*Please note that we haven't explained full-fledged annotation based Hibernate project here. Rather we have focused on explaining how bidirectional one to one association is mapped in Hibernate.*

*To know more about annotation based basic Hibernate project please thoroughly go through project* ***' Hibernate5StandaloneWithFullJavaConfig'*** *if you are not familiar with it.*

**What this project dose ?**

This project explains how bidirectional one to one association is mapped in Hibernate by using annotations.

**A short introduction**

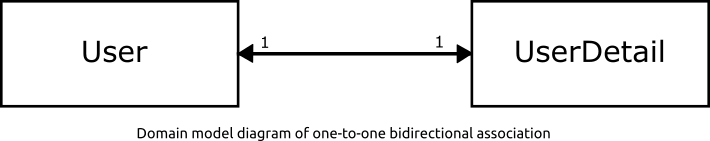
In one-to-one association, source entity has a field that references another target entity. The @OneToOne JPA annotation is used to map the source entity with the target entity.

The one-to-one association can be either unidirectional or bidirectional.

In bidirectional association, each entity (i.e. source and target) has a relationship field that refers to each other and the target entity’s table contains the foreign key. The source entity must use the mappedBy attribute to define the bidirectional one-to-one mapping.

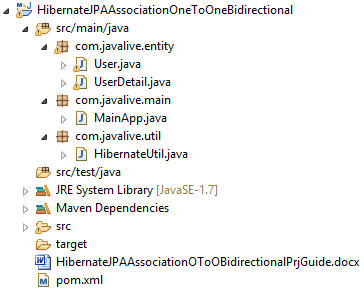
**Bidirectional one-to-one association example**

Following are the domain model and relational model diagrams of one-to-one bidirectional association.





**Project Structure**

****

**Jar dependencies**

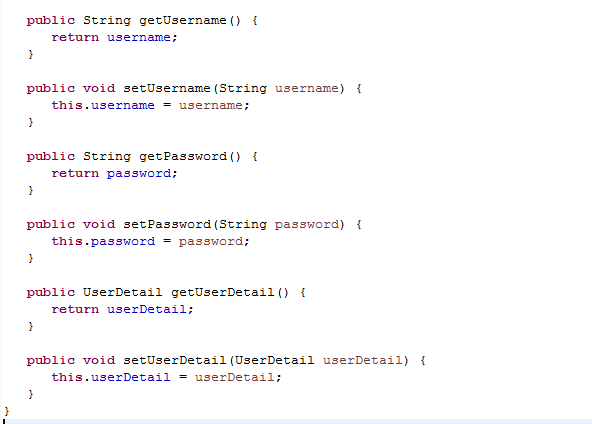
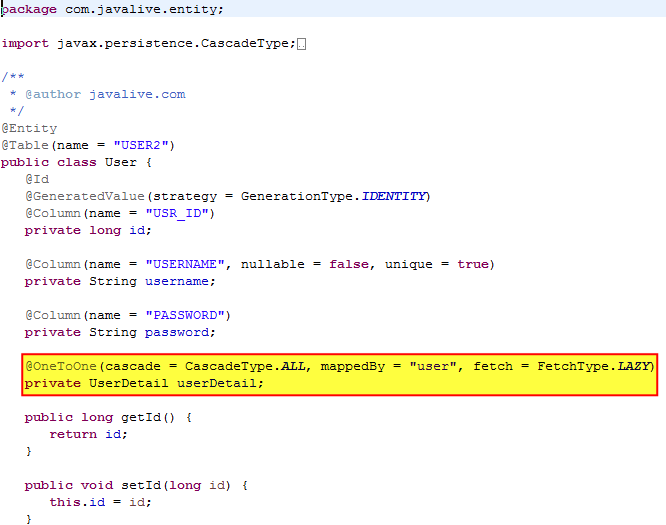
Add the following jar dependencies for Hibernate and MySQL driver in pom.xml file.



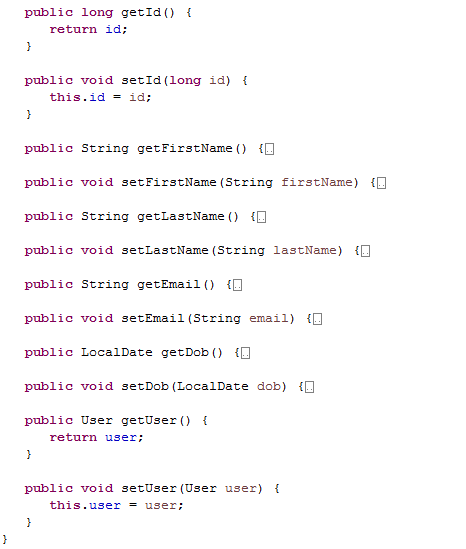
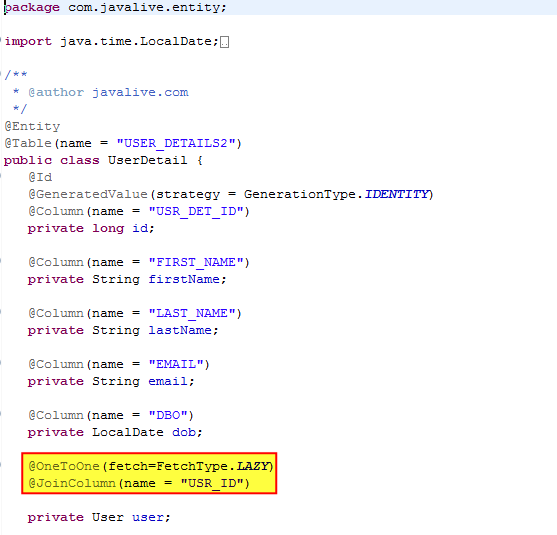
**Entity class**

Create two @Entity classes - User and UserDetail, to map with USERS and USER\_DETAILS tables respectively.

**User.java**



**UserDetail.java**

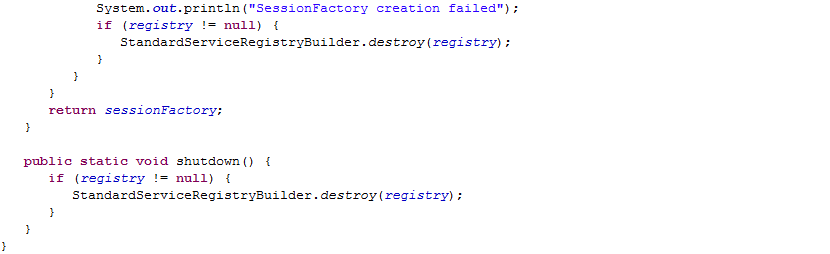
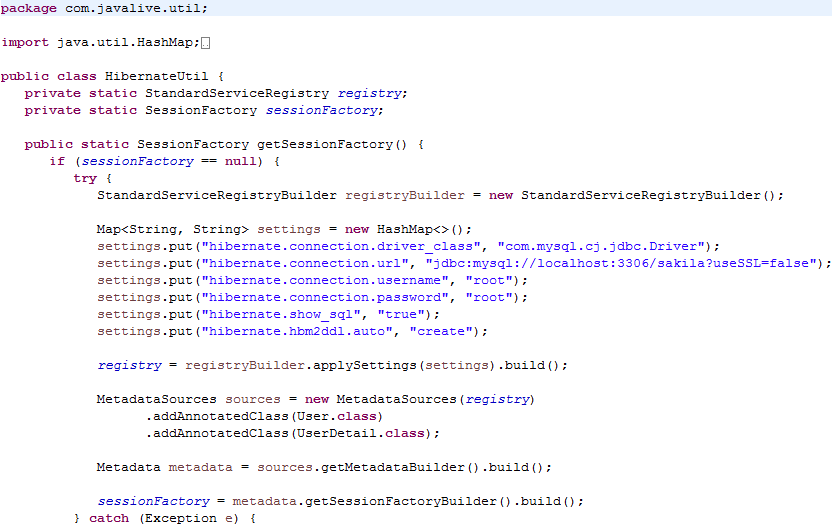
****

**Hibernate utility class**

Create a helper class HibernateUtil to bootstrap hibernate.

Map the User and UserDetails entities using the #MetadataSources.addAnnotatedClass() method.

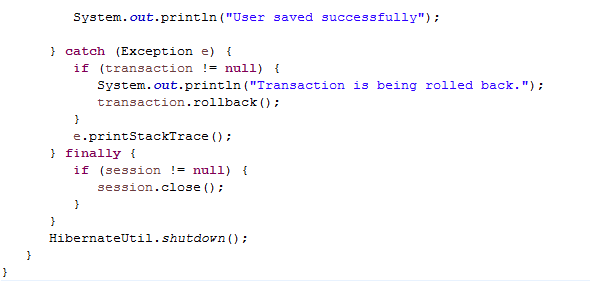
**HibernateUtil.java**



**Main class**

Create the MainApp class to run the application.

**MainApp.java**



**Output**

****

****

On executing the MainApp class, you will see the following records in USER2 and USER\_DETAILS2 tables.

