Spring Data JPA

3rd semester @ Erhvervsakademi København

What is Spring Data JPA?

- **Spring Data JPA** is a part of the Spring Framework that simplifies database access using **JPA** (Jakarta Persistence API) and **Hibernate**.
- It provides a powerful way to interact with databases using Java objects.
- It **abstracts the boilerplate code** required for database operations, allowing developers to focus on business logic.
- It integrates seamlessly with Spring Boot, making it easy to set up and use.

What is JPA?

- **JPA** defines a standard for object-relational mapping (ORM) in Java.
- JPA contains annotations and interfaces to map Java objects to database tables = **No implementation**.
- It uses **Hibernate** as the default implementation in Spring Boot.
- **Hibernate** is a popular ORM framework that implements the JPA specification.

What is an ORM?

- **ORM** = **O**bject-**R**elational **M**apping
- A technique to convert data between incompatible type systems in object-oriented programming languages.
- Maps Java objects to rows in a table and vice versa.
- Allows developers to work with Java objects instead of SQL queries.

How does Spring Data JPA work?

- **Repositories**: Spring Data JPA uses repositories to perform database operations.
- **Annotations**: It uses annotations to define entities, relationships, and queries.
- **Query Methods**: You can define query methods in repositories without writing SQL.
- **Automatic Implementation**: Spring Data JPA automatically generates the implementation of repository interfaces at runtime.

Example: Entity

```
@Entity // Maps this class to a database table
public class Course {
    @Id // Primary key
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String title;
    private String description;
    // Getters and Setters
}
```

Example: Repository

We create a repository interface to perform CRUD operations on the `Course` entity:

```
@Repository // Indicates that this interface is a repository
public interface CourseRepository extends JpaRepository<Course, Long> {
}
```

Example: Using the Repository

```
@Service // Indicates that this class is a service
public class CourseService {

    private final CourseRepository courseRepository;

    // Constructor injection - Injects the CourseRepository
    public CourseService(CourseRepository courseRepository) {
        this.courseRepository = courseRepository;
    }

    public List<Course> getAllCourses() {
        return courseRepository.findAll(); // Fetches all courses from the database
    }
}
```

Exercise: Adding Spring Data JPA