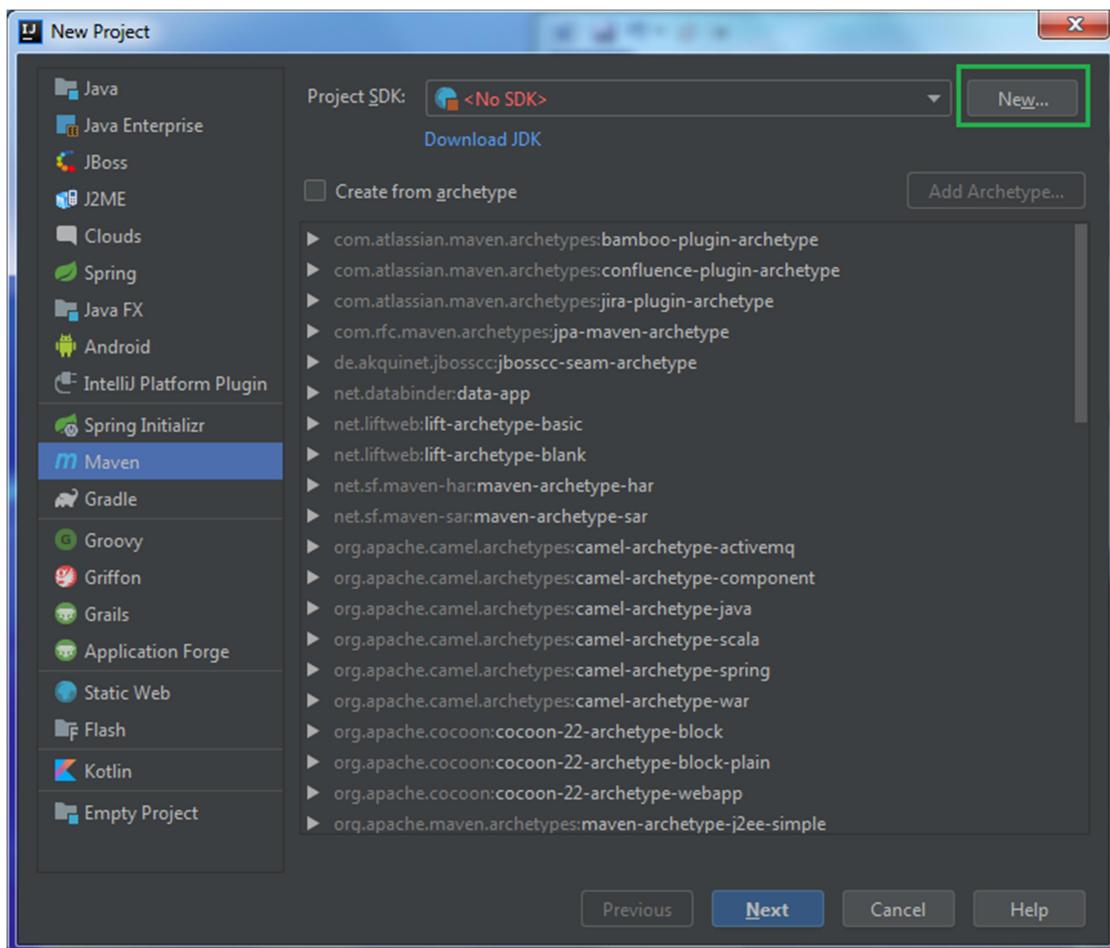


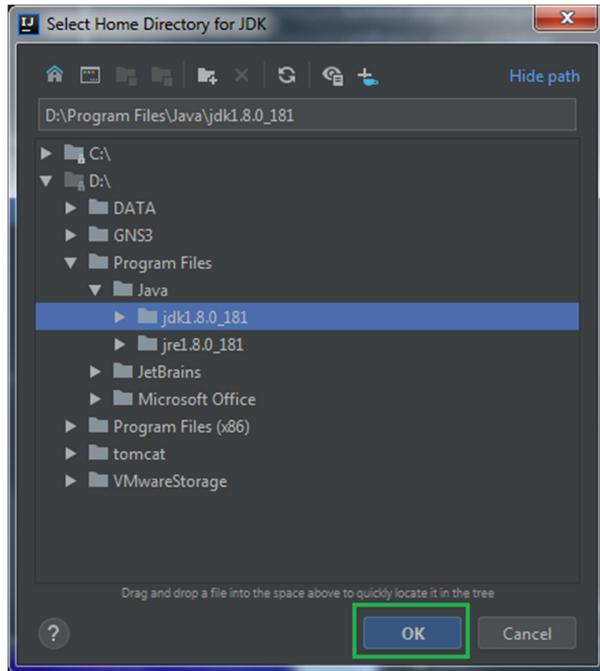
## Create a JavaEE webapp Maven Project on IntelliJ with Tomcat Support

1. Download and unzip [apache-tomcat-8.5.33-windows-x64.zip](#)
2. Download and install JDK 8
3. Install IntelliJ Ultimate Edition
4. Create a new project (*you **absolutely** must be connected to the Internet before creating the new project because of Maven and the webapp archetype (skeleton) we are going to select*)

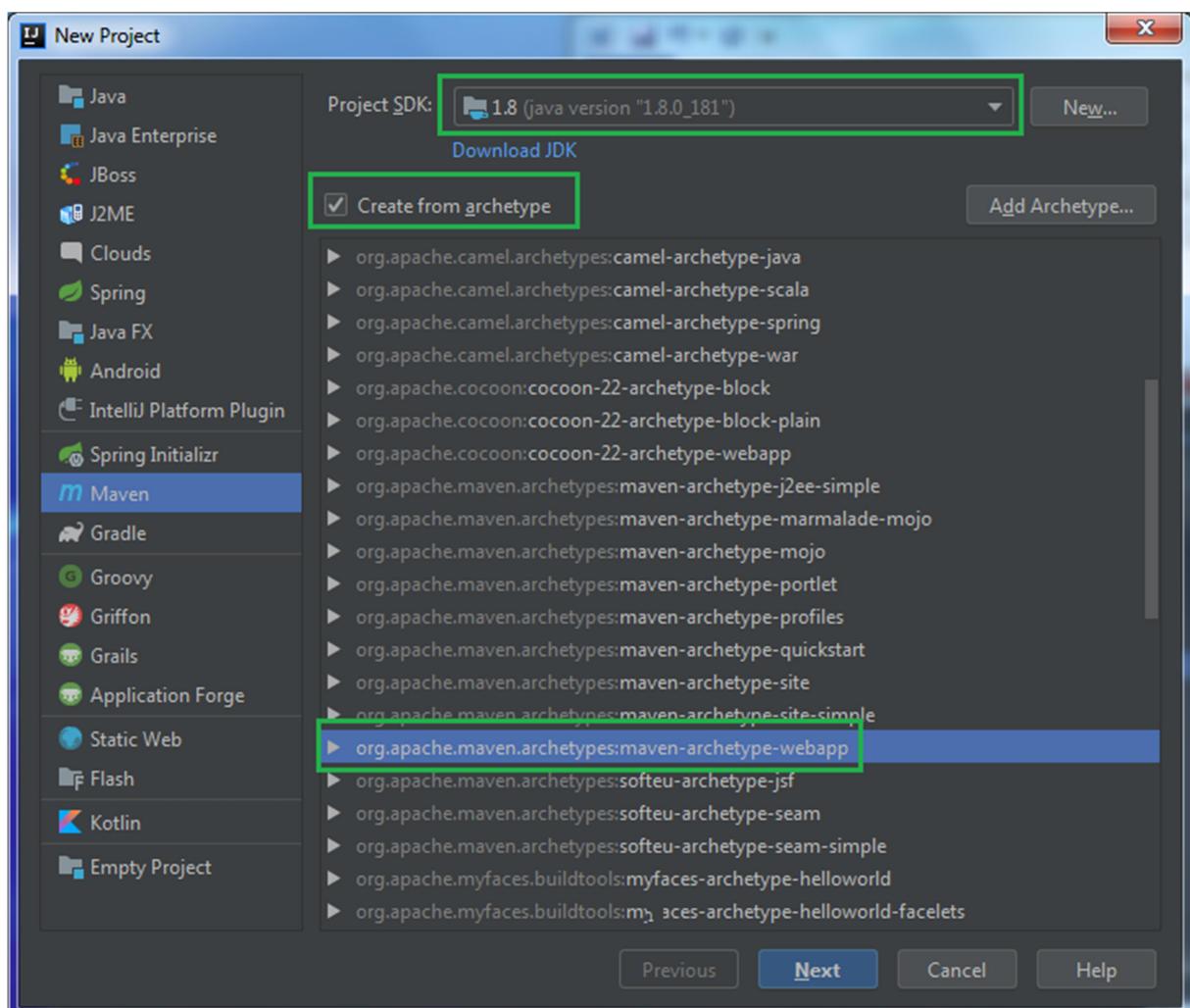


Select **Maven** and configure JDK by clicking on **New...**





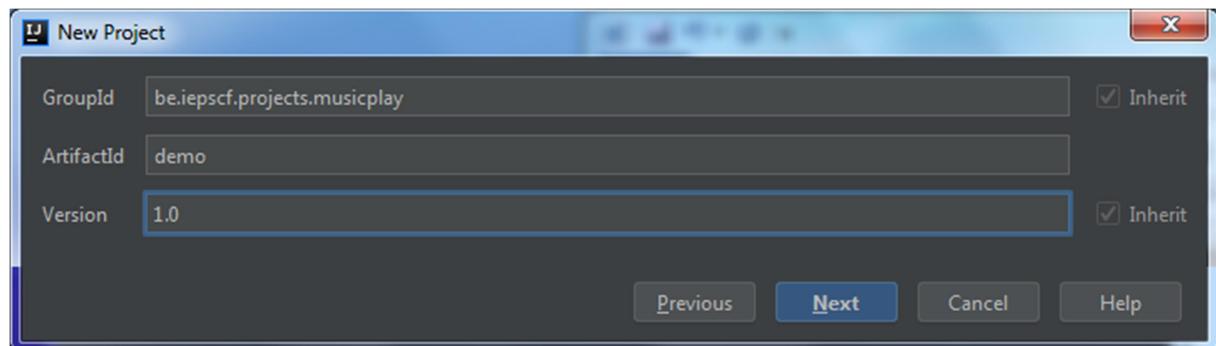
Verify that Project SDK is set to the correct Java version then select **Create from archetype** and click on **maven-archetype-webapp**.



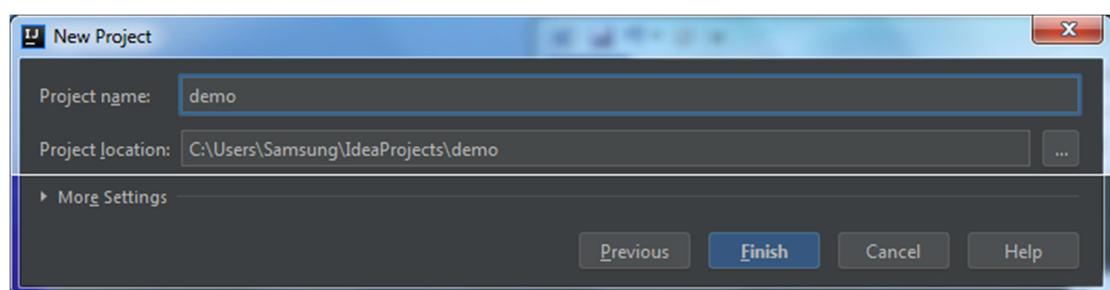
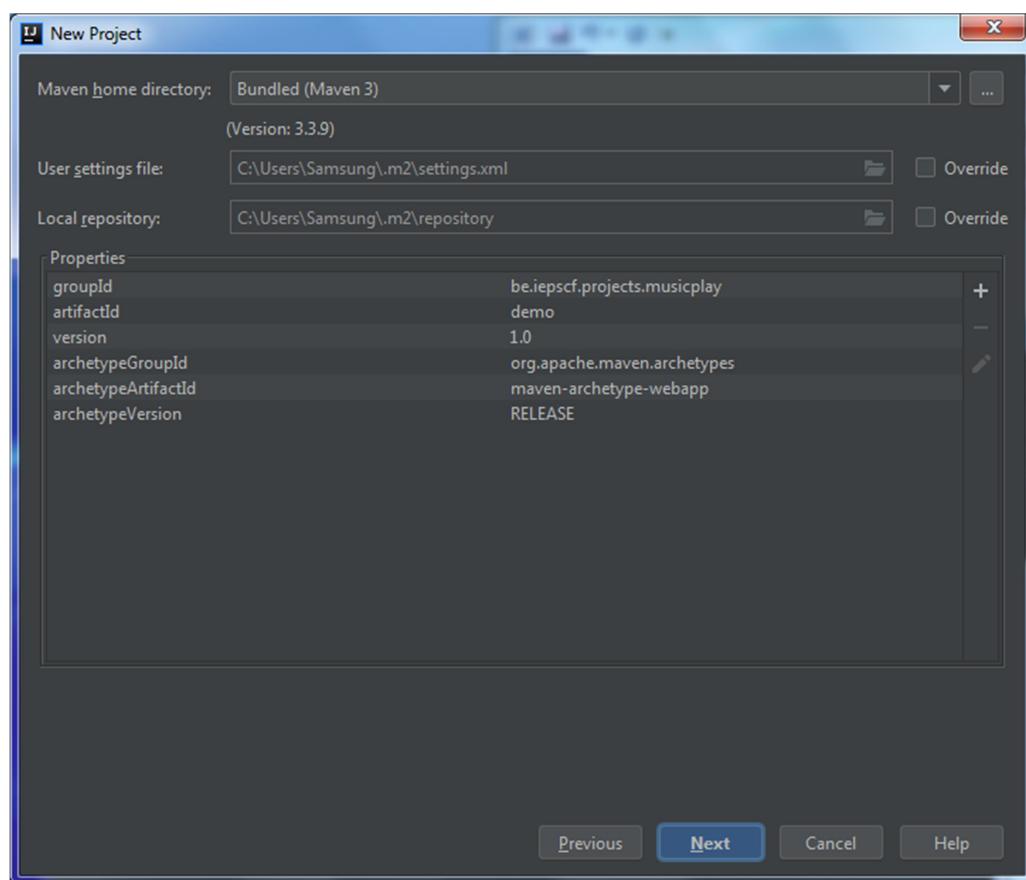
GroupId: *be.iepscf.projects.musicplay*

ArtifactId: *demo*

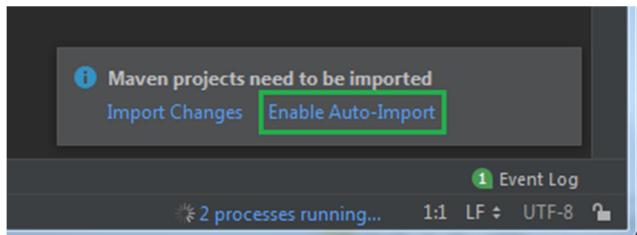
Version: *1.0*



Verify the settings configured previously and click on **Next** then **Finish**



Enable [auto-import feature for Maven](#) pom.xml. You must be connected to the Internet.



The **pom.xml** should look like the following when Maven is done downloading everything.

```
<?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>be.iepscf.projects.musicplay</groupId>
  <artifactId>demo</artifactId>
  <version>1.0</version>
  <packaging>war</packaging>

  <name>demo Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

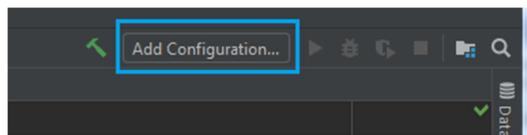
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <finalName>demo</finalName>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven
defaults (may be moved to parent pom) -->
      <plugins>
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.0.0</version>
        </plugin>
        <!-- see http://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin_bindings_for_war_packaging -->
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>

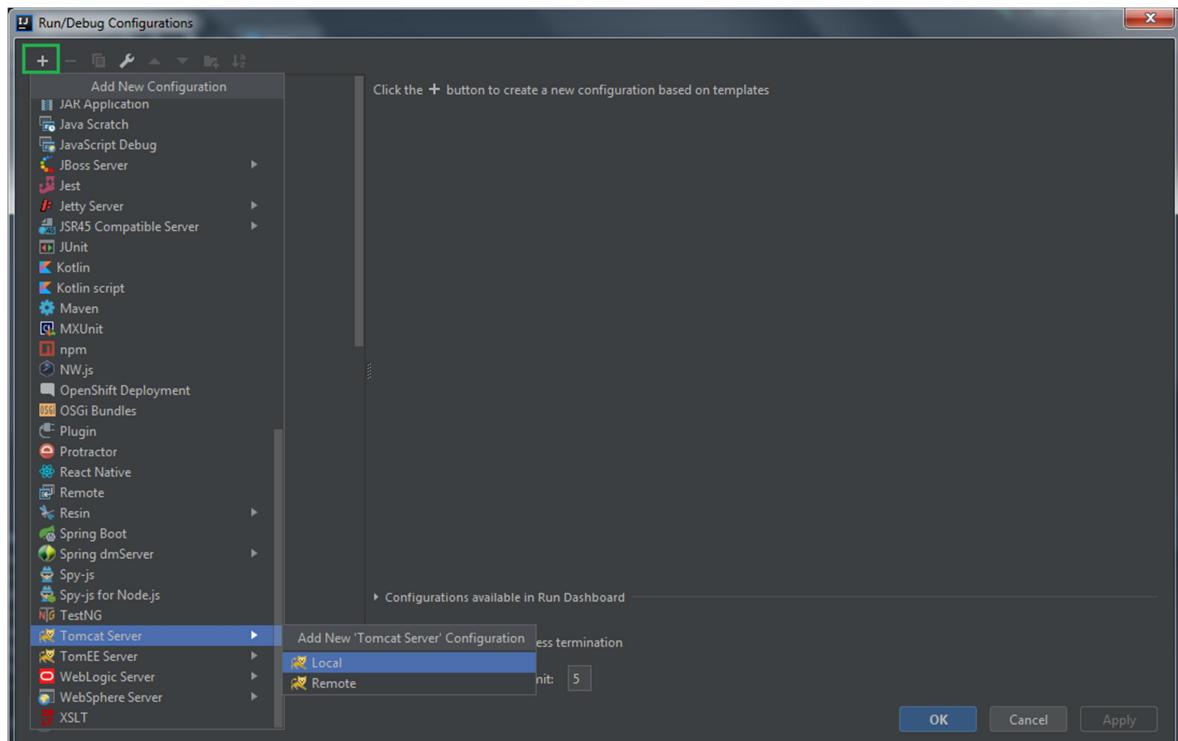
```

```
<plugin>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.7.0</version>
</plugin>
<plugin>
    <artifactId>maven-surefire-plugin</artifactId>
    <version>2.20.1</version>
</plugin>
<plugin>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.2.0</version>
</plugin>
<plugin>
    <artifactId>maven-install-plugin</artifactId>
    <version>2.5.2</version>
</plugin>
<plugin>
    <artifactId>maven-deploy-plugin</artifactId>
    <version>2.8.2</version>
</plugin>
</plugins>
</pluginManagement>
</build>
</project>
```

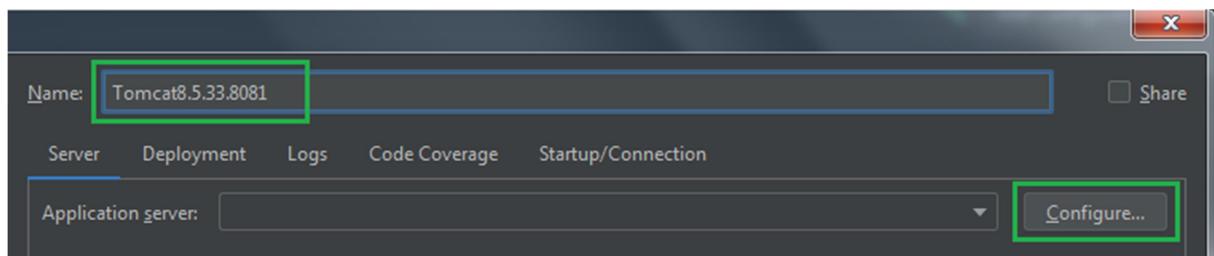
Click on **Add Configuration...** in the upper right corner



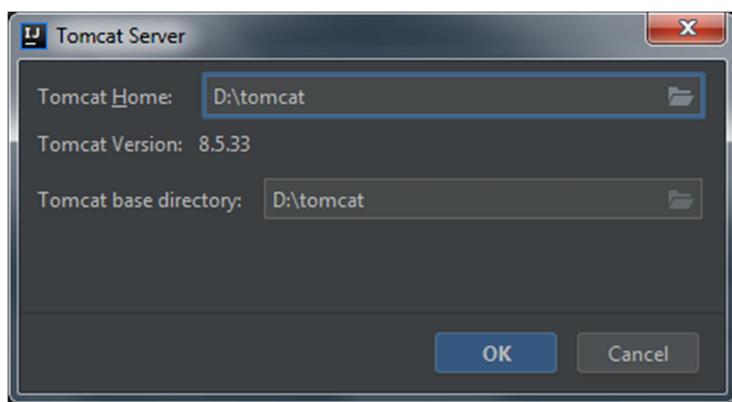
Click on “+” then select **Tomcat Server -> Local**



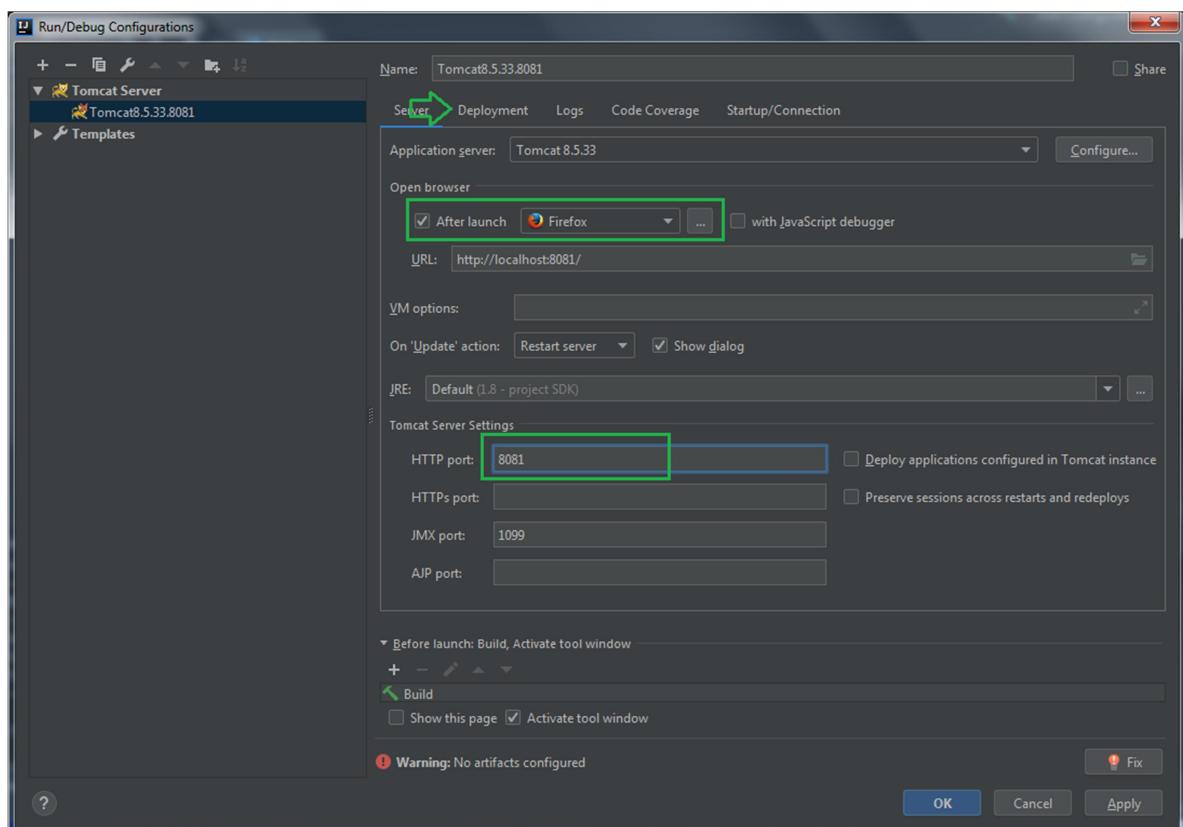
Enter a name then click on **Configure...**



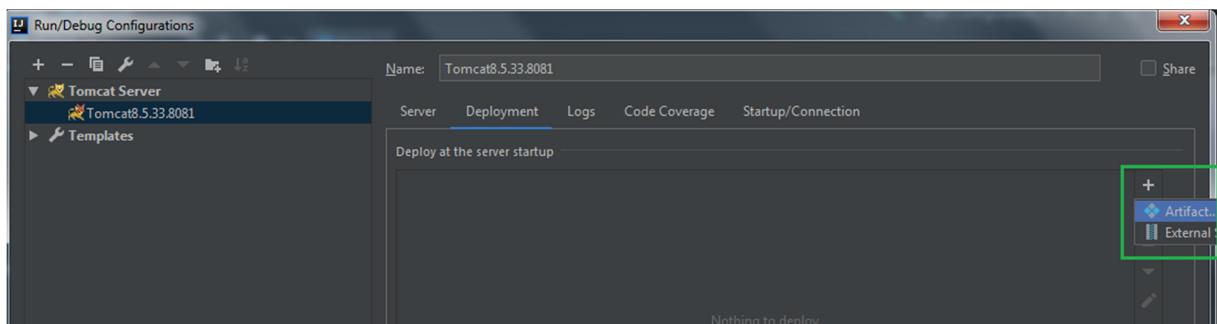
Indicate the directory where Tomcat has been installed and click on **OK**



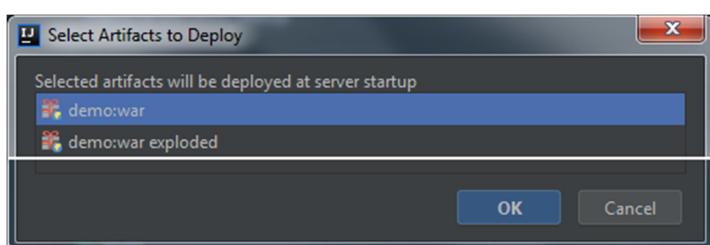
Change the port to 8081, select the browser you want the IDE to open when your project is build and deployed on your local Tomcat, then click on the **Deployment** tab.



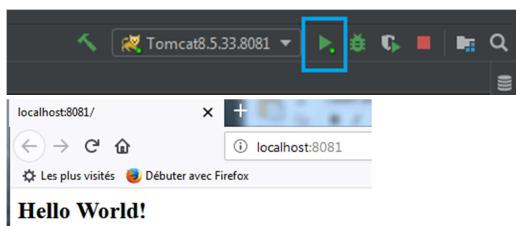
Click on “+” then **Artifacts...**



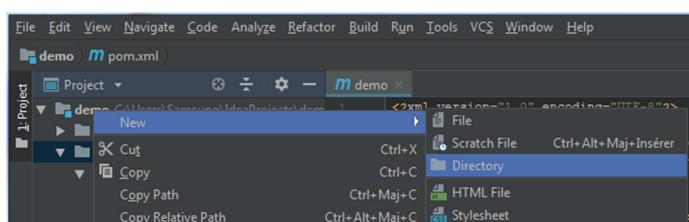
Select **demo:war** then click on **OK**



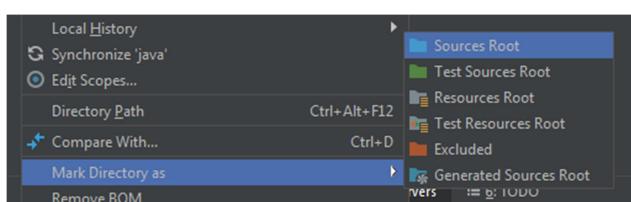
Click on **Run** next to Tomcat in the upper right corner of the application. Your browser should open automatically and you should see “Hello World!”.



Right-click on **src/main** then **New** then **Directory**



Call the new folder **java** then right-click on it and select **Mark Directory As Sources Root**



Edit the **pom.xml** and add the following dependencies inside the `<dependencies>` tag below the junit dependency.

```
<!-- https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-jsp-api -->
<dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>tomcat-jsp-api</artifactId>
    <version>8.5.33</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-servlet-
api -->
<dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>tomcat-servlet-api</artifactId>
    <version>8.5.33</version>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/org.webjars/webjars-servlet-2.x -->
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>webjars-servlet-2.x</artifactId>
    <version>1.5</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.webjars/jquery -->
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>jquery</artifactId>
    <version>3.3.1</version>
</dependency>

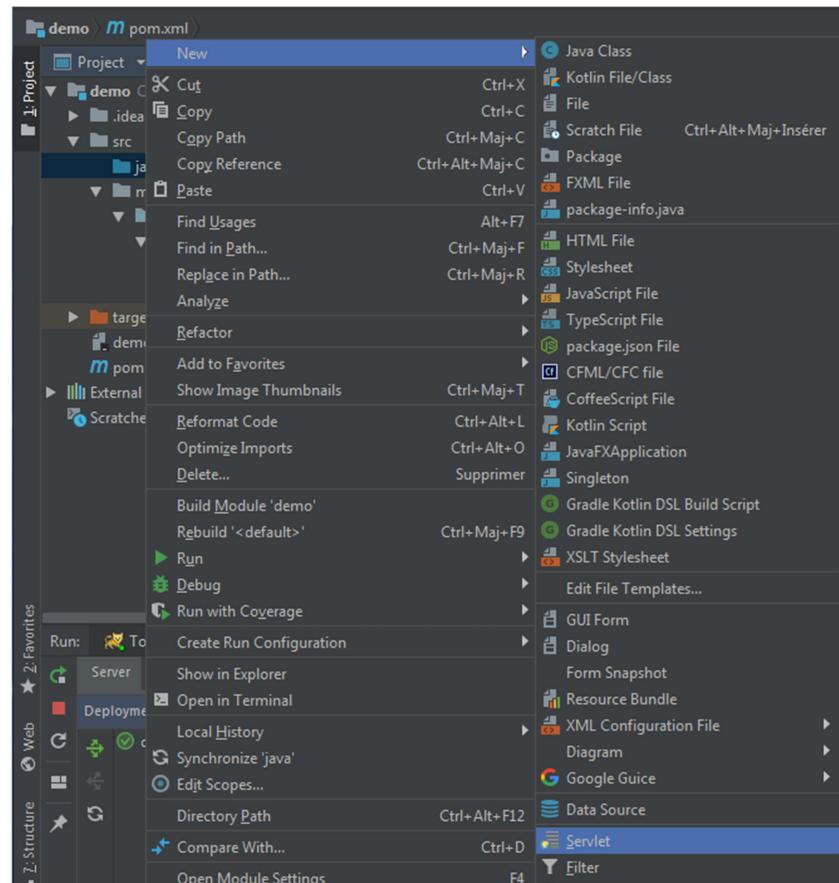
<!-- https://mvnrepository.com/artifact/org.webjars/bootstrap -->
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>bootstrap</artifactId>
    <version>4.1.3</version>
    <exclusions>
        <exclusion>
            <groupId>org.webjars</groupId>
            <artifactId>jquery</artifactId>
        
```

```

        </exclusion>
    </exclusions>
</dependency>

```

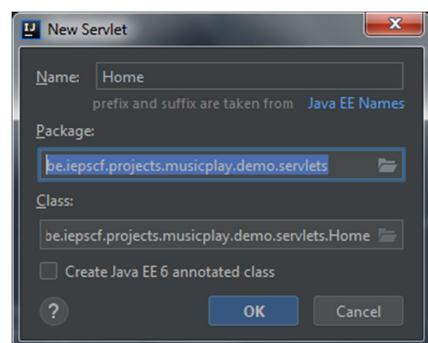
Right-click on the java folder and select **New** then **Servlet**



Name: *Home*

Package: *be.iepscf.projects.musicplay.demo.servlets*

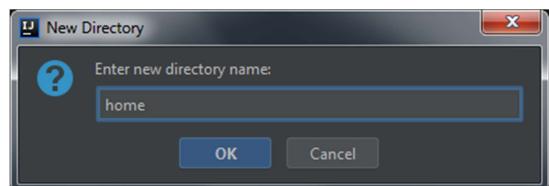
Unselect “Create Java EE 6 annotated class”



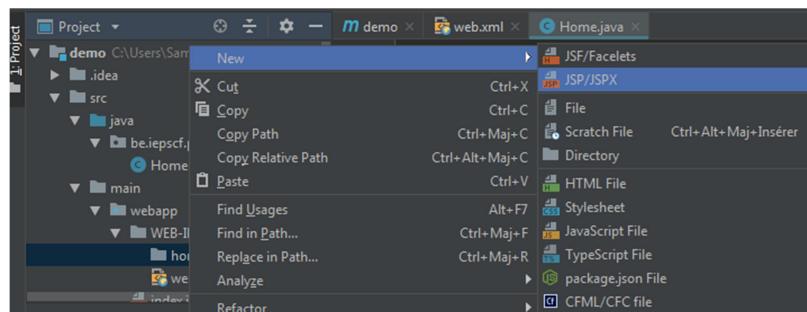
Inside the **doGet** method, add the following:

```
String title = "Home";
int servletInfo = this.getServletContext().getMajorVersion();
request.setAttribute("title", title);
request.setAttribute("servletInfo", servletInfo);
this.getServletContext().getRequestDispatcher("/WEB-
INF/home/index.jsp").forward(request, response);
```

Right-click on **WEB-INF** folder then **New** then **Directory** and create a new directory named *home*



Then right-click on the newly created folder and select **New** then **JSP/JSPX**. Name it *index*.



The file should contain the following content:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title><c:out value="${title}" /></title>
    <link rel="stylesheet"
href="webjars/bootstrap/4.1.3/css/bootstrap.min.css">
</head>
<body>
<div class="jumbotron">
    <h1 class="display-4">Servlet Version : <c:out
value="${servletInfo}" /></h1>
    <p class="lead">This is a simple hero unit, a simple jumbotron-style
component for calling extra attention to featured content or
information.</p>
    <hr class="my-4">
    <p>It uses utility classes for typography and spacing to space content
out within the larger container.</p>
    <a class="btn btn-primary btn-lg" href="#" role="button">Learn more</a>
</div>

<script src="webjars/jquery/3.3.1/jquery.min.js"></script>
```

```
<script src="webjars/bootstrap/4.1.3/js/bootstrap.min.js"></script>
</body>
</html>
```

Edit the **web.xml** file. The file should contain the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://java.sun.com/xml/ns/javaee"
          xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
          version="3.0">

    <display-name>Archetype Created Web Application</display-name>
    <servlet>
        <servlet-name>WebjarsServlet</servlet-name>
        <servlet-class>org.webjars.servlet.WebjarsServlet</servlet-class>
        <load-on-startup>2</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>WebjarsServlet</servlet-name>
        <url-pattern>/webjars/*</url-pattern>
    </servlet-mapping>
    <servlet>
        <servlet-name>Home</servlet-name>
        <servlet-
class>be.iepscf.projects.musicplay.demo.servlets.Home</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Home</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
</web-app>
```

We change the xml and web-app tag settings, otherwise we do not have access to the JSTL taglib.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns="http://java.sun.com/xml/ns/javaee"
          xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
          version="3.0">
```

We use Servlet version 3 but I was only able to configure the webjars for Bootstrap and Jquery with a Servlet 2 configuration type. Must be checked and eventually modified.

<https://www.webjars.org/documentation#servlet2>

**Delete WEB-INF/index.jsp.** Run the project on Tomcat (restart option). The browser should open automatically on <http://localhost:8081> and you should see the Bootstrap-styled jumbotron on the page + the Servlet Version set to 3 here.

