

spring-jdbc 설정

1. mvnrepository에서 디펜던시 추가해 오기

- 1) spring jdbc 검색
- 2) 4.1.4 버전 선택
- 3) pom.xml 에 아래 디펜던시 추가

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>4.1.4.RELEASE</version>
</dependency>
```

2. root-context.xml 에 빈 추가

```
<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
    <property name="url" value="jdbc:oracle:thin:@127.0.0.1:1521:xe" />
    <property name="username" value="javaweb0224" />
    <property name="password" value="1111" />
</bean>
<bean id="template" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="ds" />
</bean>
```

3. common 패키지에 Constant 클래스 추가

```
import org.springframework.jdbc.core.JdbcTemplate;

public class Constant {
    public static JdbcTemplate template;
}
```

4. Controller에서 @Autowired 작업으로 Constant 클래스의 JdbcTemplate template을 생성해 둔다.

```
// Constant.java 의 JdbcTemplate template 생성
private JdbcTemplate template; // 이건 Constant.java 의 것이 아니고, ContactController 의 것이다.
```

```
@Autowired
public void abcdefg(JdbcTemplate template) { // @Autowired 에 의해서 자동으로 생성되는 bean은 매개변수 JdbcTemplate template 이다.
    this.template = template;
    Constant.template = this.template;
}
```

5. DAO에서 Constant 클래스의 JdbcTemplate template을 가져다 사용한다.

```
public class DAO {
    private JdbcTemplate template;
    public ContactDao() {
        this.template = Constant.template;
    }
    ....
}
```