

Spring Framework

2024
분산프로그래밍2

배수연

01

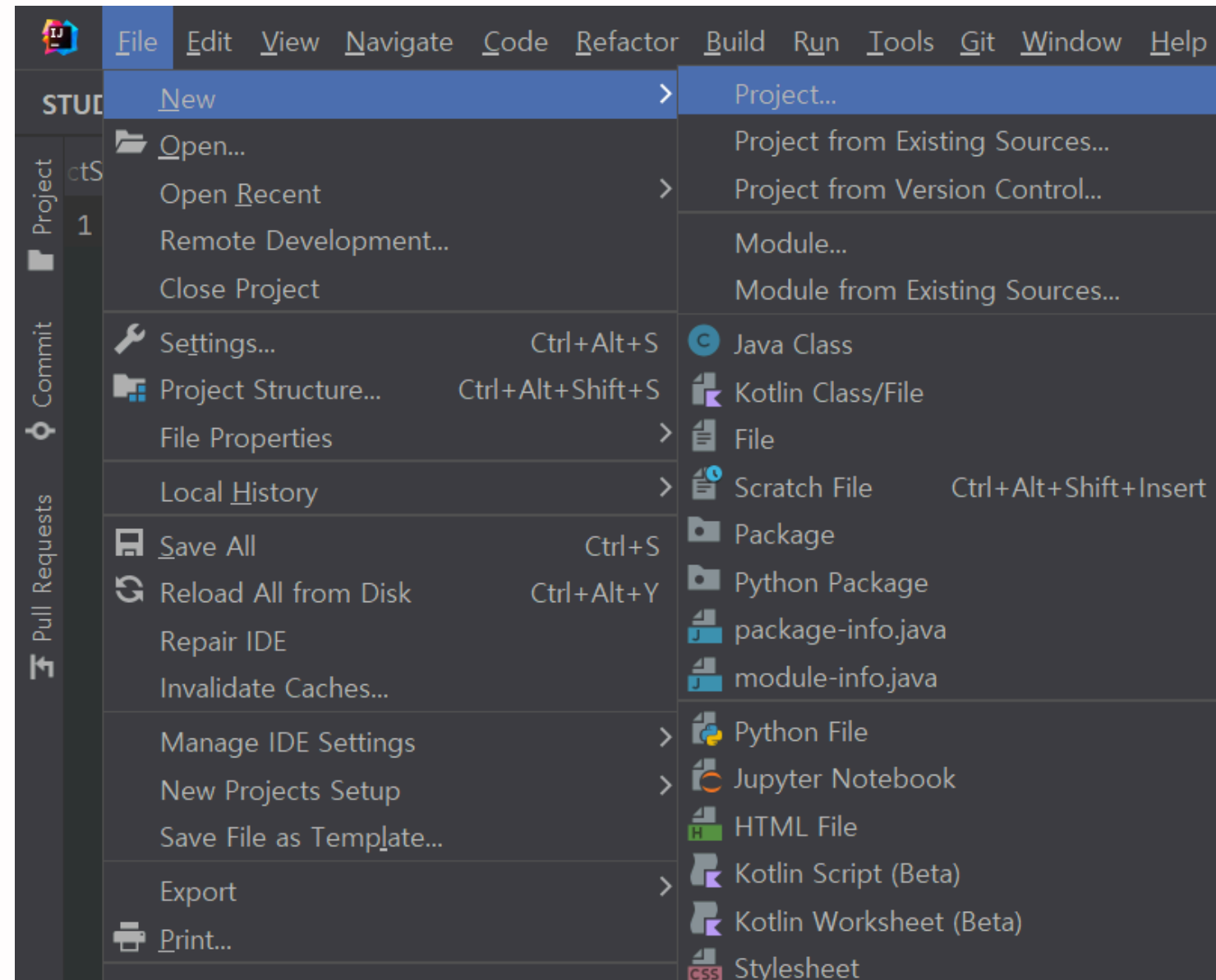
Spring Boot 프로젝트 설정

02

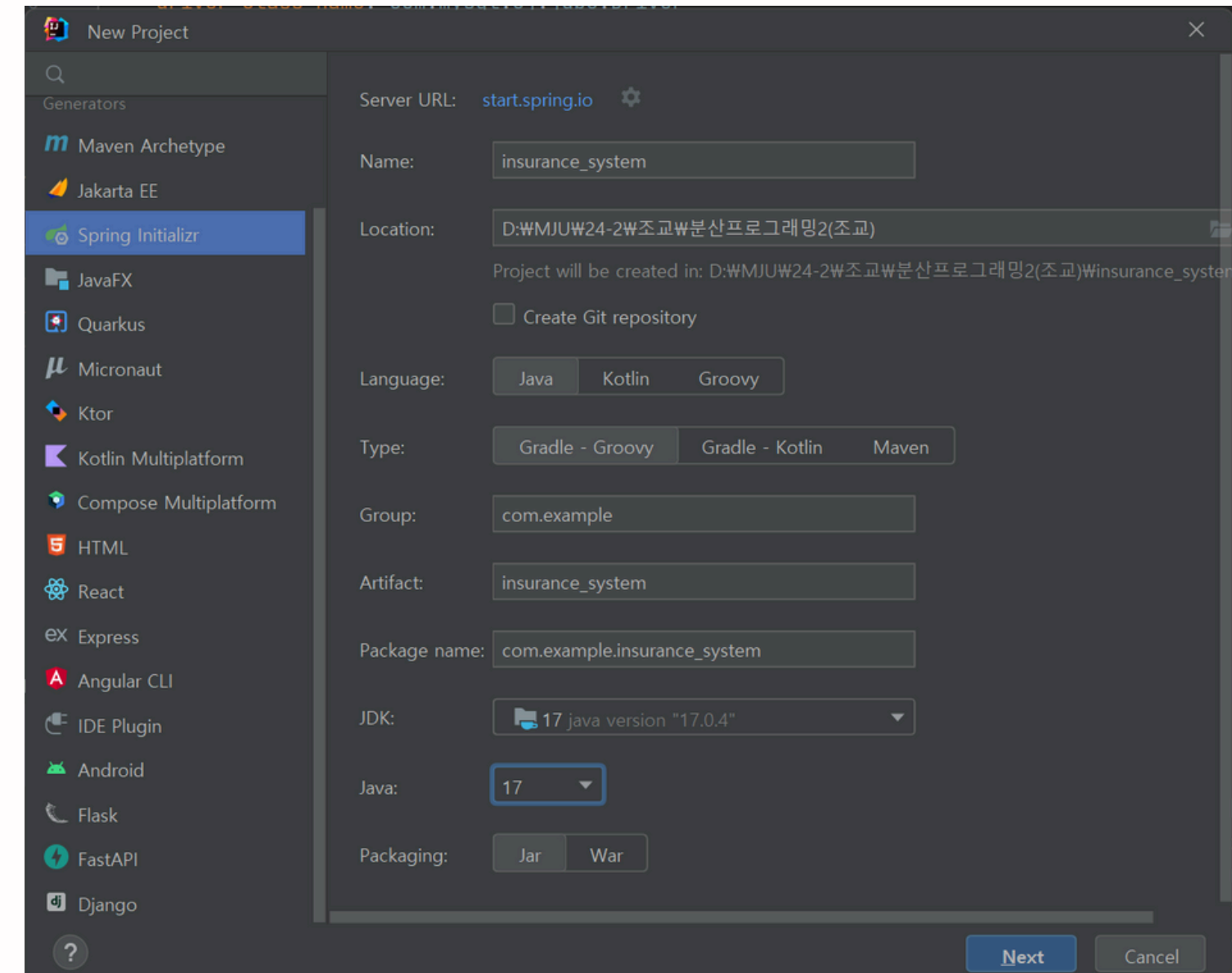
MyBatis를 이용한 DB 연결

사전 설치 : IntelliJ, jdk 17 이상, MySQL

1 인텔리제이 실행 후 좌측 상단 File > New > Project



2 좌측 Spring Initializer 선택 후 이름, 저장 경로 입력

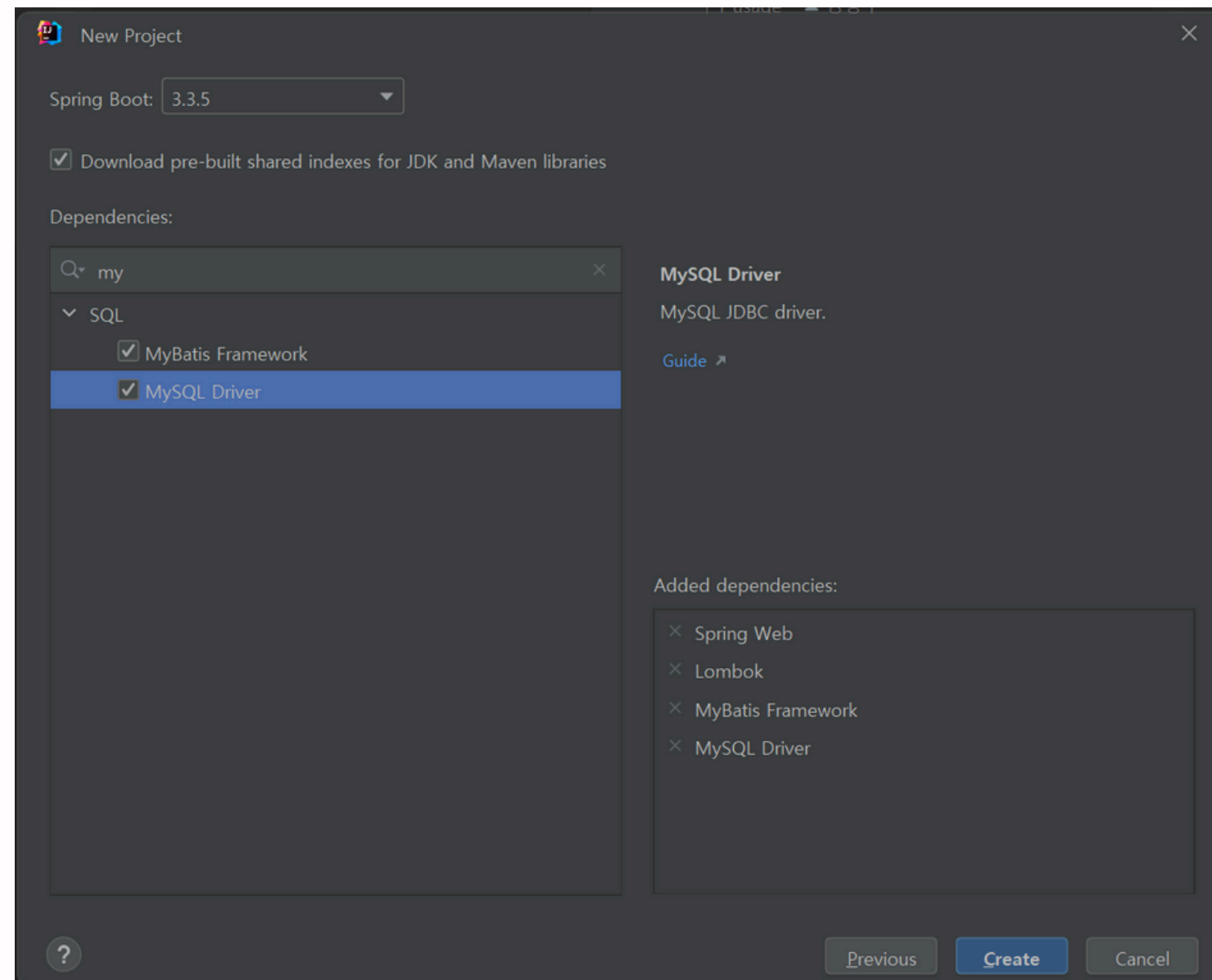


Type : 빌드 유형 선택 (Gradle - Groovy)

Group : 도메인 이름 또는 회사명

Artifact : 프로젝트명

3 의존성 추가



Spring Web : WEB MVC를 사용하여 웹 어플리케이션을 만드는 데 필요한 스프링부트의 기본적인 요소를 가진다. 내장형 컨테이너로 톰캣을 기본으로 탑재하고 있다.

Lombok : 클래스의 Getter, Setter 등을 자동으로 생성해주는 어노테이션을 제공한다.

MyBatis Framework : 자바 오브젝트와 SQL사이의 자동 매핑 기능을 지원하는 ORM인 MyBatis를 사용하기 위해 필요하다.

MySQL Driver : MySQL을 사용하기 위해 필요한 드라이버이다.

MyBatis는 자바 오브젝트와 SQL사이의 자동 매핑 기능을 지원하는
ORM(Object relational Mapping) 프레임워크



SQL문과 프로그래밍 언어 코드를 분리하여 유지보수성과 생산성을 높일 수 있다

SQL 쿼리를 직접 작성할 수 있으므로 유연성이 높다

MyBatis는 캐시 기능을 제공하므로 데이터베이스 연산 속도를 높일 수 있다

다양한 데이터베이스에 대한 지원을 제공한다

·
·
·

1 MySQL CLI 접속하여 DB, 테이블 생성

```
C:\Users\ibaes>mysql -u root -p
Enter password: *****
```

```
mysql> CREATE DATABASE insurance_system;
Query OK, 1 row affected (0.04 sec)
```

```
mysql> show databases;
```

Database
information_schema
insurance_system
mysql
performance_schema
sakila
sys
world

```
14 rows in set (0.06 sec)
```

[MySQL CLI 접속 방법]

cmd창 접속 후 `mysql -u root -p`
password 입력

[데이터베이스 생성 방법]

`CREATE DATABASE 데이터베이스_이름;`

- '데이터베이스_이름'으로 데이터베이스를 생성한다.

SHOW DATABASES;

- 현재 계정에 생성된 데이터베이스 명단을 보여준다.

1 MySQL CLI 접속하여 DB, 테이블 생성

```
mysql> use insurance_system;
Database changed
mysql> CREATE TABLE insurance (
  ->     id BIGINT AUTO_INCREMENT,
  ->     name VARCHAR(255) NOT NULL,
  ->     description VARCHAR(255),
  ->     type VARCHAR(100),
  ->     PRIMARY KEY (id)
  -> );
Query OK, 0 rows affected (0.03 sec)
```

[생성한 데이터베이스 사용]

use 데이터베이스_이름;

[테이블 생성 (예제)]

```
CREATE TABLE insurance (
  id BIGINT AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  description VARCHAR(255),
  type VARCHAR(100),
  PRIMARY KEY (id)
);
```

2 yml 파일 수정

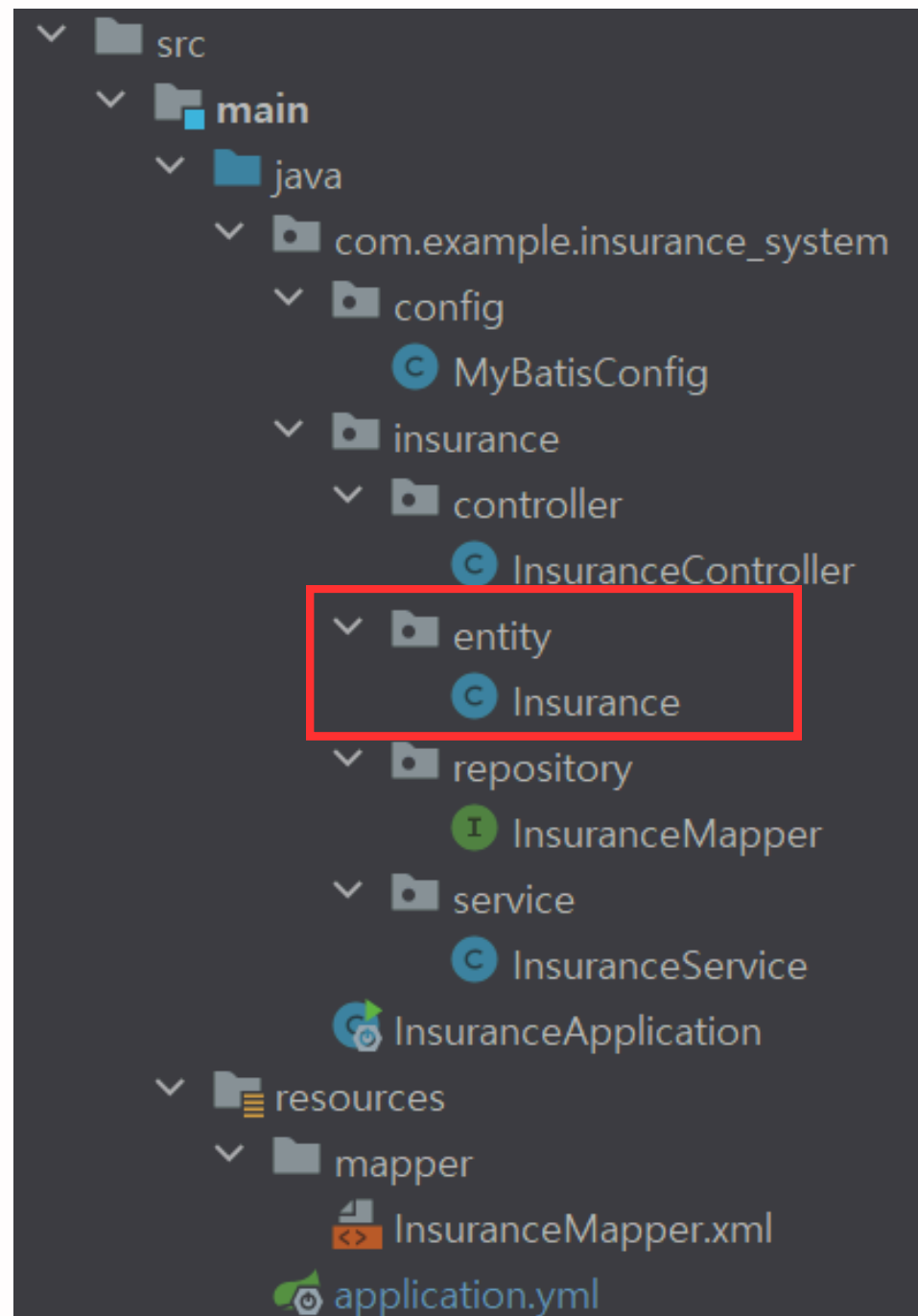
먼저 가독성을 위해, src>main>resources 폴더 내의
application.properties -> application.yml로 확장자 변경

```
1  spring:
2    application:
3      name: insurance_system
4    datasource:
5      driver-class-name: com.mysql.cj.jdbc.Driver
6      url: jdbc:mysql://localhost:3306/insurance_system
7      username: root
8      password:          mysql 비밀번호
9  mybatis:
10   mapper-locations: classpath:mapper/*.xml
11  server:
12   port: 8080
```

spring.application.name : 생성한 프로젝트명
datasource : DB 정보
mybatis.mapper-locations : mapper파일의 위치

본인이 생성한 데이터베이스 이름

3 Insurance 엔티티 생성

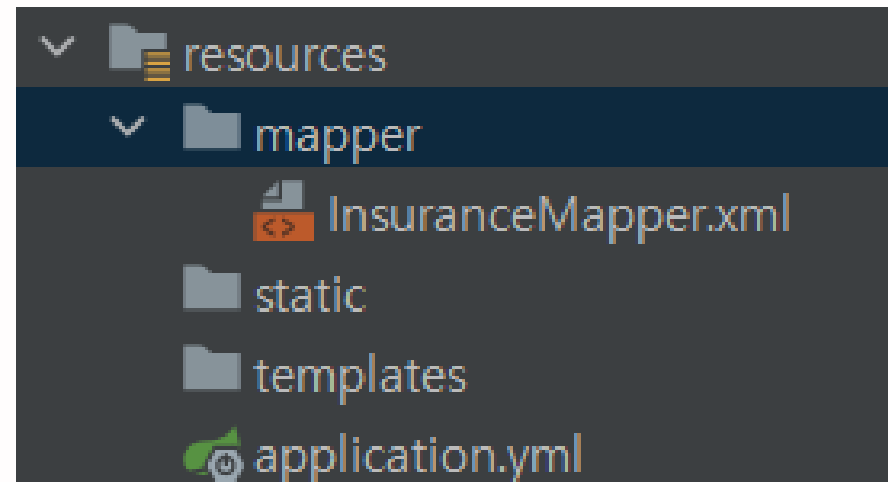


패키지 구조

```
1 package com.example.insurance_system.insurance.entity;  
2  
3 @ibaesuyeon  
4 public class Insurance {  
5     private Long id;  
6     private String name;  
7     private String description;  
8     private String type;  
9 }
```

Insurance.java

4 Mapper 설정

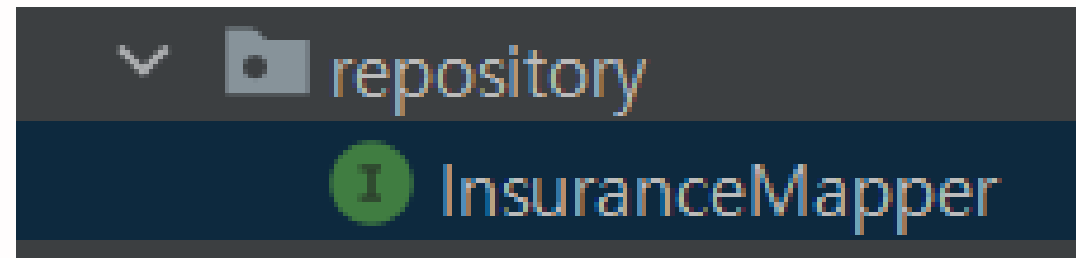


파일 경로

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4
5  <mapper namespace="com.example.insurance_system.insurance.repository.InsuranceMapper">
6      <select id="selectInsuranceList" resultType="com.example.insurance_system.insurance.entity.Insurance">
7          select *
8          from insurance
9      </select>
10 </mapper>
```

InsuranceMapper.xml

5 Mapper 인터페이스 생성

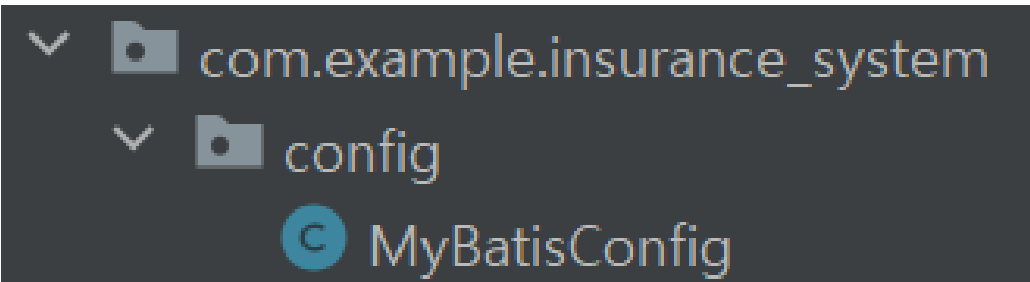


패키지 구조

```
1 package com.example.insurance_system.insurance.repository;
2
3 import ...
4
5
6
7
8 @Mapper
9 public interface InsuranceMapper {
10     List<Insurance> selectInsuranceList();
11 }
```

InsuranceMapper.java

6 MyBatis Config



패키지 구조

```
1 package com.example.insurance_system.config;
2
3 import ...
4
5
6 @Configuration
7 @MapperScan("com.example.insurance_system") // Mapper 인터페이스가 위치한 패키지
8 public class MyBatisConfig {
9 }
```

MyBatisConfig.java

7 컨트롤러, 서비스 코드 작성

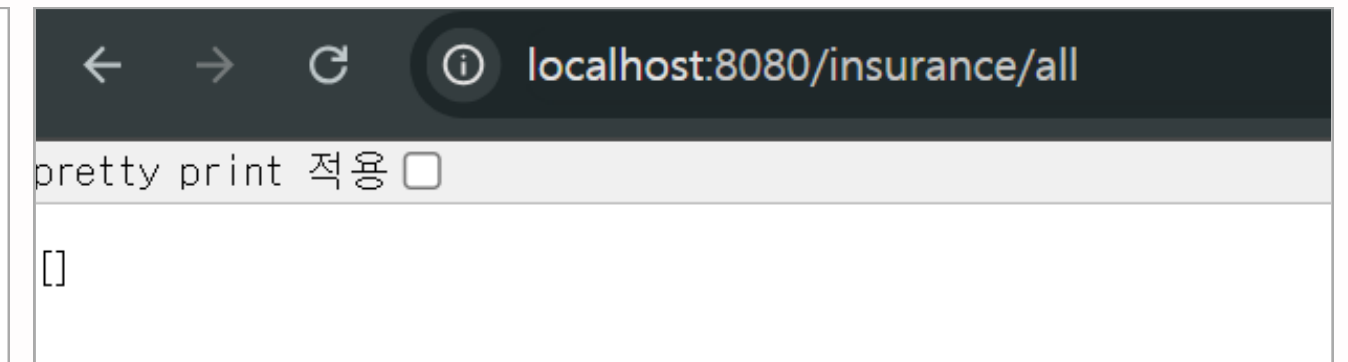
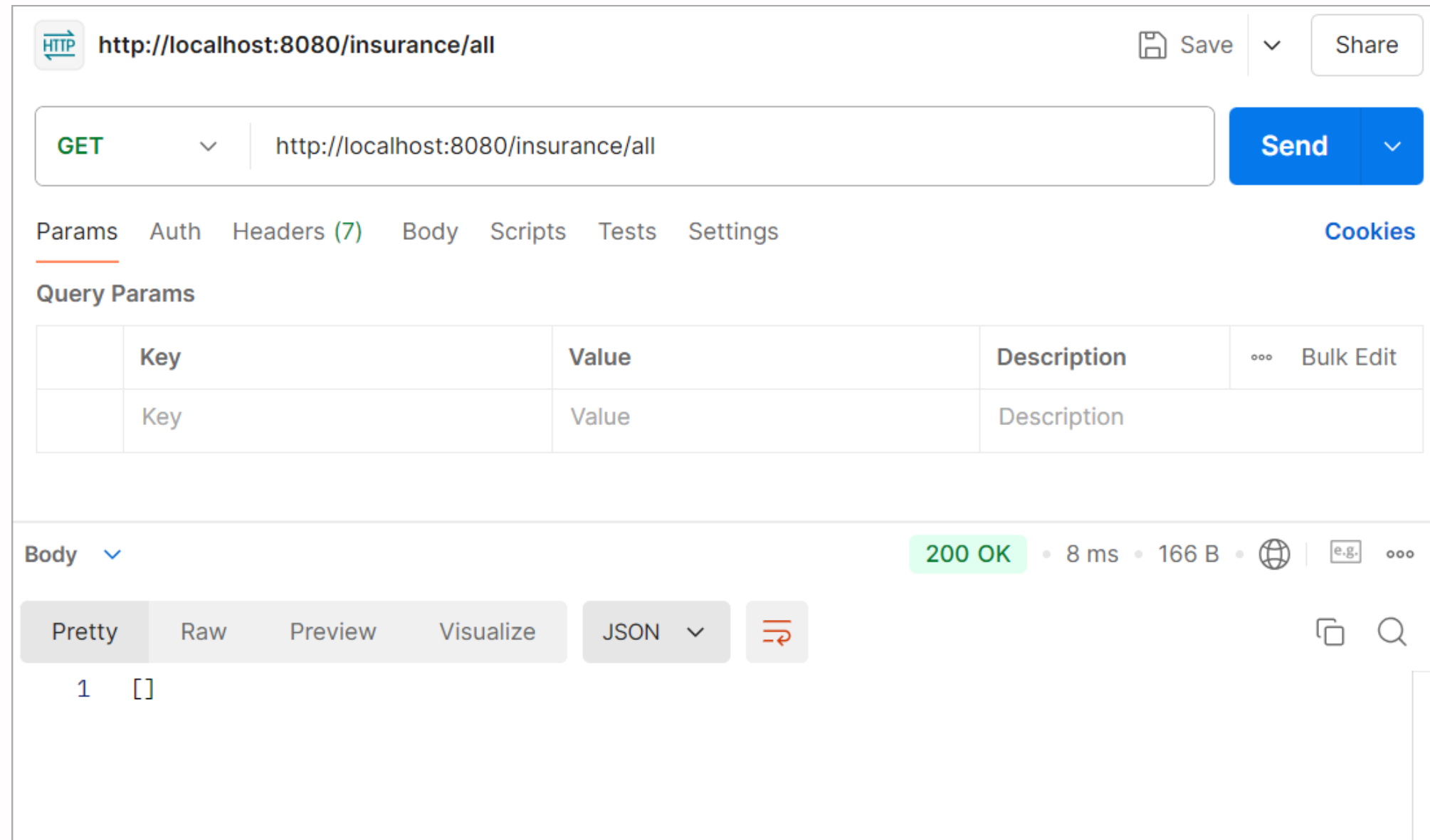
```
1 package com.example.insurance_system.insurance.controller;
2
3 import ...
4
11
12 @RestController
13 @RequestMapping("/insurance")
14 public class InsuranceController {
15     @Autowired
16     InsuranceService insuranceService;
17     @GetMapping("/all")
18     public List<Insurance> getAllInsurance() {
19         return insuranceService.getAllInsurance();
20     }
21 }
```

InsuranceController.java

```
1 package com.example.insurance_system.insurance.service;
2
3 import ...
4
9
10 @Service
11 public class InsuranceService {
12     @Autowired
13     InsuranceMapper insuranceMapper;
14     public List<Insurance> getAllInsurance() {
15         return insuranceMapper.selectInsuranceList();
16     }
17 }
```

InsuranceService.java

8 실행 후 Postman 또는 브라우저에서 확인 현재 테이블이 비어있으므로 []가 맞는 응답



브라우저에서 확인한 결과

Postman으로 확인

SOURCE CODE

<https://github.com/ibaesuyeon/Insurance-System-2024.git>

E.O.D.