

## Goal

Became familiar with the JPA, be able to define entities and do basic operations with them. Understand to entity lifecycle.

## First JPA project

### Preparing the test project

1. Create new maven project with suitable name.
2. Add dependency to hibernate. Group id is org.hibernate, artefact id is hibernate-entitymanager, use version 4.1.4.Final.
3. Add dependency to derby client. Group id is org.apache.derby, artefact id is derbyclient, use version 10.7.1.1.
4. Create file persistence.xml with given content (see end of this section) and place it to correct place. Do not use NetBeans *New persistence unit wizard!*
5. Create Derby database with name TestDB.
6. Change persistence.xml to contain actual jdbc url, user name and password of created database.
7. Modify main method to create new instance of EntityManagerFactory for our persistence unit.
8. Try to run main method.

#### persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
    xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/
persistence http://java.sun.com/xml/ns/persistence/
persistence_1_0.xsd">
  <persistence-unit name="TestPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="hibernate.connection.username" value="app"/>
<property name="hibernate.connection.driver_class"
value="org.apache.derby.jdbc.ClientDriver"/>
      <property name="hibernate.connection.password" value="app"/>
      <property name="hibernate.connection.url" value="jdbc:derby://
localhost:1527/HelloDB"/>
<property name="hibernate.cache.provider_class"
value="org.hibernate.cache.NoCacheProvider"/>
      <property name="hibernate.hbm2ddl.auto" value="update"/>
    </properties>
  </persistence-unit>
</persistence>
```

## Create first entity

1. Create entity Person with attributes firstName, lastName and address.
2. Try to run main method.
3. Check in database if appropriate table was created and look at its structure.
4. Update entity to:
  - Use table SimplePerson
  - Use column givenName for storing attribute firstName
  - Change column size to 50 for attribute lastName
5. Change table generation strategy to *Drop and Create* in persistence.xml. Try to run the main method and check if the database structure has been correctly changed.

## Manipulation with entity

1. Modify main method to do these transactions in single persistence context:
  - a. Create new instance of person with name Paul Smith and address Botanicka 68a, Brno. Store this instance into database and print id of newly created entity to standard output.
  - b. Change address of Paul Smith to Chvalovice 12.
  - c. Change name of Paul Smith to Paul Brown.
2. Create method dump(...) that prints list of all person entities in database to standard output. Modify main method to call dump(...) after each transaction.
3. Try to execute main method and check if it behaves well.
4. Change the main method to do transaction changing Pauls address in different persistence context.
5. Try to execute main method and check if it behaves well.
6. Add transaction that deletes Paul from the database.\