

- TD 4 -

Gérer les personnes avec JPA

In the application greeting people we used a persistence approach based on JDBC. We want now to use a more robust approach based on JPA. We use the same resources as for the other applications (`jdbc/gla` in the pool `GLAPool`) or the default data source (`JNDI=java:comp/DefaultDataSource`). Note that in an application server we generally use a Persistence Unit of type JTA. *The PU can be created with NetBeans (category Persistence-PersistenceUnit); in the development phase we could use a Table Creation Strategy of type Drop and Create.*

1. A very simple case

As a first step, we must modify the application (or better yet, develop a new one) to persist persons without using an explicit connection to the database. We should mainly annotate the `Person` class previously provided to make it an entity. As with the other applications, it's better to separate the tier devoted to the persistence from the business logic (using, for example, a library).

For the moment, we can see the list of registered persons using the tools offered by the IDE (or the application developed previously) - we will eventually develop the features allowing us to list all the persons in the application.

2. The status of persons

You can now add more information with respect to registered persons: date of birth and status. The status concerns, for the moment, the full title (e.g., Misses, Mister, Doctor, ...) and the acronym (e.g. Mrs., Mr., Dr., ...) but more information could be added later. It is therefore necessary to add an entity `Status`, complete the `Person` entity (with corresponding fields and annotations) and develop the web page as well as the managed beans needed to persist this kind of persons. For example, you can use a simple form to persist the corresponding person with its status. Don't forget that you should persist entities in the correct order (or propagate correctly the operations). Check the database to verify that the data was recorded.

Registration Form

Firstname:

Lastname:

Birthdate:

Title:

Full title:

3. The addresses

A person can have several addresses (characterized by the street number, the street code, the city, the country) that can be entered with the following (simplified) form:

We may consider associating an address to a person only if a minimum amount of information is available (e.g., provide at least the city or the postal code).

Propose an entity `Address` and complete the other entities to correctly persist everything.

Civil

Firstname:

Lastname:

Birthdate:

Title:

Full title:

Address 1

Code:

Town:

Address 2

Code:

Town:

4. Search by id

The application must be able to find persisted persons. First, the only search criterion is the identifier of the person. So we want to offer an interface for searching for people:

Id:

You can use outputs in the console to check the fetch type (eager or lazy) for the fields and the associations of the corresponding entity.

All interfaces are given as an indication and can be easily improved.