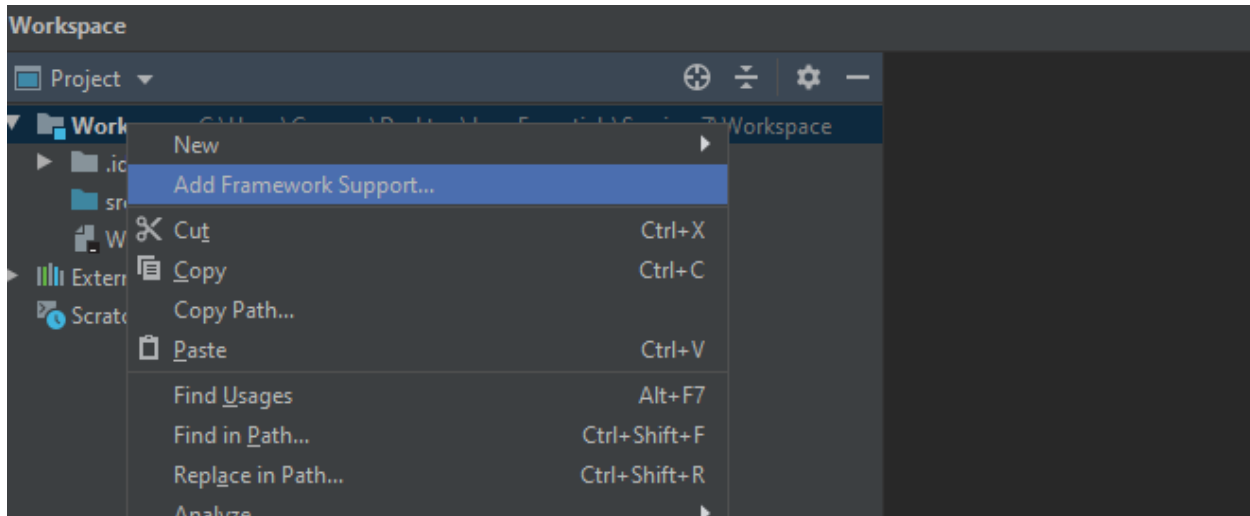


## IntelliJ Workspace Configuration

1. Create a new Java Project (as we have learned) and Right click root project folder > select Add Framework Support > choose Maven click ok



2. edit groupId and artifact id in the pom.xml file:

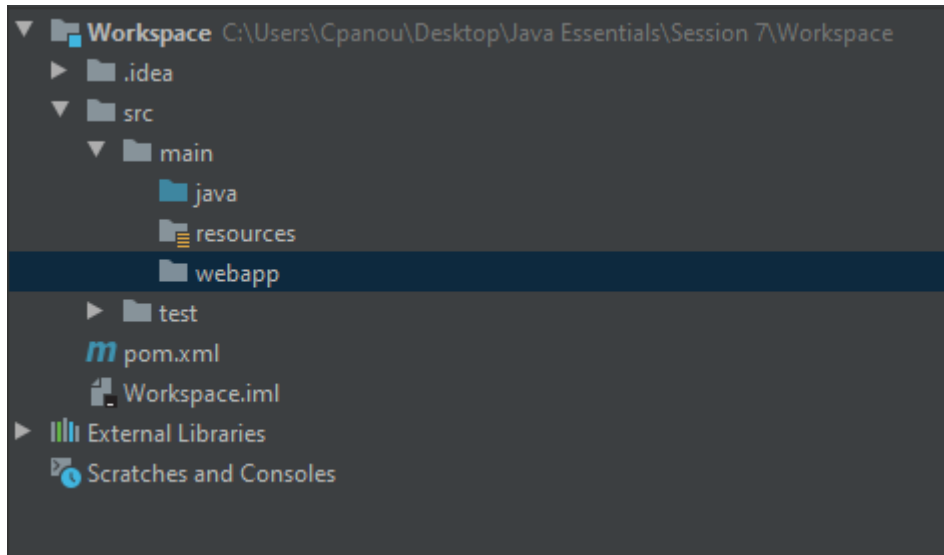
GroupId : com.company

ArtifactId : garage

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.company</groupId>
  <artifactId>garage</artifactId>
  <version>1.0-SNAPSHOT</version>
```

3. create a folder named webapp inside src/main:



4. copy the following into your pom.xml file inside the <project> tag:

```
<packaging>war</packaging>
<dependencies>
  <!-- https://mvnrepository.com/artifact/org.glassfish.jersey.bundles/jaxrs-ri -->
  <dependency>
    <groupId>org.glassfish.jersey.bundles</groupId>
    <artifactId>jaxrs-ri</artifactId>
    <version>2.26-b01</version>
  </dependency>

  <!-- API -->
  <dependency>
    <groupId>jakarta.xml.bind</groupId>
    <artifactId>jakarta.xml.bind-api</artifactId>
    <version>2.3.2</version>
  </dependency>

  <!-- Runtime -->
  <dependency>
    <groupId>org.glassfish.jaxb</groupId>
    <artifactId>jaxb-runtime</artifactId>
    <version>2.3.2</version>
  </dependency>
</dependencies>
```

```

    <dependency>
      <groupId>org.glassfish.jersey.media</groupId>
      <artifactId>jersey-media-json-jackson</artifactId>
      <version>2.22.2</version>
    </dependency>
  </dependencies>
  <build>
    <finalName>garage</finalName>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.6.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>

```

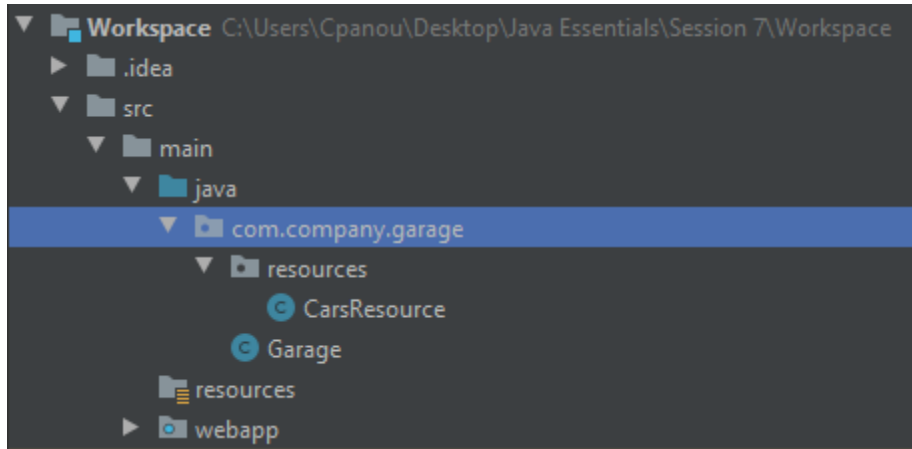
The <packaging> tag defines that the resulting artifact of our project will be a .war file

The <dependencies tag defines a List of external libraries that we want to use in our project. The libraries are downloaded from maven central and included in our artifact.

The in the <build> tag we define the maven plugin to use/

When you make changes to the pom.xml file you will be prompted to import changes, do it or enable auto import.

5. Create the following packages and classes in your project:



The Garage class must extend the Application class from javax.ws.rs.core package (jax-rs package) and be Annotated with the following `@ApplicationPath("/garage")`. The annotation declares that our endpoints will be exposed under the /garage on the host(app server).

The class must look like this:

```
@ApplicationPath("/garage")
public class Garage extends Application {

    @Override
    public Set<Class<?>> getClasses() {
        HashSet set = new HashSet<Class<?>>();
        set.add(JacksonJsonProvider.class);
        set.add(CarsResource.class);
        return set;
    }
}
```

The `getClasses()` method we Override will be called by the JAX-RS framework to get this application's resources.

The CarsResource Class must be annotated with the `@Path("/car")` annotation and contain a method annotated with `@GET`.

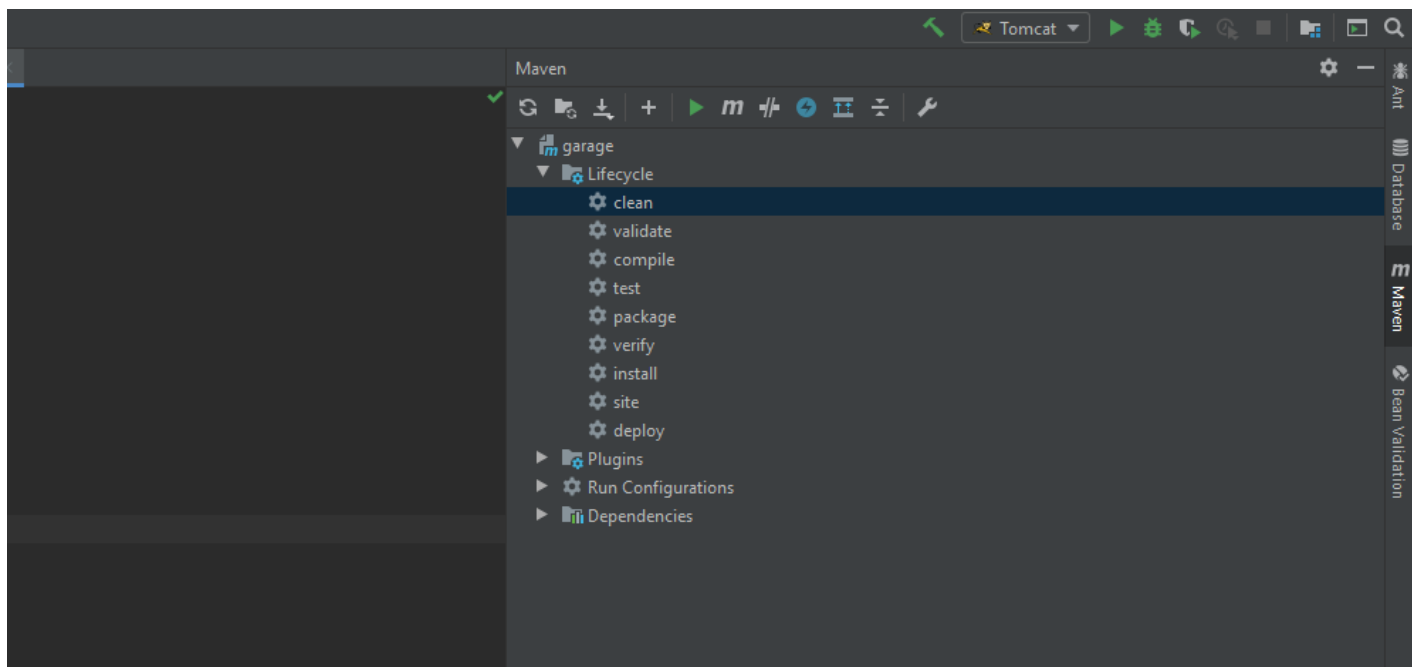
```
@Path("/car")
public class CarsResource {

    @GET
    public String getCars(){
        return "List of Cars";
    }

}
```

6. Lets use Maven to Build the project:

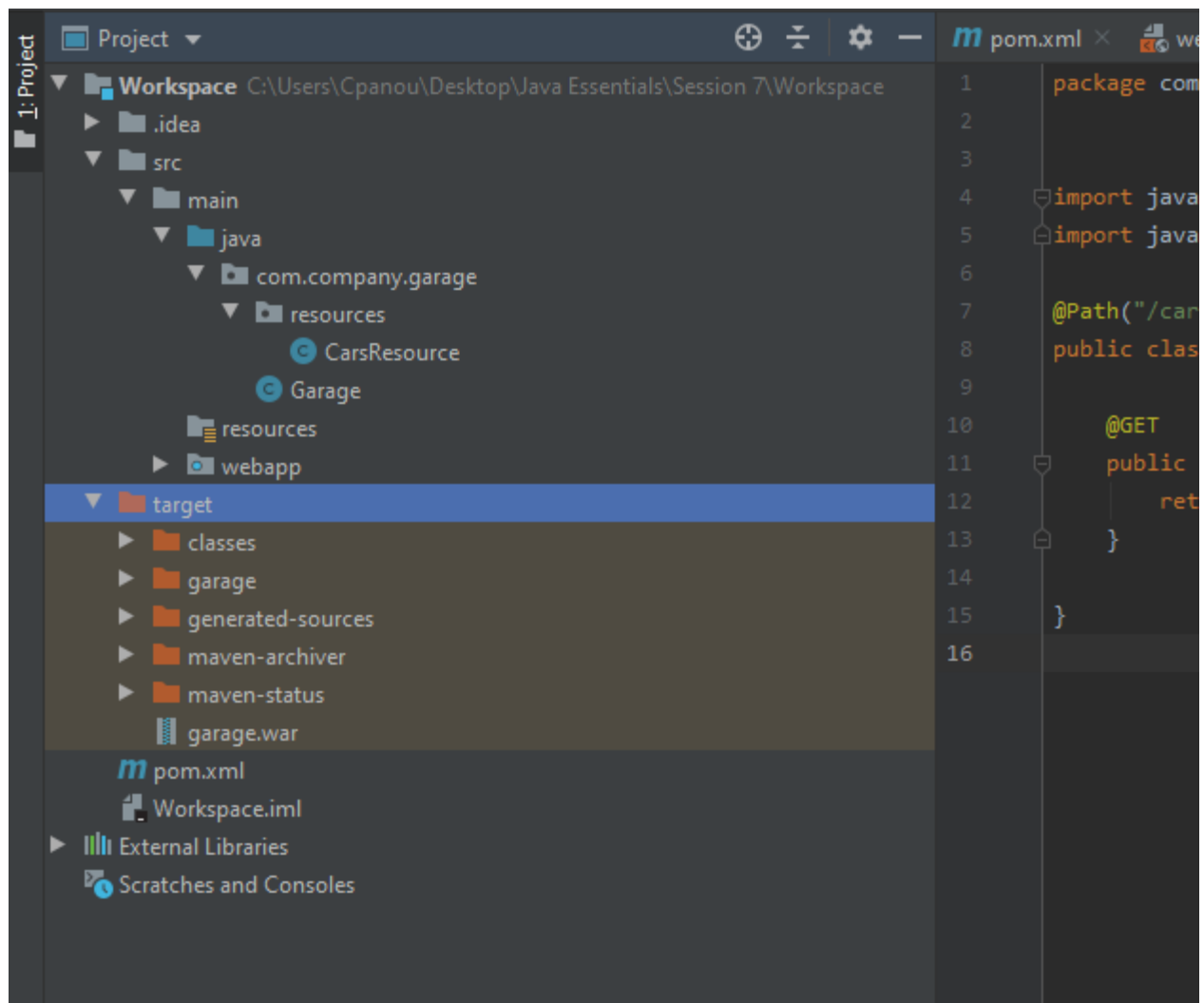
Choose the maven tab on the right column of your workspace:



This is the list of jobs provided by maven. The clean job will clear all built artifacts while the package job will create our artifact as described in the pom.xml file.

Run the package task

On your Project structure you should see a folder named target as in the example:



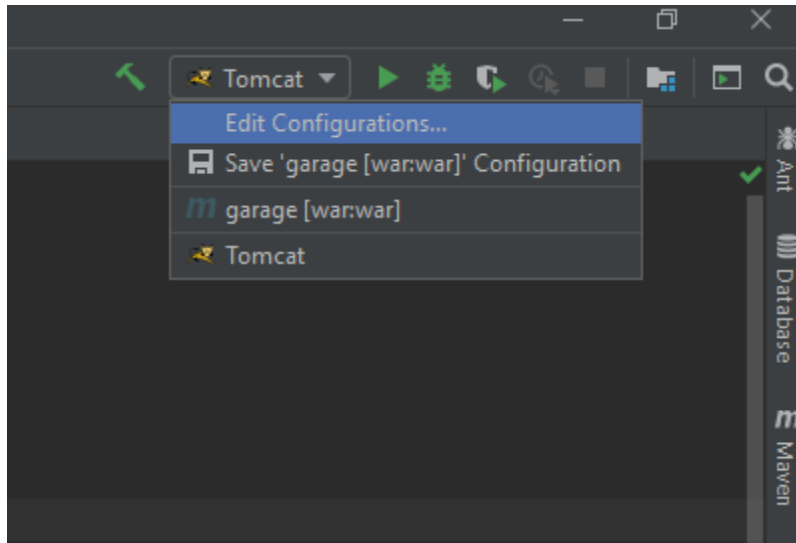
The garage.war file is the artifact which is ready to be deployed on our server.

## Integrating Tomcat with IntelliJ Workspace

1. Goto <https://tomcat.apache.org/download-90.cgi> and download the 32-bit/64-bit Windows Service Installer. When download is complete run the executable. When the process has finished, the server should be installed at

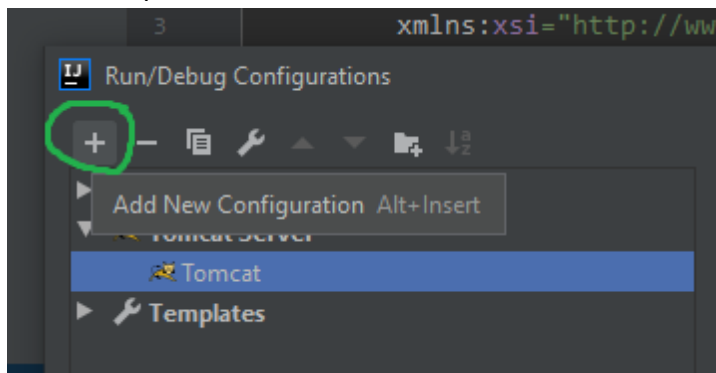
C:\Program Files\Apache Software Foundation\Tomcat 9.0

2. From the top of the screen choose the Edit Configurations option:

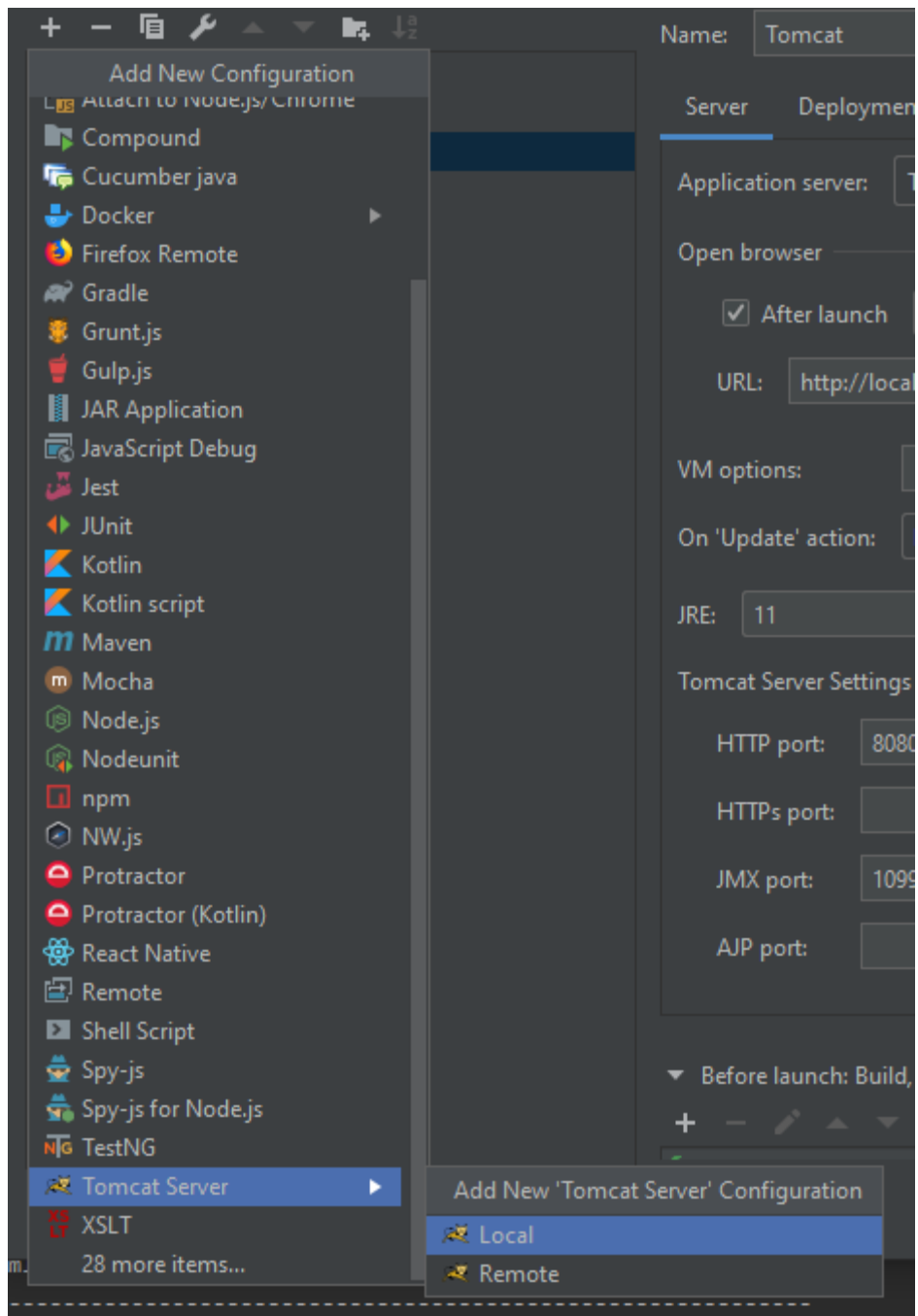


In the Pop up window press the “+” sign and then find **Tomcat Server** on the list and choose Local

step1:

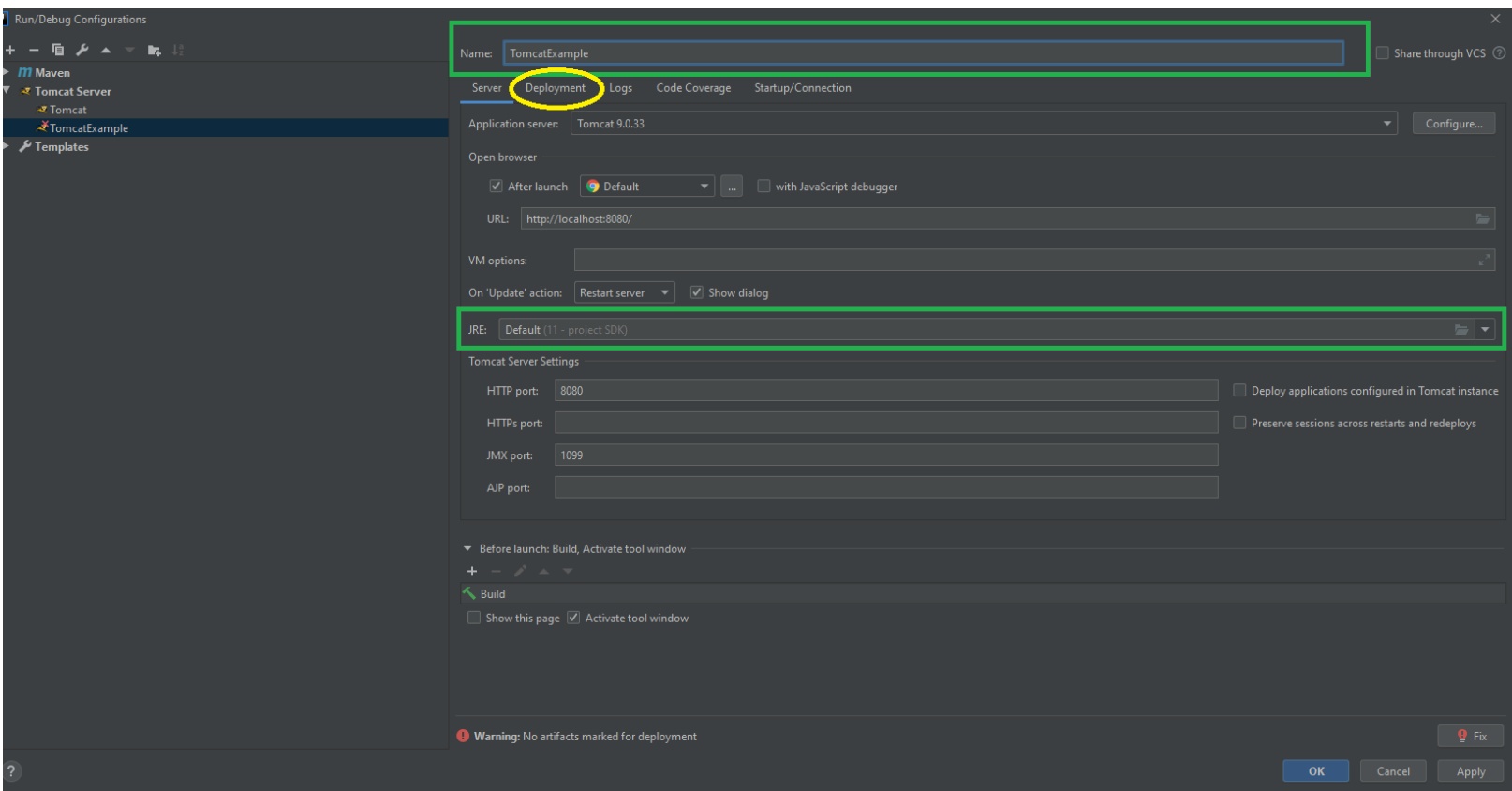


step2:



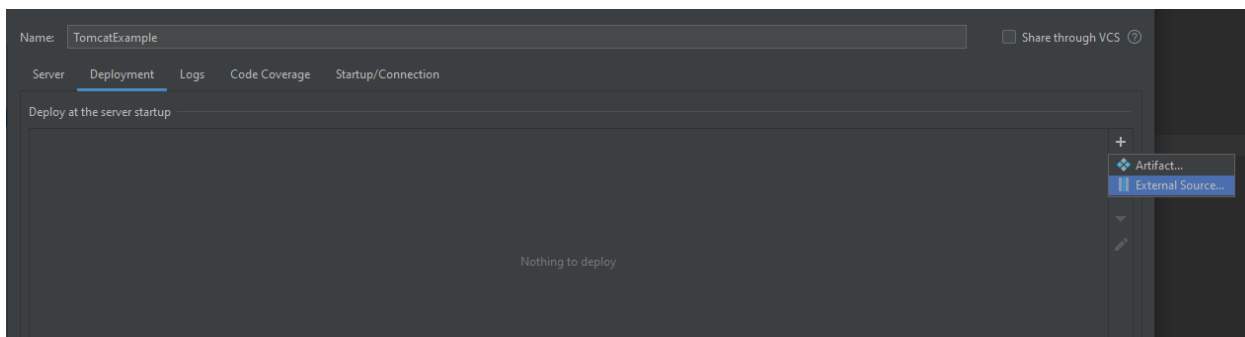


On the right side panel we configure the server:

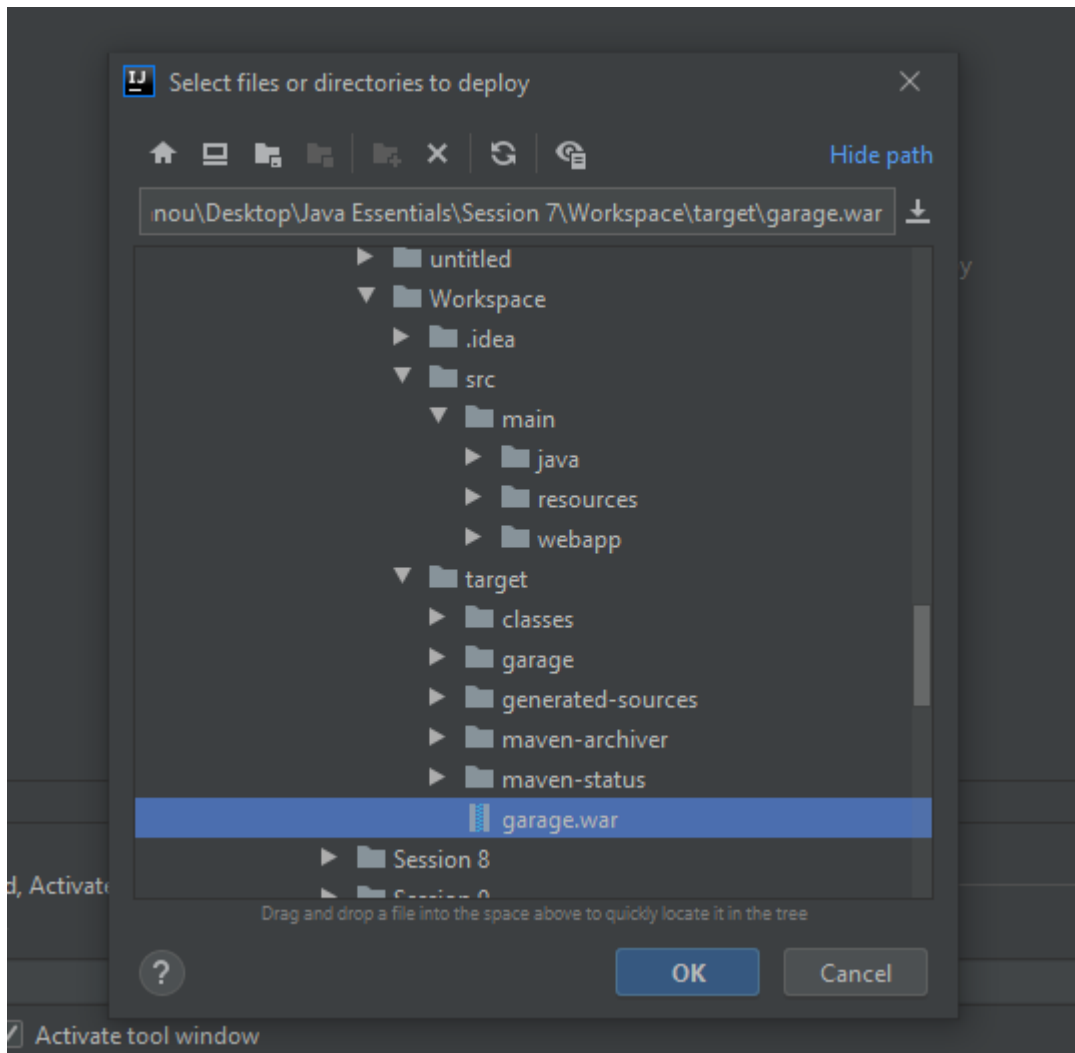


Name the Configuration Tomcat and make sure a JRE is provided. On the Application Server option select your **tomcat Installation**.

3. Goto the Deployments tab and on the right side choose the “+” icon and select External Source



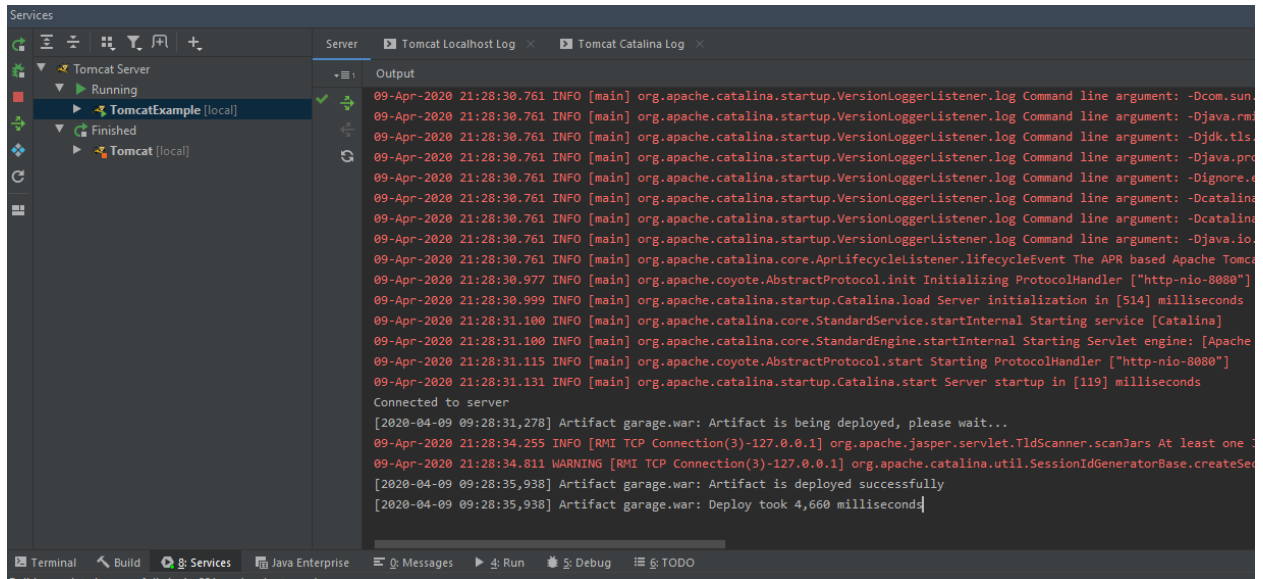
From the popup window choose the garage.war file in the target folder produces by maven.



Click OK then apply and OK. Now Run the Tomcat Configuration from the top panel

Every time you make changes to your code you run the package task from the Maven menu and the new .war will be deployed on your server.

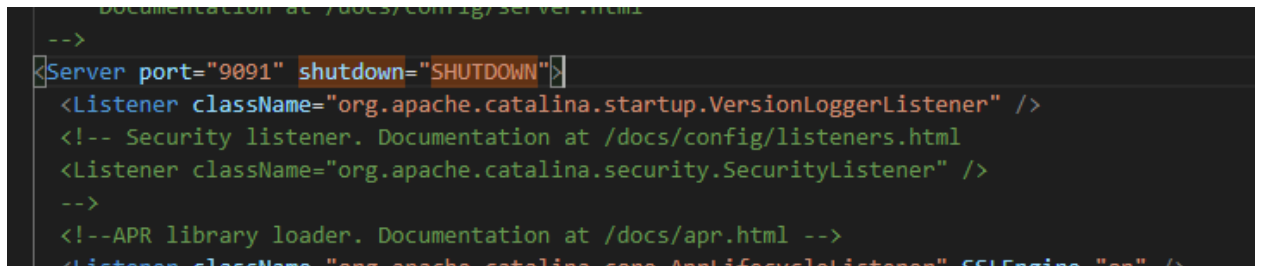
You should see the following message on the server logs:



If IntelliJ complains about the shutdown port : -1 open server.xml file located at:

(TOMCAT INSTALL DIR)/conf

Search for (Ctl+F) SHUTDOWN and change the port vaalue from -1 to 9090(or whatever)



Restart the server from IntelliJ

open a Browser enter the following url:

[http://localhost:8080/garage\\_war/garage/car](http://localhost:8080/garage_war/garage/car)

You should see the following:

