JPA Tutorial

Adriano Botti, Antonio Le Caldare, Francesco Merola, Giacomo Ponziani

Information Systems, University of Pisa

Fall, 2019

What is and Why JPA

JPA (Java Persistence API) is a Java specification for ORM (Object Relational Mapping). The purpose of ORM is to map the concepts from ObjectOriented Programming with concepts of relational databases to let the programmer to focus on the business logic instead of the translation between the two worlds. Once the mapping is done it is possble to work with instances of classes that are directly mapped to rows of table without performing any transformation. In this tutorial we focus on Hibernate which is one of the implementation of the JPA specification.

How to use JPA

Mapping may be done using *annotations* or *persistence.xml* file or both. Usually the *persistence.xml* file is used to store information about the connection with the DB while *annotations* are used actually map classes into tables.

1. Define the *persistence.xml* file;
2. Define classes using *annotations* to map them with tables;
3. Use *JPQL* to perform *CRUD* operations;

Examples Definition

All the following examples come from the application developed in
Task 1. This application handles university exams creation,
registration, deregistration and validation. For clarity the
class-diagram is reported.

–INSERT CLASS DIAGRAM–

JPA Annotations

- *@Entity*, specifies the following class is an entity to be mapped to one or more tables in the DB.
- *@Table*, specifies the primary table for the annotated entity. The attribute *name* indicates the referenced table in the DB.
- *@Column*, specifies the mapped column for a persistent property or field. The attribute *name* indicates the name of the corresponding table column. It may be neglected when names of the attribute and of the columns are the same. Note that if the names are not equal Hinerbate will add a column to the table with name equals to the value if the *name*.

- @Entity
- @Table
- @Column
- @Id
- @GeneratedValue
- @Embeddable
- @EmbeddedID
- @MapsId
- @JoinColumn
- @JoinColumns
- @ManyToOne
- @OneToMany
- @ManyToMany
- @OneToOne