Mavenプロジェクトの作成方法

〜〜〜

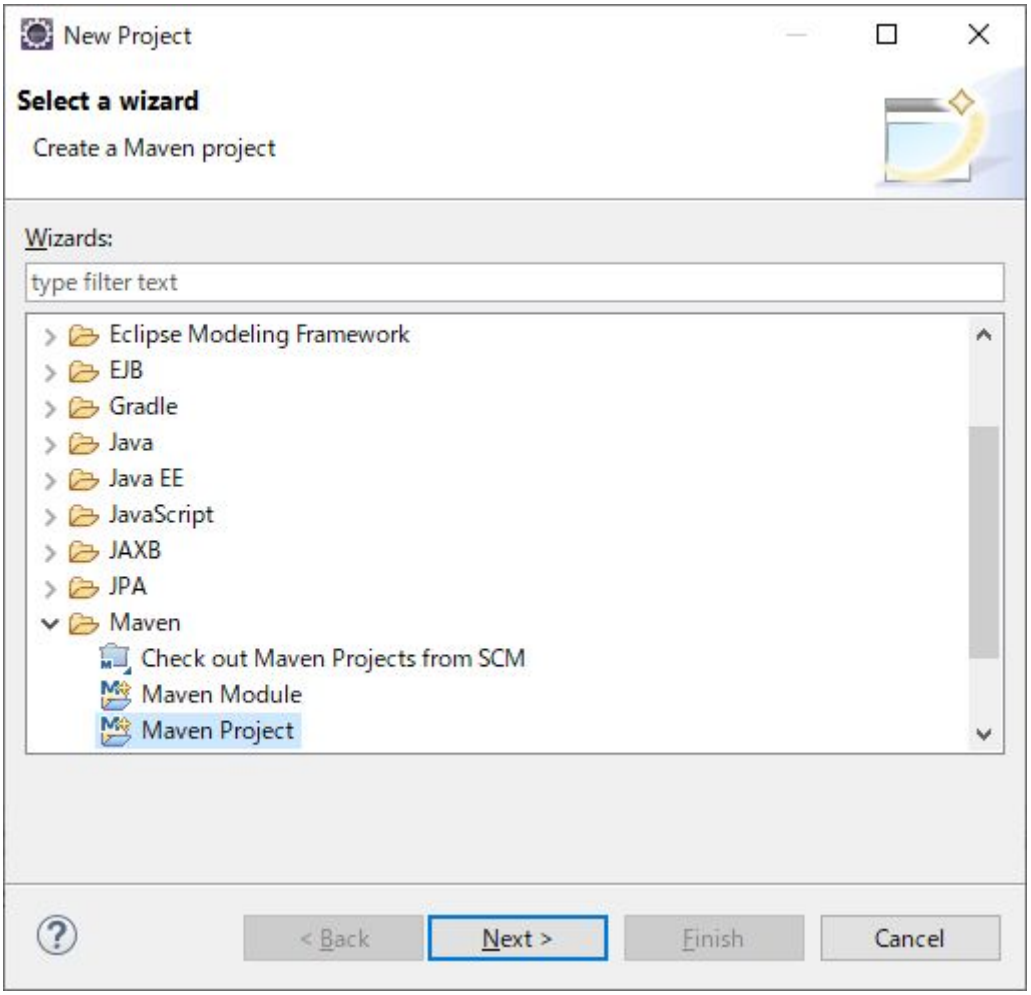
〜〜〜

１．前提条件

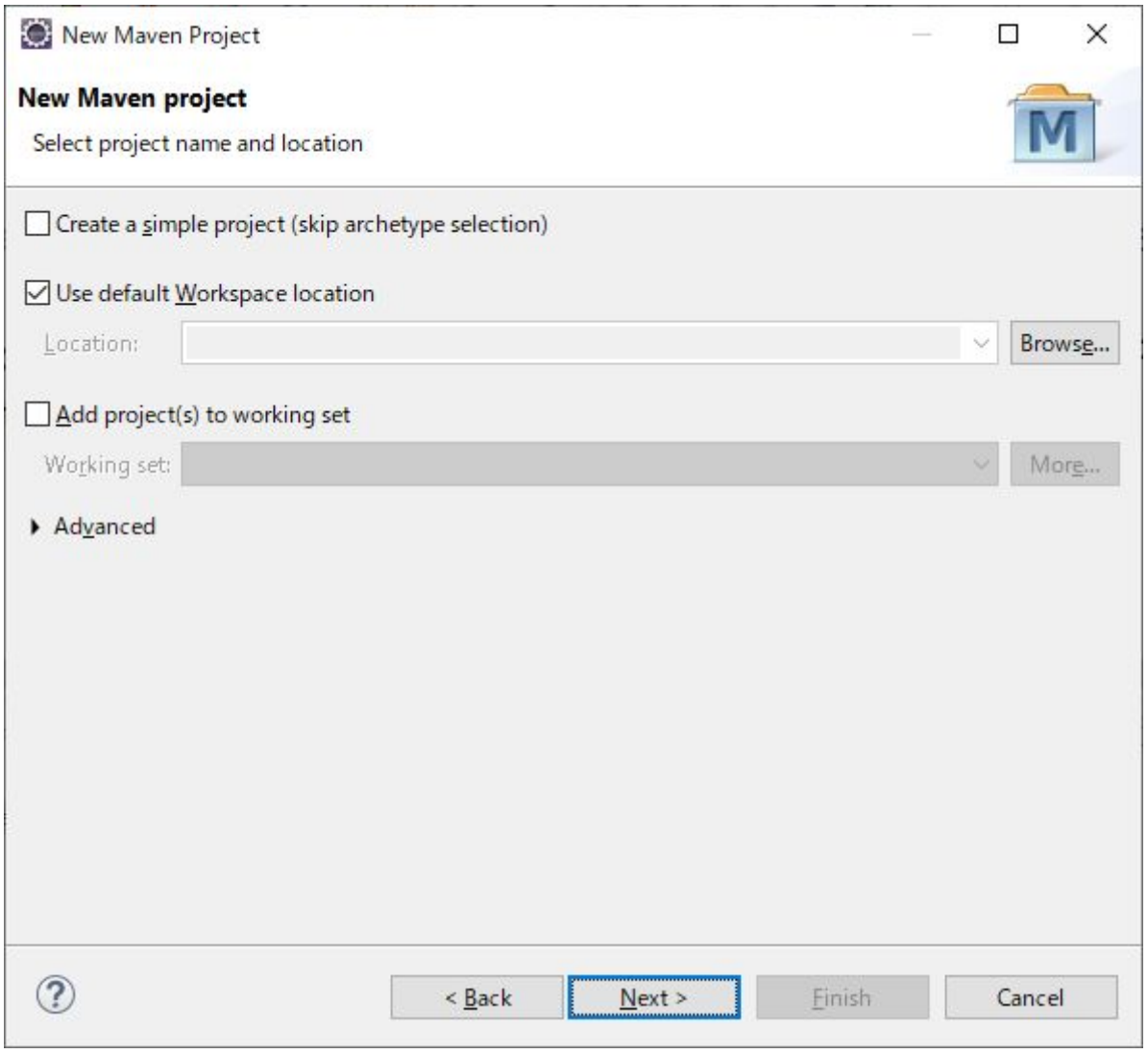
- Eclipse IDE for Enterprise Java Developers.
  - eclipse-jee-2019-03-R-win32-x86_64
  - Version: 2019-03 (4.11.0)、Build id: 20190314-1200
- Java1.8
  - jdk-8u202-windows-x64
- Tomcat 8.5
  - apache-tomcat-8.5.39
- MySQL 8
  - mysql-installer-community-8.0.15.0

２．Mavenプロジェクトを作成する

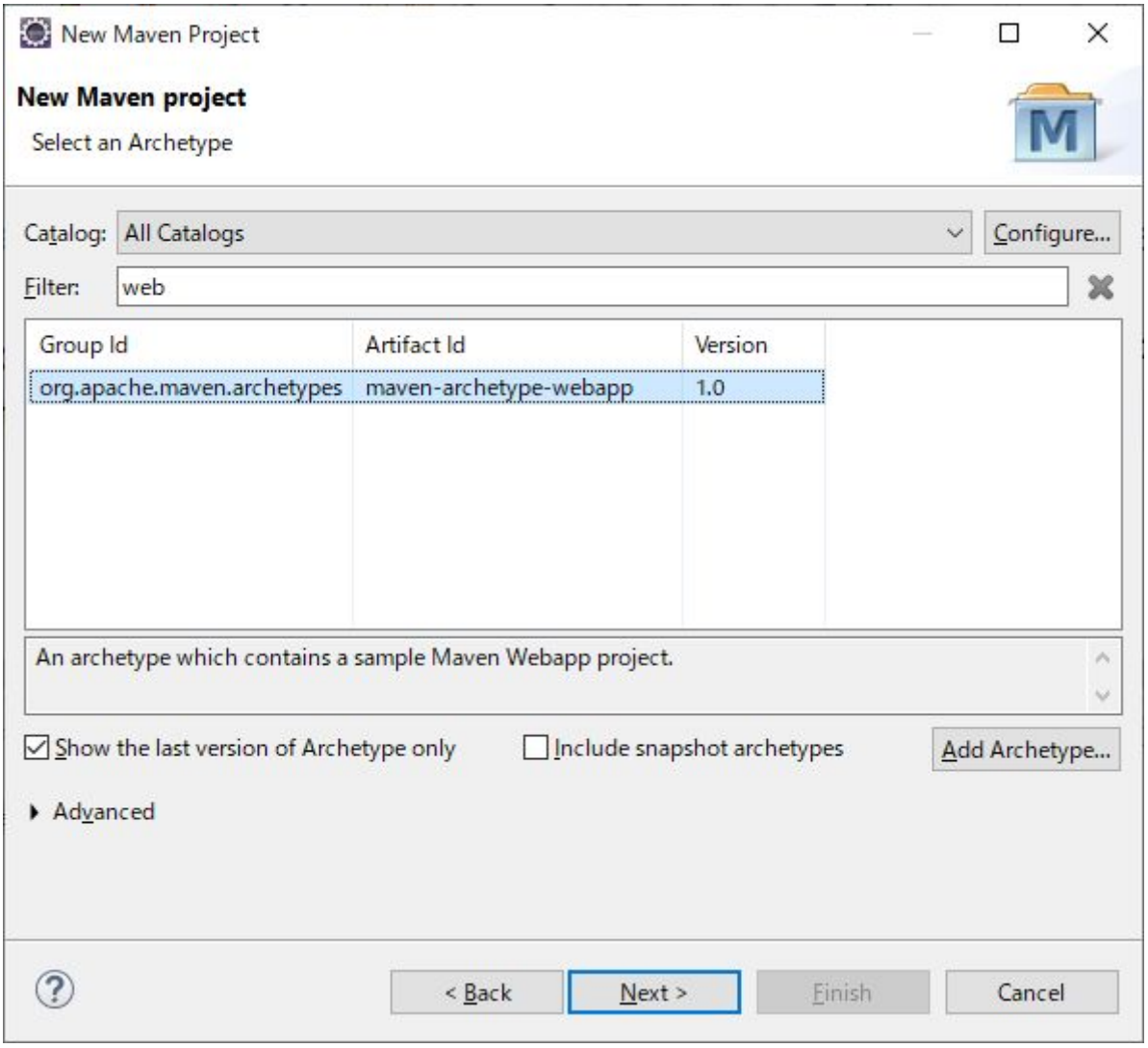
「Project Explorer」にて右クリックして、New -> Project -> Maven Projectを選択し、Next を押す



この画面はそのままで、Nextを押す



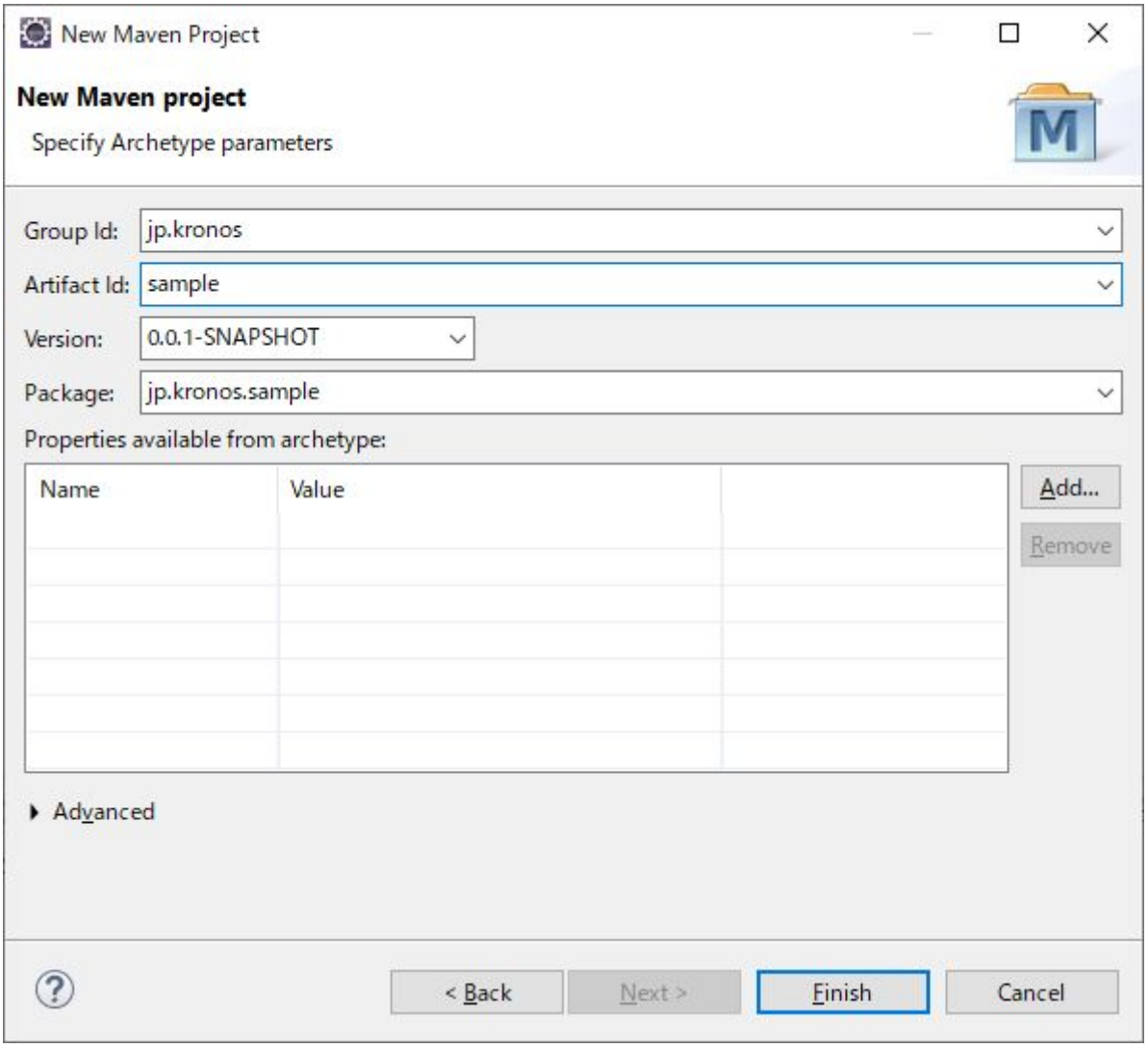
Filterにwebと入力して、一覧に出てきた「maven-archetype-webapp」を選択し、Nextを押す



Group IdとArtifact Idにそれぞれ適当な文字列を入力して、Finishを押す

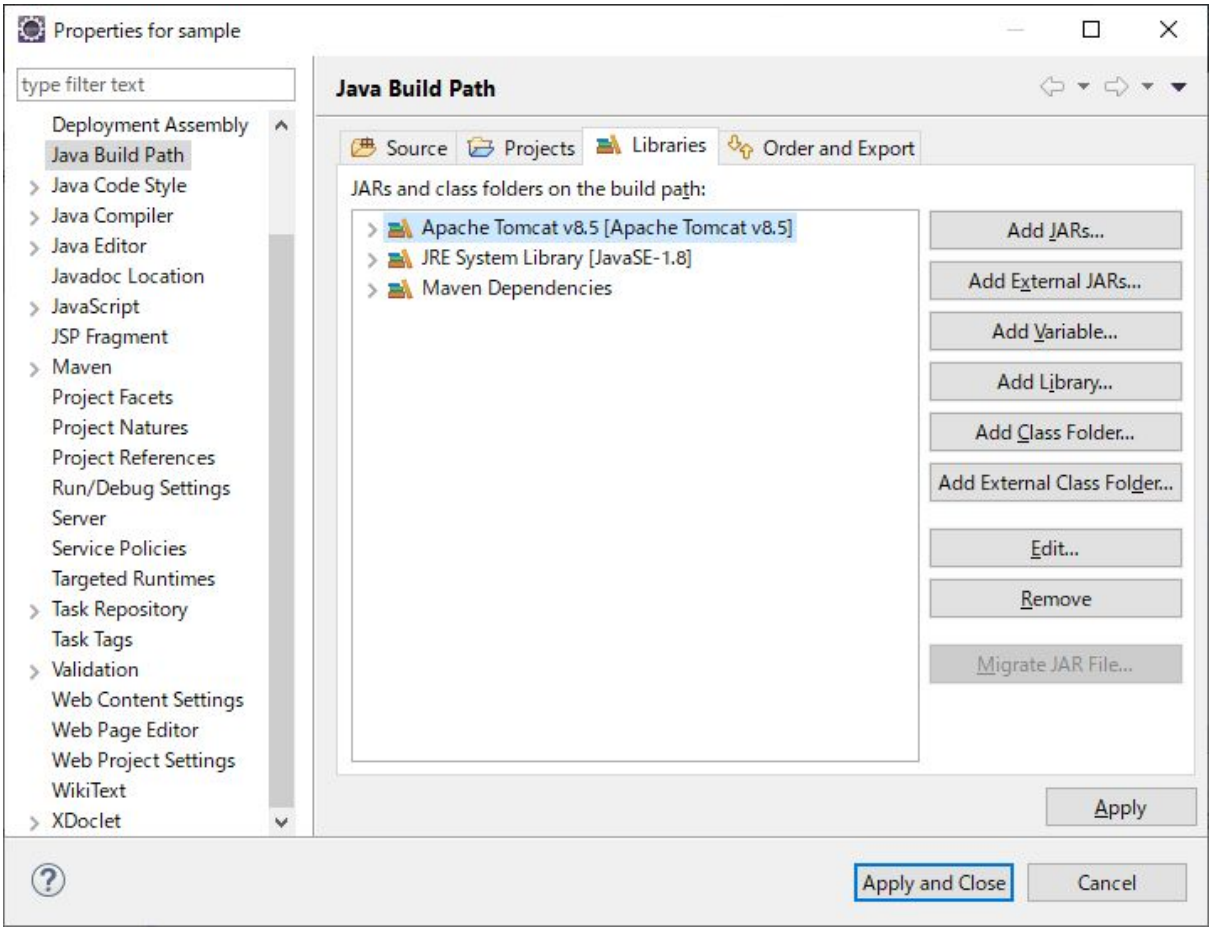
※Group Idは、他の全てのプロジェクトと区別するための名前です。 一般に使われるのは ドメイン名を逆にしたものです。

※Artifact Idはバージョンを除いた jarやwar ファイルの名前です。 プロジェクト名と言ったほうがわかりやすいでしょうか。



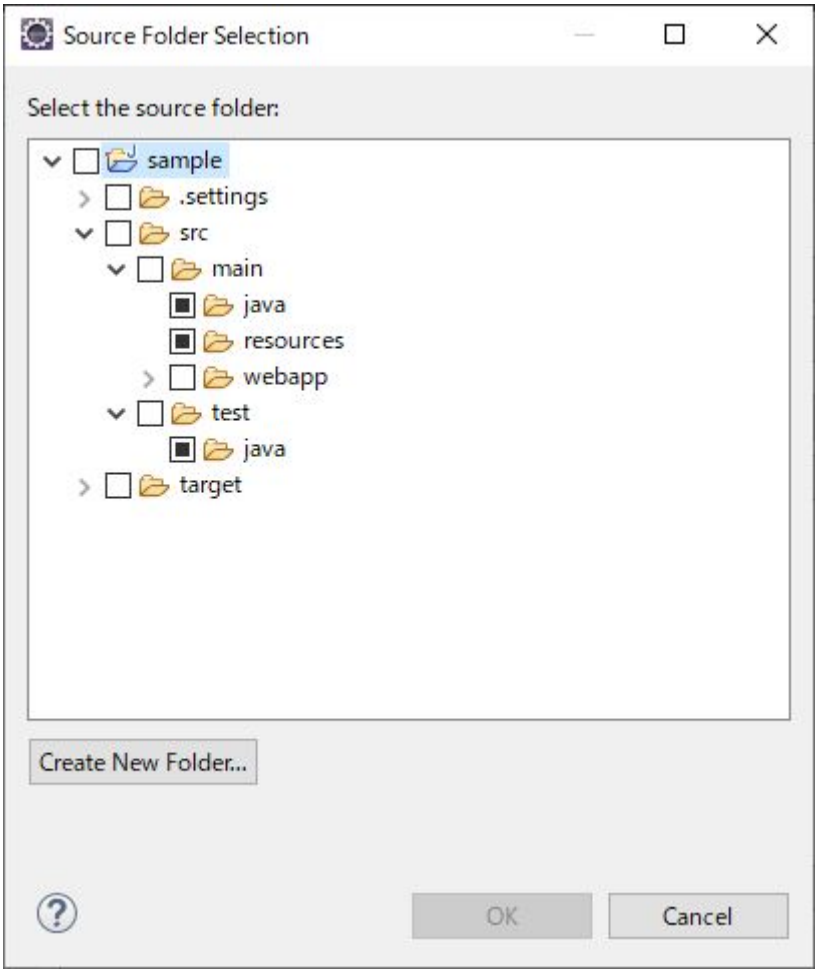
３．Tomcatのライブラリをプロジェクトに加える

プロジェクトを右クリックして、Build Path -> Configure Build Pathを選んで
Libraries タブで、Add Library -> Server Runtime で対象のサーバ（Tomcat v8.5）を選択し、Finish

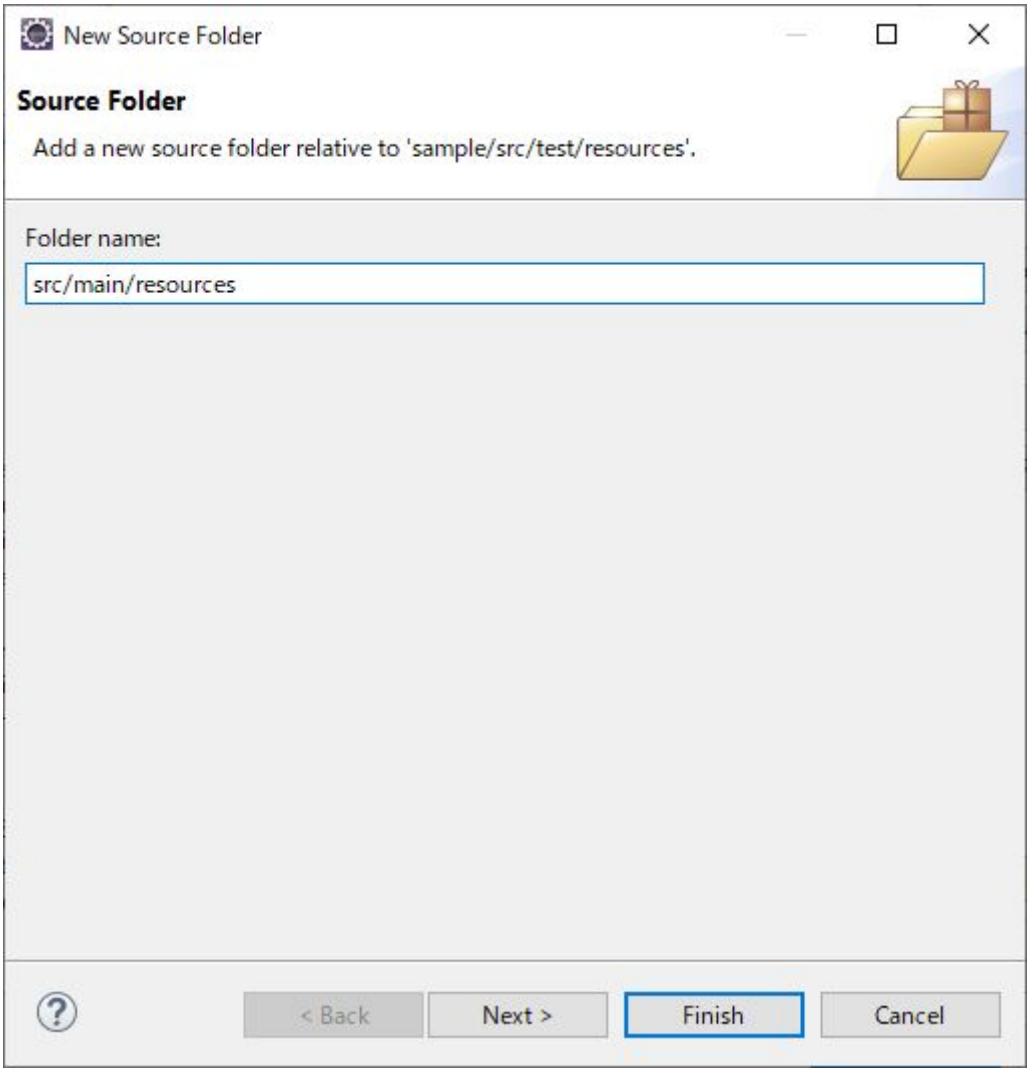


４．classpathを加える

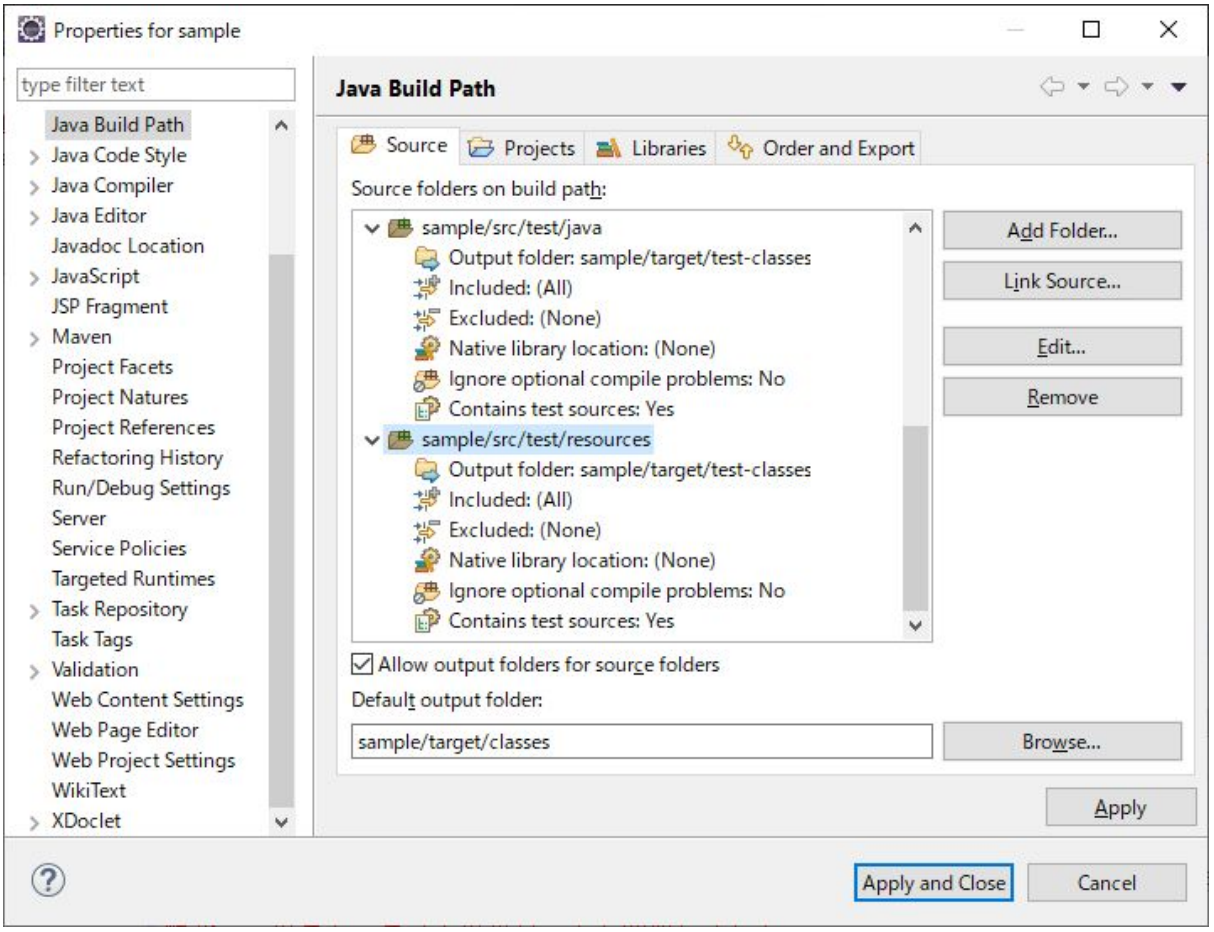
同じく、Configure Build PathのSourceタブで、Add Foderを押し、「Source Folder Selection」のCreate New Folderを押す



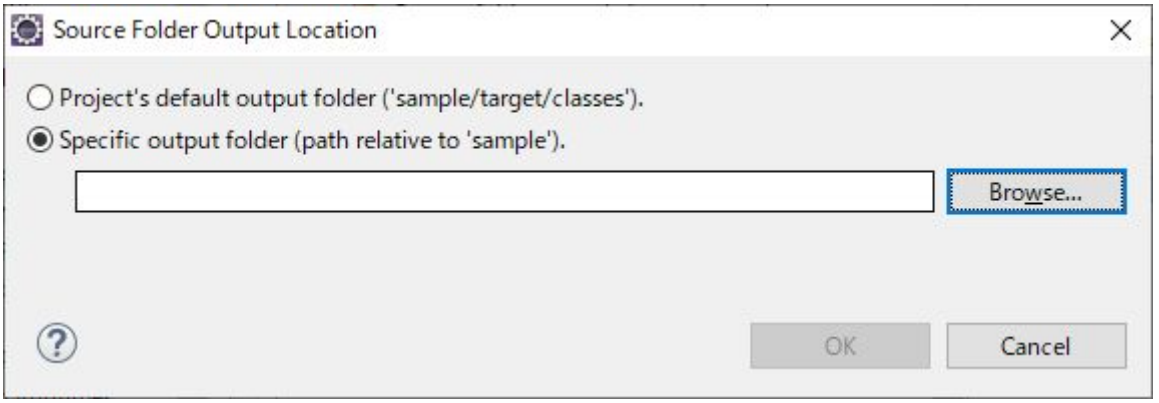
Folder nameに「src/test/resources」と入力し、Finishを押し、OKを押す



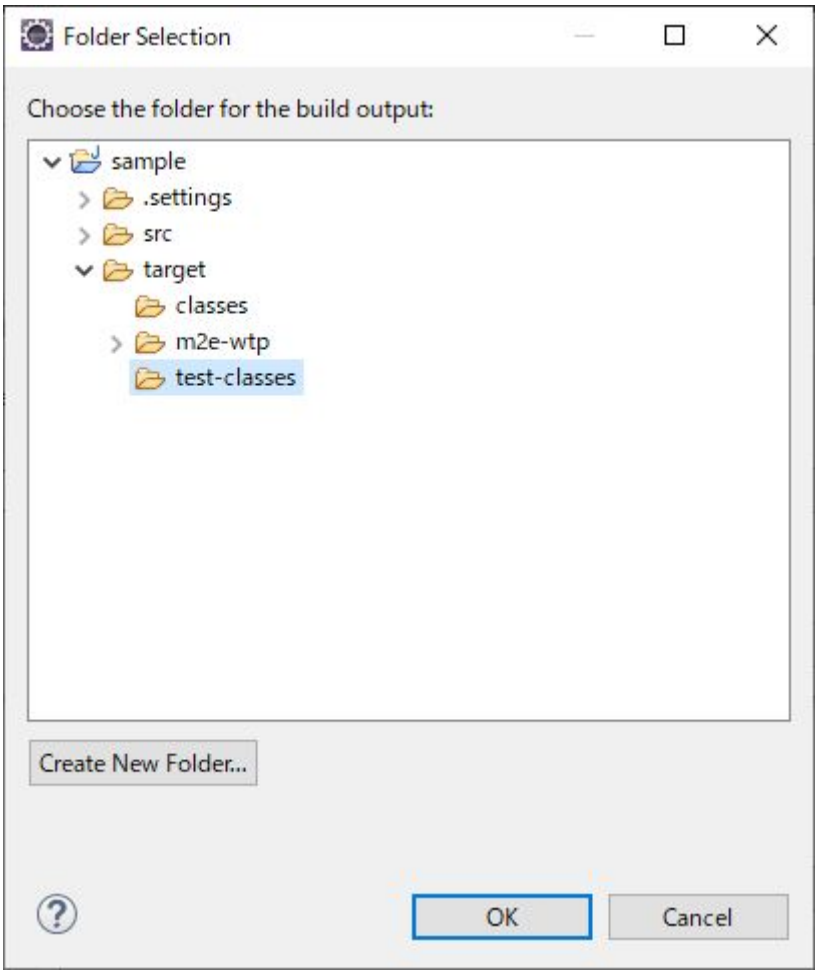
src/test/resourcesを開き、Output folderをダブルクリックする



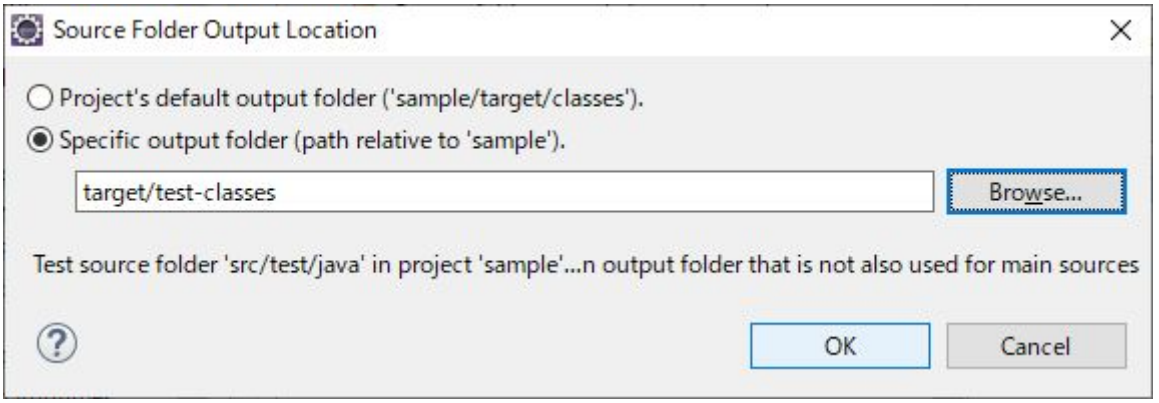
「Project's default output folder」から「Specific output folder」を選び、Browseを押す



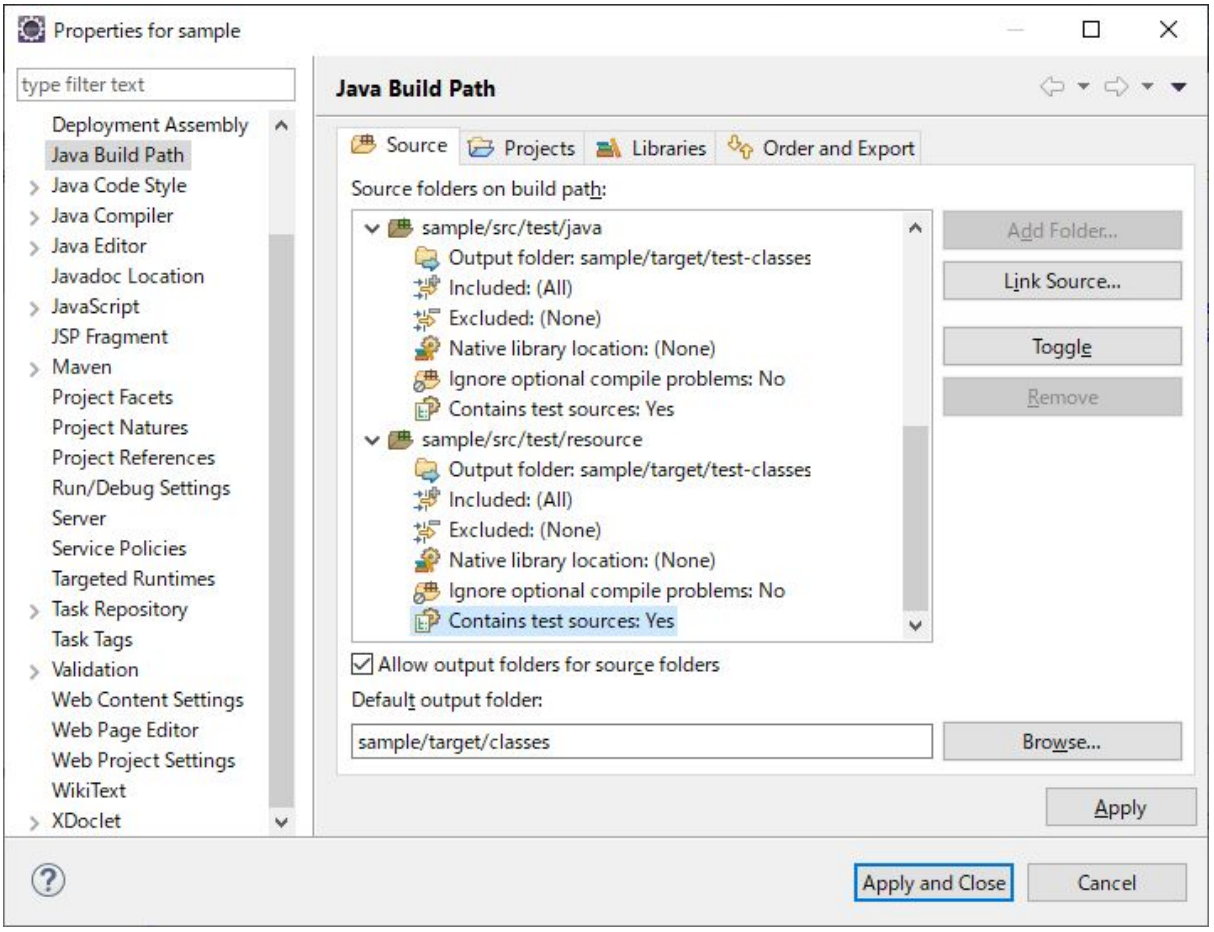
target/test-classを選び、OKを押す



OKを押す



Contains test resourcesがNoになっているが、ダブルクリックしてYesにする
Apply and Closeを押す



５．pom.xmlを修正する

対象プロジェクトの直下にあるpom.xmlの<dependencies>と<build>を以下の内容に書き換える

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <java.version>1.8</java.version>
  <maven.compiler.target>${java.version}</maven.compiler.target>
  <maven.compiler.source>${java.version}</maven.compiler.source>
  <junit.version>4.12</junit.version>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
```

```
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.15</version>
</dependency>
<dependency>
  <groupId>org.dbunit</groupId>
  <artifactId>dbunit</artifactId>
  <version>2.5.4</version>
</dependency>
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-all</artifactId>
  <version>1.9.5</version>
</dependency>
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.12.0</version>
</dependency>
<dependency>
  <groupId>tomcat</groupId>
  <artifactId>servlet</artifactId>
  <version>4.1.36</version>
</dependency>
<dependency>
  <groupId>com.codeborne</groupId>
  <artifactId>selenide</artifactId>
  <version>4.12.1</version>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.webjars/bootstrap -->
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>bootstrap</artifactId>
  <version>4.0.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.webjars/jquery -->
<dependency>
    <groupId>org.webjars</groupId>
    <artifactId>jquery</artifactId>
    <version>3.3.1</version>
```

```
  </dependency>
 </dependencies>
 <build>
  <plugins>
   <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.1.0</version>
    <configuration>
     <failOnMissingWebXml>false</failOnMissingWebXml>
    </configuration>
   </plugin>
  </plugins>
 </build>
```

６．Dynamic Web Projectを3.1に変更する

@WebServletを有効にするために、以下の設定をします。

エクスプローラーで、workspaceにある対象のプロジェクトのディレクトリの中にある.settingsディレクトリ内の「org.eclipse.wst.common.project.facet.core.xml」というファイルをテキストエディタで開き、「jst.web」のversionを「3.1」に修正する

また、以下の内容を加えて、保存する（ファイルを閉じる）

```
 <fixed facet="wst.jsdt.web"/>
 <fixed facet="jst.web"/>
```

７．web.xmlを修正する

プロジェクトの[Deployed Resources] -> [webapp] -> [WEB-INF] にあるweb.xmlを以下の内容に変更する

```
<?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/web-app_3_1.xsd"
         version="3.1">
</web-app>
```

８．Tomcat Serverを登録しなおす

Serversタブに「Tomcat v8.5 Server at localhost」があれば、それを消して、もう一度登録しなおす

９．プロジェクトをリフレッシュする

プロジェクトを右クリックして、Refleshして、eclipseのProjectメニューからCleanを選択する

プロジェクトを右クリックして、Maven -> Update Project を選択し、対象のプロジェクトにチェックが入っていることを確認して、OKボタンを押す

１０．（付録）途中うまくいかないことがあったら・・・

途中うまくいかないことがあったら、HelpメニューからCheck for Updatesを選んでアップデートしてから、再度うまくいかなかった箇所を実施してください。

１１．（付録）ソースコードを作成する場所

- src/main/java　・・・ ServletやJavaファイル
- src/main/resources　・・・ 設定ファイルや画像ファイル、各種データファイルなど
- src/test/java　・・・ テストクラス（JUnitテストクラス）
- src/test/resources　・・・ テストで使う設定ファイルや画像ファイル、各種データファイルなど
- Deployed Resources/webapp　・・・　JSPやHTMLファイル