The goal of this assignment is to learn interacting with databases using hibernate instead of JDBC.

Exercise 1 – Books

1. Setup Maven:

- Go to https://maven.apache.org/, download the **binary** zip / tar.gz archive
- Follow the instructions at https://maven.apache.org/install.html to terminate setting up you environment for maven.
- Test your installation by opening a terminal and running

```
mvn --version
```

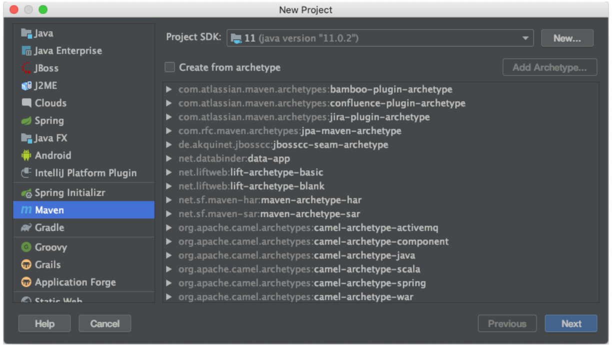
2. Maven project creation

The next step is to create a new maven project in IntelliJ / Eclipse. Each maven project requires at least to specify the following information:

- **groupId:** uniquely identifies your project across all projects: set it to your package name.
- **artifactId:** is the name of the output jar file: set it to something meaningful such as *assignment11-hibernate*.

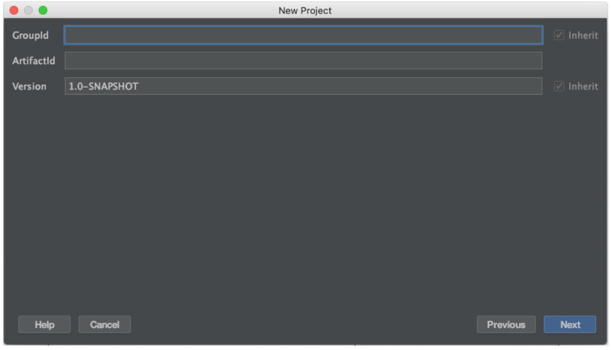
2.1 IntelliJ

Go to File → New → Project … → Maven, uncheck the "Create from archetype" checkbox and proceed with `Next` to fill *GroupId* and *ArtifactId*. Terminate the wizard with next.


New Project
Java
Java Enterprise
JBoss
J2ME
Clouds
Spring
Java FX
Android
IntelliJ Platform Plugin
Spring Initializr
Maven
Gradle
Groovy
Grails
Application Forge
Static Web
New...
Add Archetype...
com.atlassian.maven.archetypes:bamboo-plugin-archetype
com.atlassian.maven.archetypes:confluence-plugin-archetype
com.atlassian.maven.archetypes:jira-plugin-archetype
com.rfc.maven.archetypes:jpa-maven-archetype
de.akquinet.jbosscc:jbosscc-seam-archetype
net.databinder:data-app
net.liftweb:lift-archetype-basic
net.liftweb:lift-archetype-blank
net.sf.maven-har:maven-archetype-har
net.sf.maven-sar:maven-archetype-sar
org.apache.camel.archetypes:camel-archetype-activemq
org.apache.camel.archetypes:camel-archetype-component
org.apache.camel.archetypes:camel-archetype-java
org.apache.camel.archetypes:camel-archetype-scala
org.apache.camel.archetypes:camel-archetype-spring
org.apache.camel.archetypes:camel-archetype-war
Help
Cancel
Previous
Next

Project SDK: 11 (java version "11.0.2")

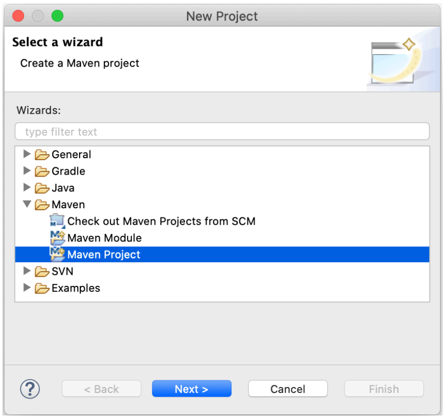
☐ Create from archetype

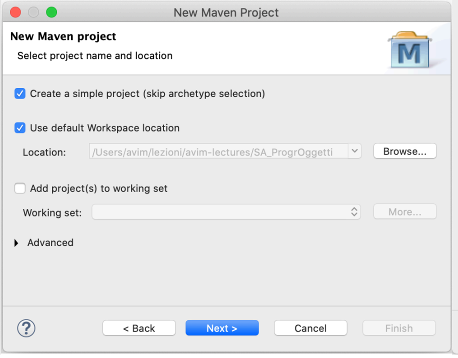


GroupId:
☑ Inherit
ArtifactId:
Version: 1.0-SNAPSHOT
☑ Inherit

2.2 Eclipse

Go to File → New Project …, Maven → Maven Project → Next. Make sure to check *"Create a simple project (skip archetype selection)"* and proceed to fill the *"Group id"* and *"Artifact id"*





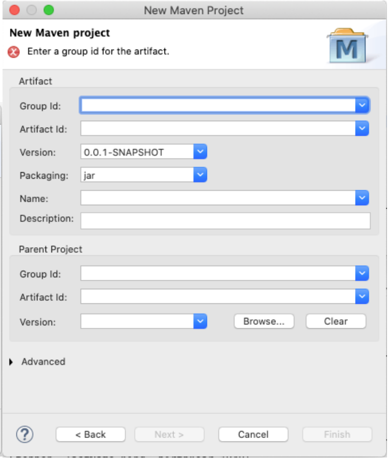
☑ Create a simple project (skip archetype selection)

☑ Use default Workspace location

Location: /Users/avim/lezioni/avim-lectures/SA_ProgrOggetti

☐ Add project(s) to working set

Working set:



Enter a group id for the artifact.

Artifact

Group Id:
Artifact Id:
Version: 0.0.1-SNAPSHOT
Packaging: jar
Name:
Description:

Parent Project

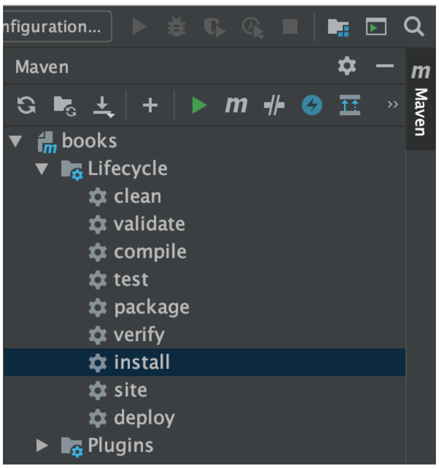
Group Id:
Artifact Id:
Version:

3. Maven project configuration

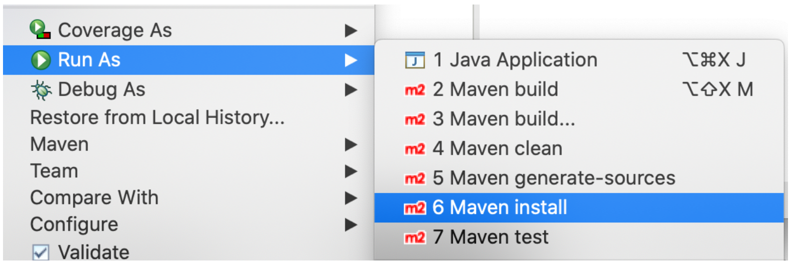
Add project dependencies, by opening the *pom.xml* file located in your project's root and add the following xml within the <project> tags.

```
<properties>
  <maven.compiler.source>17</maven.compiler.source>
  <maven.compiler.target>17</maven.compiler.target>
</properties>
<dependencies>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
    <version>5.4.10.Final</version>
  </dependency>
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>1.4.200</version>
    <scope>runtime</scope>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration><release>17</release></configuration>
    </plugin>
  </plugins>
</build>
```

Run "maven install" to download and install the required dependencies:



IntelliJ: from the right "maven" menu, select Lifecycle -> install



Eclipse: right-click on the project, go to Run As and select "Maven install"

4. Hibernate configuration

Open the *es1/persistence.xml* file and add it to the project's `src/main/resources/META-INF` folder, changing at least the highlighted values to match your implementation.

```
<persistence
  xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
                      http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd" version="2.1">
  <persistence-unit name="ch.supsi.persistence.example">
    <!-- TODO add all entity classes with fully qualified name (packagename.ClassName) -->
    <class>ch.supsi.oop.hibernate.es1.Author</class>
    <properties>
      <property name="hibernate.connection.driver_class" value="org.h2.Driver"/>
      <property name="hibernate.connection.url"
                value="jdbc:h2:~/h2-dbs/s10-hibernate-books;create=true"/>
      <property name="hibernate.connection.username" value="sa"/>
      <property name="hibernate.connection.password" value=""/>
      <property name="hibernate.show_sql" value="true"/>
      <property name="hibernate.hbm2ddl.auto" value="update"/>
    </properties>
  </persistence-unit>
</persistence>
```

5. Programming part

Add *es1/Main.java*, *es1/Author.java* and *es1/Book.java* to your previously created maven project.

1. Add the minimal required annotation to the *Author* and *Book* classes
   a. Add @Entity annotation on each class.
   b. Add @Id and @GeneratedValue(generator = "increment") to the id field.
2. Update the *persistence.xml* file and add the fully qualified (with package name) *Author* and *Book* classes to the persistence-unit.
3. Create an instance of *EntityManager* using the *Persistence.createEntityManagerFactory*. **Important**: make sure to specify the correct *persistenceUnitName* parameter when calling the *createEntityManagerFactory* method: this string must exactly match the value of the *persistence-unit name* attribute in your *persistence.xml* file.
4. Implement the empty methods using the *entityManager* (save it as a private static field of the *Main* class to access it directly from the methods):
   - *insertAuthor* : adds a new *Author* to the database.
   - *insertBook* : adds a new *Book* to the database.
   - *getAuthorById*: returns an *Author* instance using its primary key id.
   - *getBookById* : returns a *Book* instance using its primary key id.

Hints:

- Remember to use transactions.
- Use the *find* method to query the database for entities.

Exercise 2 – Address book (Optional)

Write a java program that implements a basic address book that stores its contents to a H2 database using Hibernate. The address book should allow to store contacts with at least the following information: *First name, Last name, Phone number, Mobile phone number* and *Email address.*
Add the required methods to create contacts and find them by their id.
Test your implementation by adding some contacts to the DB and then finding them by their id.