KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

JDBC 프로그래밍

[KB] IT's Your Life

KB 국민은행

- **다음과 같이 데이터베이스를 준비하세요.**
  - 데이터베이스명: jdbc_ex
  - 사용자: scoula
  - 비밀번호: 1234
  - jdbc_ex 데이터베이스에 대한 모든 사용권한 부여

데이터베이스 준비

```
CREATE DATABASE jdbc_ex;
```

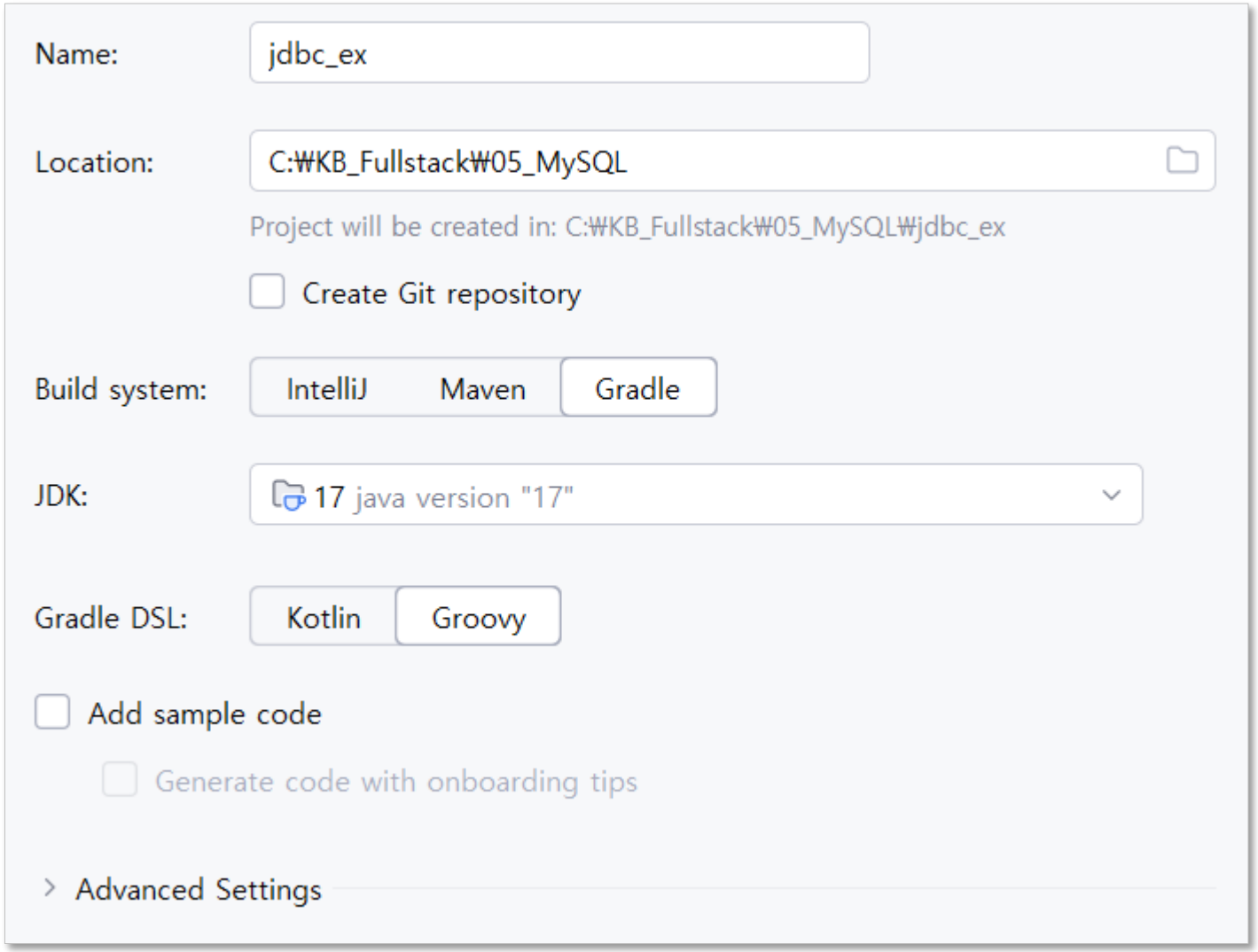
사용자 준비

```
CREATE USER scoula'@'%' IDENTIFIED BY '1234';
GRANT ALL PRIVILEGES ON jdbc_ex.* TO 'scoula'@'%';
FLUSH PRIVILEGES;
```

- **자바 프로젝트를 생성하세요.**
  - Name: jdbc_ex
  - Build system: gradle
  - 의존성
    - lombok
    - mysql-connector-j
  - Annotation Processing 활성화

프로젝트 생성

- Name: jdbc_ex
- Build System : gradle
- Group Id: org.scoula



build.gradle

```
...
dependencies {
    implementation 'com.mysql:mysql-connector-j:8.3.0'
    compileOnly 'org.projectlombok:lombok:1.18.30'
    annotationProcessor 'org.projectlombok:lombok:1.18.30'

    testCompileOnly 'org.projectlombok:lombok:1.18.30'
    testAnnotationProcessor 'org.projectlombok:lombok:1.18.30'

    testImplementation platform('org.junit:junit-bom:5.9.1')
    testImplementation 'org.junit.jupiter:junit-jupiter'
}
...
```

- 수정 후 Sync 실행

- 프로젝트 설정
  - Annotation Processing 활성화

Intellij에서 다음과 같은 users 테이블을 정의하세요.

컬럼명	데이터타입	제약조건	설명
id	varchar(12)	PK	사용자 id
password	varchar(12)	NN	비밀번호
name	varchar(30)	NN	이름
role	varchar(6)	NN	역할

다음과 같은 테스트 데이터를 추가하세요.

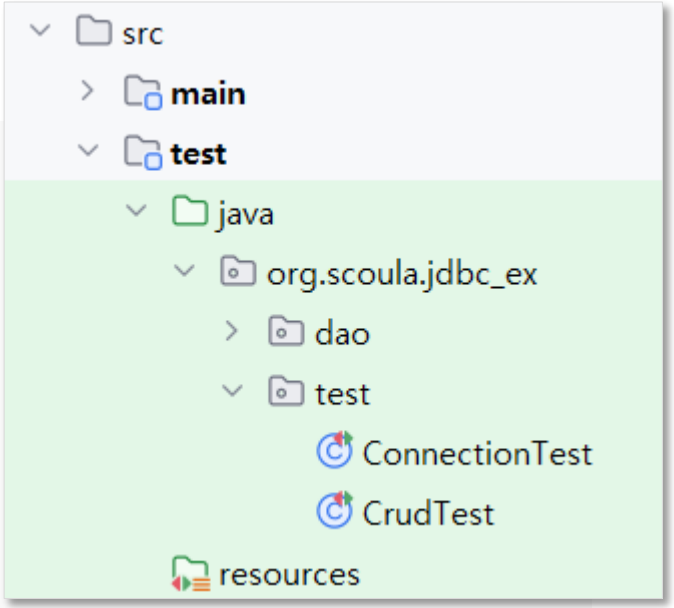
id	password	name	role
guest	guest123	방문자	USER
admin	admin123	관리자	ADMIN
member	member1234	일반회원	USER

데이터 준비

```
use jdbc_ex;

CREATE TABLE USERS (
  ID VARCHAR(12) NOT NULL PRIMARY KEY,
  PASSWORD VARCHAR(12) NOT NULL,
  NAME VARCHAR(30) NOT NULL,
  ROLE VARCHAR(6) NOT NULL
);

INSERT INTO USERS(ID, PASSWORD, NAME, ROLE)
VALUES('guest', 'guest123', '방문자', 'USER');

INSERT INTO USERS(ID, PASSWORD, NAME, ROLE)
VALUES('admin', 'admin123', '관리자', 'ADMIN');

INSERT INTO USERS(ID, PASSWORD, NAME, ROLE)
VALUES('member', 'member123', '일반회원', 'USER');

SELECT * FROM USERS;
```

- **JUnit5를 이용하여 데이터베이스에 접속하는 JDBC 코드를 테스트하세요.**
  - test 폴더에서 작업함
  - 패키지명: org.scoula.jdbc_ex.test
  - 테스트 클래스명: ConnectionTest
    - 테스트 메소드: testConnection()

ConnectionTest.java




```
package org.scoula.jdbc_ex.test;

import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;
import org.scoula.jdbc_ex.common.JDBCUtil;

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

public class ConnectionTest {

    @Test
    @DisplayName("jdbc_ex 데이터베이스에 접속한다.")
    public void testConnection() throws SQLException, ClassNotFoundException {
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url = "jdbc:mysql://127.0.0.1:3306/jdbc_ex";
        String id = "scoula";
        String password = "1234";
        Connection conn = DriverManager.getConnection(url, id, password);
        System.out.println("DB 연결 성공");
        conn.close();
    }
}
```

```
DB 연결 성공
```

- **resource 폴더에 application.properties 파일을 생성하고 다음 내용을 작성하세요.**

속성명	속성값
driver	com.mysql.cj.jdbc.Driver
url	jdbc:mysql://127.0.0.1:3306/jdbc_ex
id	scoula
password	1234

resource::/application.properties

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

- **org.scoula.jdbc_ex.common 패지키에 JDBCUtil 클래스를 작성하세요.**
  - Properties 클래스를 이용하여 application.properties 파일을 로드함
  - 데이터베이스 접속에 필요한 정보를 추출함
  - jdbc_ex 데이터베이스에 접속하여 Connection 객체를 보관함
  - getConnection() 스태틱 메서드를 정의함
    - 외부에서 Connection 객체를 추출하기 위함
  - 데이터베이스 접속을 끊는 close() 메서드를 정의함

JDBCUtil.java

```
package org.scoula.jdbc_ex.common;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class JDBCUtil {
  static Connection conn = null;

  public static Connection getConnection() {

    if(conn != null)
      return conn;

    try {
      Properties properties = new Properties();
      properties.load(JDBCUtil.class.getResourceAsStream("/application.properties"));
      String driver = properties.getProperty("driver");
      String url = properties.getProperty("url");
      String id = properties.getProperty("id");
      String password = properties.getProperty("password");

      Class.forName(driver);
      conn = DriverManager.getConnection(url, id, password);
    } catch (Exception e) {
      e.printStackTrace();
    }

    return conn;
  }
```

JDBCUtil.java

```
    public static void close() {
     try {
        if (conn != null) {
           conn.close();
           conn = null;
        }
     } catch (SQLException e) {
        e.printStackTrace();
     }
  }

}
```

- ConnectionTest 클래스의 testConnection2 메서드에서 JDBCUtil 클래스의 기능을 테스트하세요.

ConnectionTest.java

```
package org.scoula.jdbc_ex.test;
...

public class ConnectionTest {

    ...
    @Test
    @DisplayName("jdbc_ex에 접속한다. (자동 닫기)")
    public void testConnection2() throws SQLException {
        try(Connection conn = JDBCUtil.getConnection()) {
            System.out.println("DB 연결 성공");
        }
    }
}
```