...
Entity Class...
Entity Classes from Database...
JSF Pages from Entity Classes...
Web Service...
Web Service Client...
RESTful Web Services from Database...
Session Bean...
Timer Session Bean...
Other...


New File
Steps
1. Choose File Type
2. ...
Choose File Type
Project:
TemperatureConverter
Categories:
Web
HTML5/JavaScript
JavaServer Faces
Bean Validation
Struts
Spring Framework
Enterprise JavaBeans
Contexts and Dependenc
File Types:
JavaScript File
json (→)
Tag Handler
Tag Library Descriptor
Tag File
Service Locator
Caching Service Locator
JSF Pages from Entity Classes
Standard Deployment Descriptor (web.x
Description:
Creates new web.xml deployment descriptor for web application.
Help
< Back
Next >
Finish
Cancel


TemperatureConverter
Web Pages
WEB-INF
web.xml
index.html
RESTful Web Services
Source Packages
navsingh.org.uk.temperatureconverter
TemperatureConverterService.java
Dependencies
Java Dependencies
Project Files
pom.xml
nb-configuration.xml

5 Servlet Specification (web.xml) file

```
<!-- Servelet Specification -->
<servlet>
    <servlet-name>rest</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>navsingh.org.uk.temperatureconverter</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>rest</servlet-name>
    <url-pattern>/api/*</url-pattern>
</servlet-mapping>
<!-- -->
```

Code:

```
<!-- Servelet Specification -->
<servlet>
  <servlet-name>rest</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>navsingh.org.uk.temperatureconverter</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>rest</servlet-name>
  <url-pattern>/api/*</url-pattern>
</servlet-mapping>
<!-- -->
```

Now, your web.xml should look somewhat like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/we
         version="3.1">
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>

    <!-- Servelet Specification -->
    <servlet>
        <servlet-name>rest</servlet-name>
        <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
        <init-param>
            <param-name>jersey.config.server.provider.packages</param-name>
            <param-value>navsingh.org.uk.temperatureconverter</param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>rest</servlet-name>
        <url-pattern>/api/*</url-pattern>
    </servlet-mapping>
    <!-- -->

</web-app>
```

6 Adding Plugin Servlet Specification pom.xml

```xml
80
81              <!-- Adding Plugin Servlet Specification pom.xml -->
82              <plugin>
83                  <groupId>org.eclipse.jetty</groupId>
84                  <artifactId>jetty-maven-plugin</artifactId>
85                  <version>9.2.4.v20141103</version>
86                  <configuration>
87                      <httpConnector>
88                          <!-- make jetty listen on 127.0.0.1 for better security -->
89                          <!--host>127.0.0.1</host-->
90                      </httpConnector>
91                      <scanIntervalSeconds>3</scanIntervalSeconds>
92                      <webAppConfig>
93                          <!-- defaultsDescriptor>src/test/resources/jetty-maven-plugin-webdefault.xml
94                      </webAppConfig>
95                  </configuration>
96              </plugin>
97              <!-- -->
```

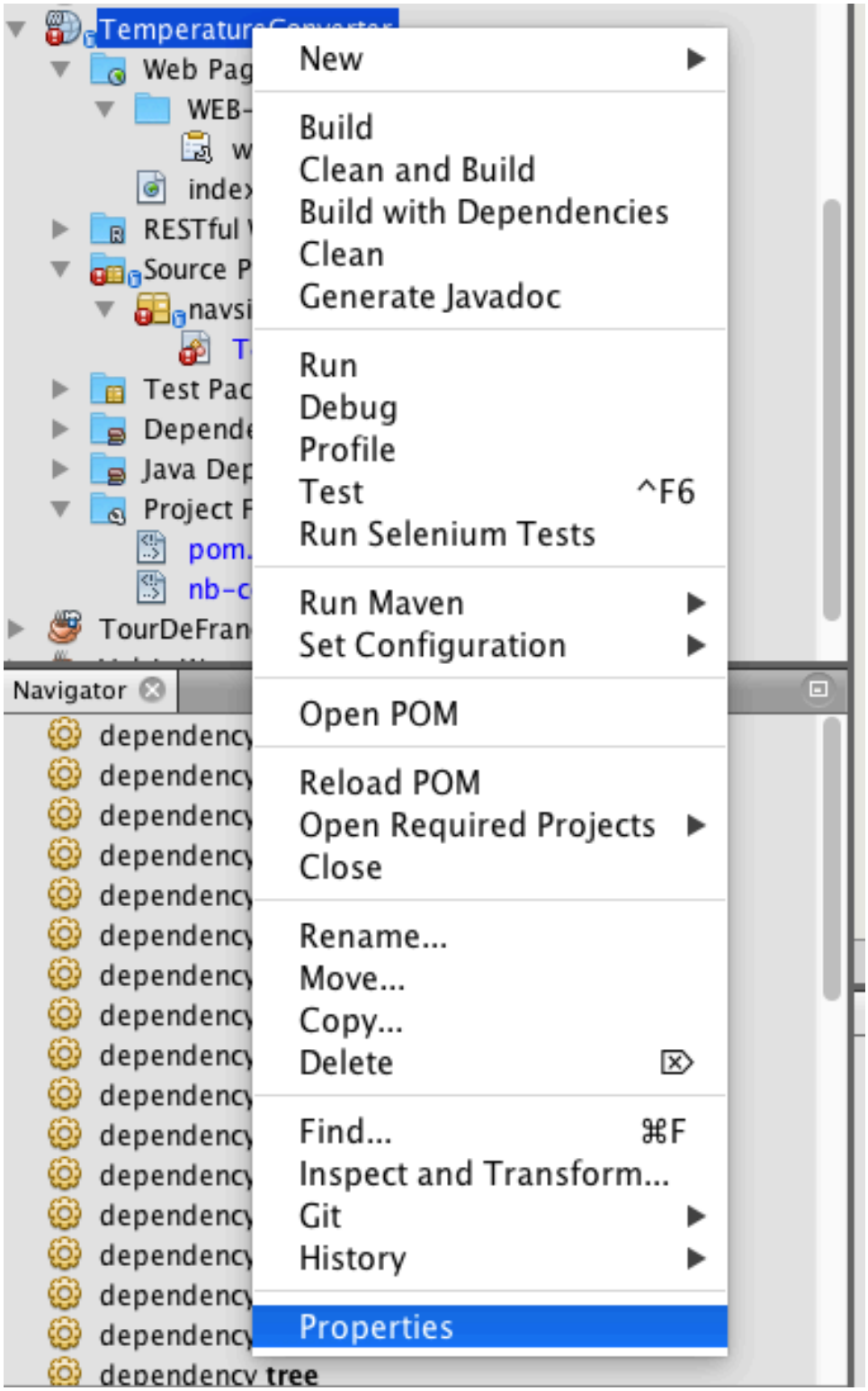
Code:

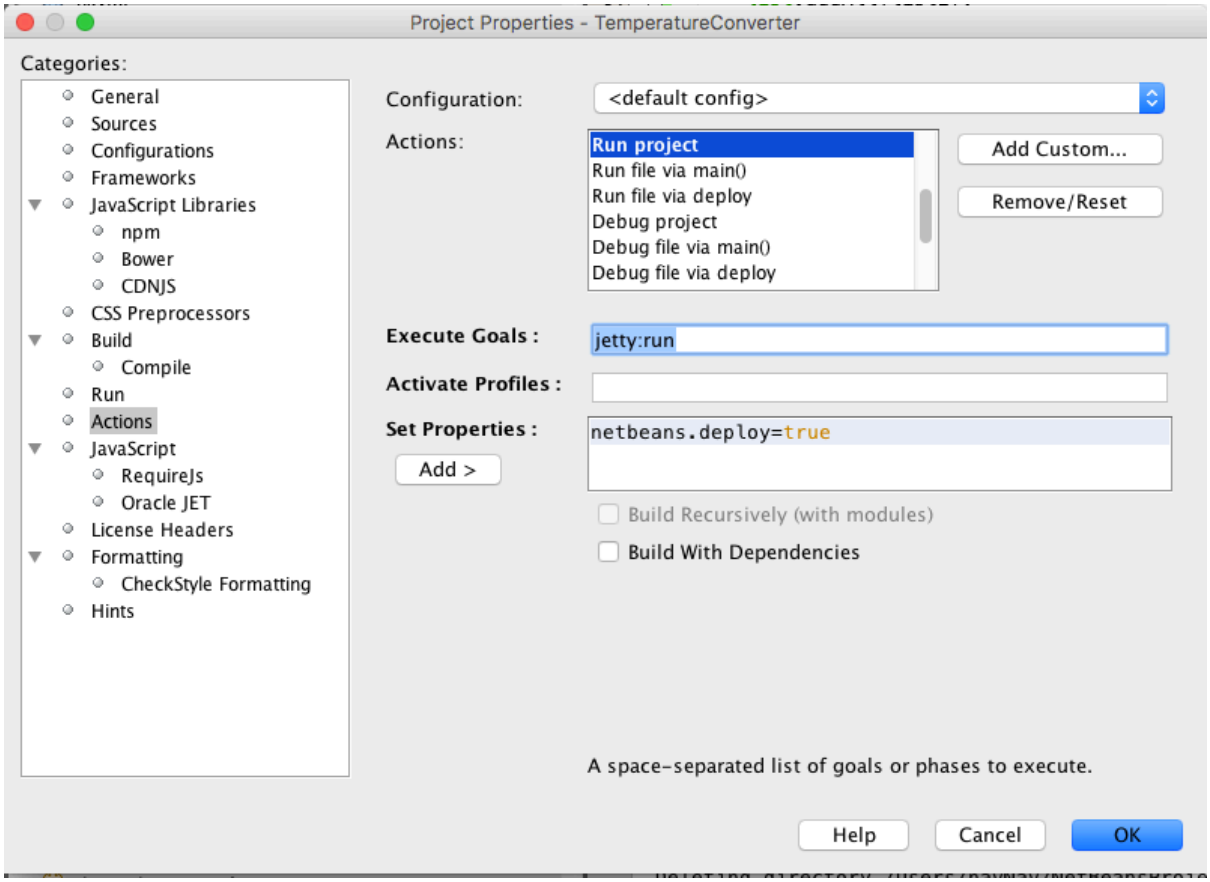
```
<!-- Adding Plugin Servlet Specification pom.xml -->
<plugin>
  <groupId>org.eclipse.jetty</groupId>
  <artifactId>jetty-maven-plugin</artifactId>
  <version>9.2.4.v20141103</version>
  <configuration>
    <httpConnector>
```

```
            <!-- make jetty listen on 127.0.0.1 for better security -->
            <!--host>127.0.0.1</host-->
          </httpConnector>
          <scanIntervalSeconds>3</scanIntervalSeconds>
          <webAppConfig>
            <!-- defaultsDescriptor>src/test/resources/jetty-maven-plugin-
webdefault.xml</defaultsDescriptor -->
          </webAppConfig>
        </configuration>
      </plugin>
      <!-- -->
```

7 Run Project

Before Running the project follow the 2 screenshots below and once done run the project.


New
Build
Clean and Build
Build with Dependencies
Clean
Generate Javadoc
Run
Debug
Profile
Test ^F6
Run Selenium Tests
Run Maven
Set Configuration
Open POM
Reload POM
Open Required Projects
Close
Rename...
Move...
Copy...
Delete
Find... ⌘F
Inspect and Transform...
Git
History
Properties
Navigator
dependency tree



8 Add http method (GET,POST, DELETE OR MORE) to your Java file created in step 3.

The End