

# Spring Boot

## Workshop N°7

### Online Store Application (REST API CRUD Operations) With Spring DATA

#### Objectives:

- Implement a business layer with service interfaces and service implementations to handle the core functionalities of the online store, such as creating, reading, updating, and deleting products.
- Learn about the data access layer and its responsibility in interacting with the underlying database using Spring Data JPA.
- Integrate the API with a MySQL database to persist and retrieve data.
- Gain hands-on experience in developing RESTful APIs using Spring Boot, Spring Data JPA, and MySQL.

#### Description:

In this workshop, you will continue working on the existing "**onlinestorerestapi**" project and enhance it by implementing a business layer with service interfaces and service implementations. We will focus on separating the business logic from the presentation (web) layer and integrating the API with a MySQL database using Spring Data JPA.

#### Tasks:

Follow the instructions below to set up the project and implement the required functionality.

##### 1. Data Access Layer with JPA:

- Set up the database connection configuration using Spring Data JPA and MySQL.
- Create the necessary repositories and entities to perform CRUD operations on the products.

##### 2. Business Layer:

- Create the required packages and classes for the business layer.
- Define a service interface, such as "ProductService", that declares the methods for handling CRUD operations on products.
- Implement the service interface with a service implementation class, such as "ProductServiceImpl", to provide the business logic for the operations.
- Inject the necessary dependencies, such as repositories, into the service implementation.

##### 3. Controller Integration:

- Modify the "ProductController" class to use the service interface instead the static list of products
- Inject the service implementation into the controller using dependency injection.
- Delegate the business logic operations to the service implementation in the controller endpoints.

##### 4. Test the Endpoints:

- Use tools like Talend API Tester to send requests to the implemented endpoints and verify the functionality of the CRUD operations.

- Ensure that the responses include the appropriate HTTP status codes and JSON data.