

KIV/PIA - JPA Basics

This lab covers:

- basic set-up of JPA persistence context backed by Hibernate provider;
- elementary mapping of entity classes to database tables

Java Persistence API

JPA is specification of Java interface for **Object-Relational-Mapping (ORM)**. For details consult the [Oracle Documentation](#).

The specification provides three main areas:

- Entity Mapping Interface - set of annotations for describing how the mapping should be done
- API for entity management
- Query interface - **Java Persistence Query Language (JPQL)** and **Criteria API**

Dependencies

There are two mandatory dependencies in the *pom.xml* file:

- hibernate-jpa-2.1-api - artifact with JPA interfaces (**javax.persistence** package)
- hibernate-entitymanager - entity manager implementation from [Hibernate ORM Project](#).

One optional dependency is [C3P0 Connection Pooling Library](#). Remember the struggle with connection management from the previous lab?

Configuration

In order to use JPA in an application, a so-called **Persistence Unit** must be defined. **Persistence Unit** defines:

- set of entities managed by an entity manager
- data store in which the entities are persisted
- configuration of the entity manager and additional mechanisms

Persistence Units are defined in *META-INF/persistence.xml* file. In a Maven project the location is *src/main/resources/META-INF/persistence.xml*.

Entity Mapping

This section covers field and association mapping of entities.

Class that represents an entity is annotated with `@Entity` annotation. You can use `@Table` to add own specification of the table the entity is mapped to.

```
@Entity
@Table(name="my_entity_table")
public class MyEntity {}
```

Elementary Fields

Primitive data types (such as int, double) and enums are mapped automatically as well as certain common classes such as Date, String. Yet in certain cases it might be necessary to specify additional information about the mapping. For this the following annotations are used:

- `@Basic` - allows to specify whether the property is fetched lazily or whether it is optional. Use of the annotation is optional, if not provided, default settings are used (fetch = EAGER, optional = true)
- `@Temporal` - used to specify type of information persisted in Date attribute - DATE, TIME, TIMESTAMP.
- `@Enumerated` - used to decide whether enum type values are stored as ordinal number or string representation of the value.
- `@Column` - can be used in conjunction with any of the previously mentioned annotations to provide additional information about the mapped column - is it nullable, updatable, etc.

License

Base of the JPA setup has been created by Karel Zibar during one of the courses at the University.

This work is licensed under the Creative Commons license BY-NC-SA.



Exercises for Programming of Web Applications by [Jakub Danek](#) is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#).