KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

스프링과 MySQL Database 연동

[KB] IT's Your Life

KB 국민은행

스프링과 MySQL Database 연동

다음과 같이 데이터베이스와 사용자를 준비하세요.

```
create database scoula_db;

create user 'scoula'@'%' identified by '1234';

grant all privileges on scoula_db.* to 'scoula'@'%';
```

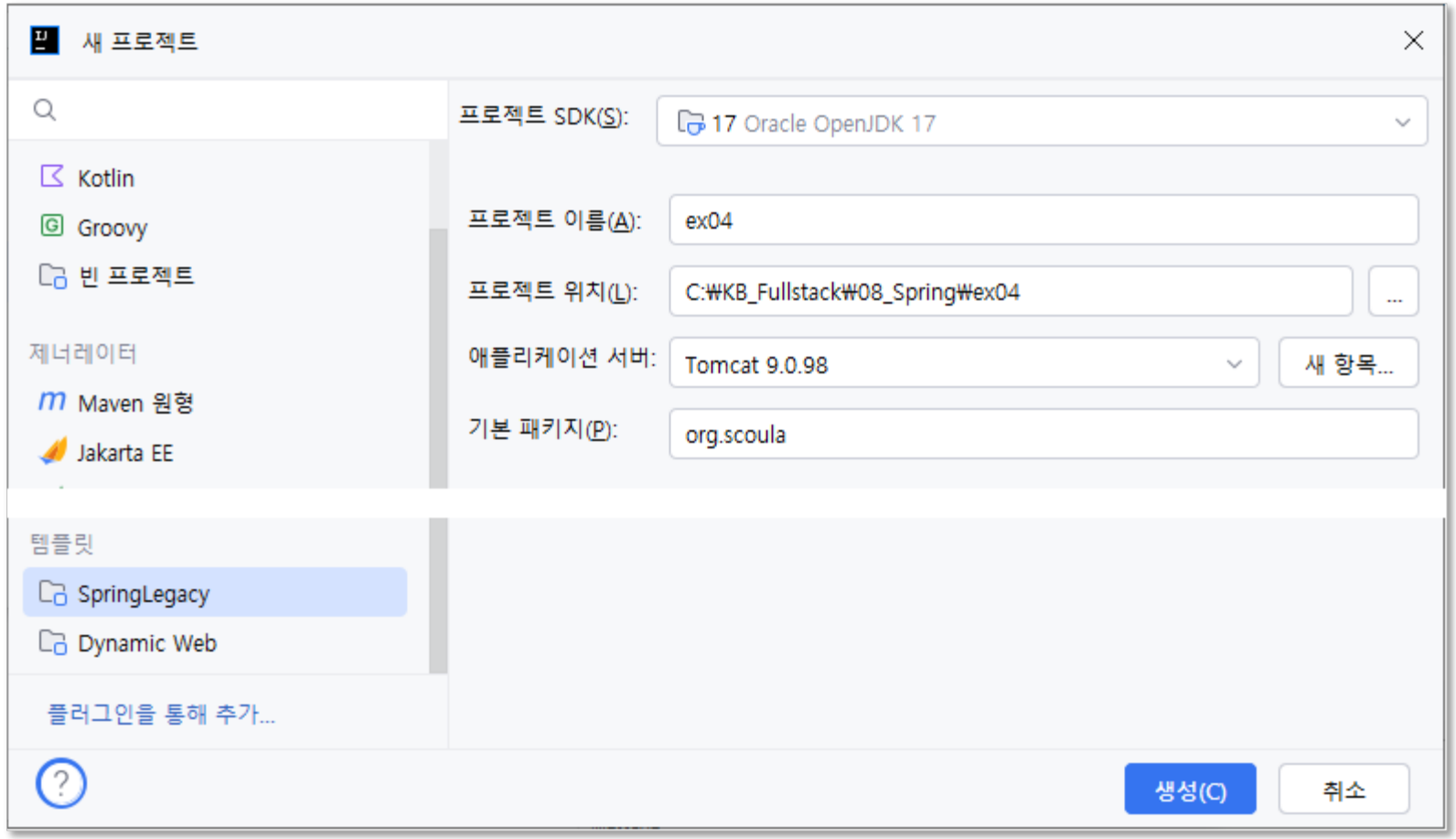
ex04 프로젝트를 생성하세요.

- **프로젝트 템플릿**: SpringLegacy
- 의존성 설정에 MySQL JDBC 드라이버 추가

스프링과 MySQL Database 연동

프로젝트 만들기

- Templates: SpringLegacy
- Project name: ex04



스프링과 MySQL Database 연동

settings.gradle

```
rootProject.name = 'ex04'
```

- Enable Annotation Processor 활성화

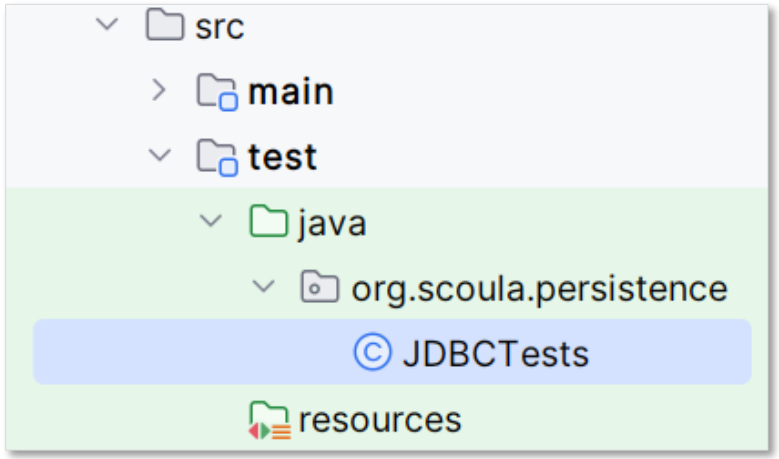
build.gradle

```
// 데이터베이스
implementation 'com.mysql:mysql-connector-j:8.1.0'
```

→ gradle sync

스프링과 MySQL Database 연동

- **JDBC 드라이버를 통해 DB 연결 테스트 코드를 작성하세요.**
  - src/test/java
    - org.scoula.persistence
      - JDBCTests.java



  - 데이터베이스명: scoula_db
  - 사용자명: scoula

test :: JDBCTest.java

```
package org.scoula.persistence;

import lombok.extern.log4j.Log4j2;
import org.junit.jupiter.api.Test;

import java.sql.Connection;
import java.sql.DriverManager;

import static org.junit.jupiter.api.Assertions.fail;

@Log4j2
public class JDBCTest {
    static {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
```

test :: JDBCTests.java

```
        @Test
        @DisplayName("JDBC 드라이버 연결이 된다.")
        public void testConnection() {
                String url = "jdbc:mysql://localhost:3306/scoula_db";
                try(Connection con = DriverManager.getConnection(url, "scoula", "1234")) {
                        log.info(con);
                } catch(Exception e) {
                        fail(e.getMessage());
                }
        }

}
```

INFO : org.scoula.persistence.JDBCTests - com.mysql.cj.jdbc.ConnectionImpl@13579834

스프링과 MySQL Database 연동

- HikariCP DataSource를 위한 의존성을 추가하세요.

스프링과 MySQL Database 연동

build.gradle

```
// 데이터베이스
implementation 'com.mysql:mysql-connector-j:8.1.0'
implementation 'com.zaxxer:HikariCP:2.7.4'
```

→ gradle sync

스프링과 MySQL Database 연동

- resources/application.properties에 데이터소스 연결에 필요한 설정을 추가하세요.

스프링과 MySQL Database 연동

resources/application.properties

```
jdbc.driver=com.mysql.cj.jdbc.Driver
jdbc.url=jdbc:mysql://127.0.0.1:3306/scoula_db
jdbc.username=scoula
jdbc.password=1234
```

스프링과 MySQL Database 연동

- RootConfig.java에 application.properties 정보를 이용하여 DataSource 빈을 등록하는 코드를 추가하세요.

RootConfig.java

```
package org.scoula.config;

import javax.sql.DataSource;

@Configuration
@PropertySource({"classpath:/application.properties"})
public class RootConfig {
    @Value("${jdbc.driver}") String driver;
    @Value("${jdbc.url}") String url;
    @Value("${jdbc.username}") String username;
    @Value("${jdbc.password}") String password;

    @Bean
    public DataSource dataSource() {
        HikariConfig config = new HikariConfig();

        config.setDriverClassName(driver);
        config.setJdbcUrl(url);
        config.setUsername(username);
        config.setPassword(password);

        HikariDataSource dataSource = new HikariDataSource(config);
        return dataSource;
    }
}
```

스프링과 MySQL Database 연동

- DataSourceTest.java를 테스트에 추가하고, RootConfig의 DataSource 빈 생성을 테스트하세요.

test :: config.DataSourceTest.java

```
@ExtendWith(SpringExtension.class)
@ContextConfiguration(classes= {RootConfig.class})
@Log4j2
class RootConfigTest {

    @Autowired
    private DataSource dataSource;

    @Test
    @DisplayName("DataSource 연결이 된다.")
    public void dataSource() throws SQLException {
        try(Connection con = dataSource.getConnection()){
            log.info("DataSource 준비 완료");
            log.info(con);
        }
    }
}
```

```
INFO : org.scoula.config.RootConfigTest - DataSource 준비 완료
INFO : org.scoula.config.RootConfigTest - HikariProxyConnection@270734602 wrapping com.mysql.cj.jdbc.ConnectionImpl@
3ece1e79
```