

Bài tập bắt buộc của Tuần **XX** GV sẽ công bố cuối giờ thực hành. Bài tập bắt buộc sẽ push lên Github với Repository name: Tuan07

*Sinh viên nộp sai yêu cầu bài tập Tuần nào xem như không nộp bài tập Tuần đó.*

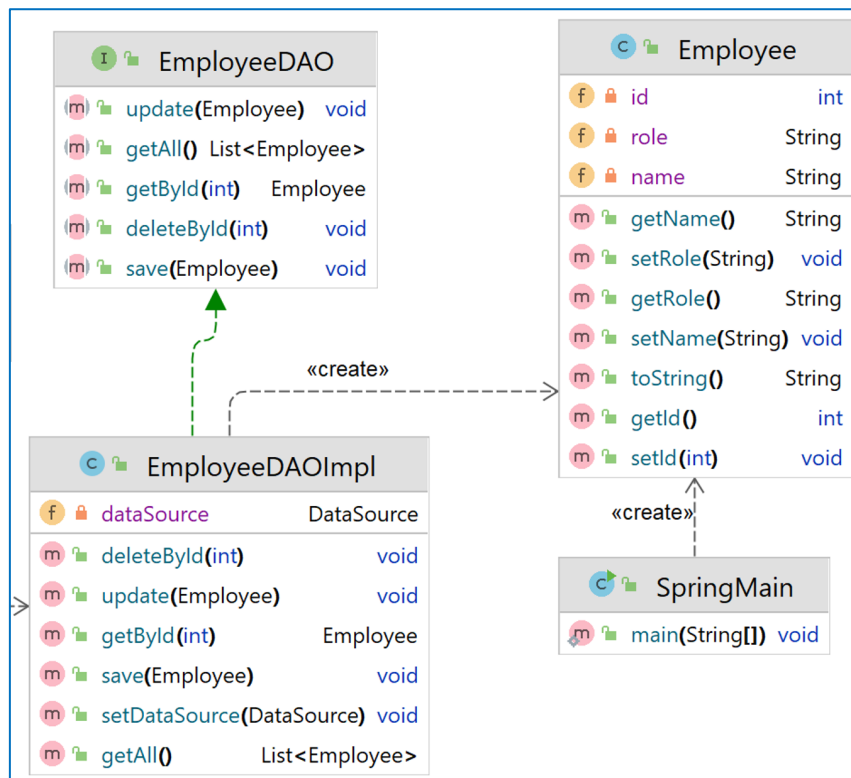
## BÀI TẬP TUẦN 7 MÔN LẬP TRÌNH WEB NÂNG CAO (JAVA)

### 1. Chương 6: SpringDataCommon

#### Mục tiêu:

- Hiểu và hiện thực được JDBC API
- Hiểu và hiện thực được Spring Data JPA

Tham khảo Slide bài giảng, làm các bài tập theo hình bên dưới:



#### Bài tập 1: Sử dụng JDBC API

Dependencies:

- JDBC API

- MariaDB Driver

## 1. Dùng XML configuration

*src/main/resources/databaseConfig.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:util="http://www.springframework.org/schema/util" xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.springframework.org/schema/util
           http://www.springframework.org/schema/util/spring-util.xsd">
  <!-- bean definitions here -->
  <bean id="dataSource"
        class="org.springframework.jdbc.datasource.DriverManagerDataSource" scope="singleton">
    <property name="url" value="jdbc:mariadb://localhost:3306/employees"></property>
    <property name="driverClassName" value="org.mariadb.jdbc.Driver"></property>
    <property name="username" value="root"></property>
    <property name="password" value="123456"></property>
  </bean>
</beans>
```

*SpringPureJdbcxmlConfigApplication.java*

```
@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})
public class SpringPureJdbcxmlConfigApplication {

    @SuppressWarnings("resource")
    public static void main(String[] args) throws SQLException {
        SpringApplication.run(SpringPureJdbcxmlConfigApplication.class, args);

        ApplicationContext context = new ClassPathXmlApplicationContext("databaseConfig.xml");
        DataSource dataSource = context.getBean("dataSource", DataSource.class);
        System.out.println(dataSource.getConnection());
    }
}
```

## 2. Dùng Annotation và Java based

*src/main/resources/database.properties*

```
*database.properties X
1 # Setting mariaDB
2 app.datasource.driver=org.mariadb.jdbc.Driver
3 app.datasource.url=jdbc:mariadb://localhost:3306/employees
4 app.datasource.username=root
5 app.datasource.password=123456
```

```
AppConfig.java X
1 package iuh.fit.se;
2
3 import javax.sql.DataSource;
10
11 @Configuration
12 @PropertySource("classpath:database.properties")
13 public class AppConfig {
14     @Value("${app.datasource.url}")
15     private String url;
16
17     @Value("${app.datasource.username}")
18     private String userName;
19
20     @Value("${app.datasource.password}")
21     private String password;
22
23     @Value("${app.datasource.driver}")
24     private String driver;
25
26     @Bean(name = {"dataSource"})
27     DataSource getDataSource() {
28         DriverManagerDataSource driverManagerDataSource = new DriverManagerDataSource();
29         driverManagerDataSource.setUrl(url);
30         driverManagerDataSource.setUsername(userName);
31         driverManagerDataSource.setPassword(password);
32         driverManagerDataSource.setDriverClassName(driver);
33
34         return driverManagerDataSource;
35     }
36 }
```

```
AppConfig.java SpringPureJdbAnnotationConfigApplication.java X
1 package iuh.fit.se;
2
3 import java.sql.SQLException;
12
13 @SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})
14 public class SpringPureJdbAnnotationConfigApplication {
15
16     @SuppressWarnings("resource")
17     public static void main(String[] args) throws SQLException {
18         SpringApplication.run(SpringPureJdbAnnotationConfigApplication.class, args);
19         ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);
20         DataSource dataSource = context.getBean("dataSource", DataSource.class);
21         System.out.println(dataSource.getConnection());
22     }
23 }
```

### 3. Dùng Spring Boot

```
application.properties X
1 spring.application.name=SpringPureJDBAutoConfig
2
3 # Setting mariaDB
4 spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
5 spring.datasource.url=jdbc:mariadb://localhost:3306/employees_jdbc
6 spring.datasource.username=root
7 spring.datasource.password=123456

@SpringBootApplication(scanBasePackages = {"iuh.fit.se.dao", "iuh.fit.se.daoImpl"})
public class SpringPureJdbAutoConfigApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringPureJdbAutoConfigApplication.class, args);
    }

    @Bean
    CommandLineRunner runner(EmployeeDAO employeeDAO) {
        return args -> {
            Employee employee = new Employee("Nguyen Van A", "Super Admin");
            employeeDAO.save(employee);

            List<Employee> employees = employeeDAO.getAll();
            employees.forEach(System.out::print);

            Employee employee2 = employeeDAO.getById(1);
            System.out.println(employee2);

            Employee employeeDirectMapper = employeeDAO.getByIdDirectMapper(1);
            System.out.println(employeeDirectMapper);
        };
    }
}
```

## Bài tập 2: Sử dụng Spring Data JPA

Dependencies:

- Spring Data JPA
- MariaDB Driver