**Persistence Framework**

persistence(데이터의 지속성, 영속성), framework(동작에 필요한 구조를 어느 정도갖춘 개발 도구)

데이터의 저장, 조회, 변경, 삭제를 다루는 클래스 및 설정 파일들의 집합

- 장점 : JDBC프로그래밍의 복잡함이나 번거로움 없이 간단하게 데이터베이스와 연동되는 시스템 개발, 안정적인 구동

- 종류

1. SQL Mapper(SQL 문장으로 직접 DB데이터를 다룸) : mybatis

2. ORM(객체를 통해 간접적으로 DB데이터를 다룸) : JPA(Hibernate)

**JDBC (java Database Connecting)**

DB에 접근할 수 있도록 JAVA에서 제공하는 API, 모든 Persistence Framework는 내부적으로 JDBC API를 이용함

**동작 과정**

Application - Persistence Framework – JDBC Interface – JDBC Implemetation - DB

**ORM (Object Relational Mapping, JPA)**

객체와 DB테이블을 맵핑하여 method를 호출해 쿼리 실행

JPA (Java Persistence API) : JAVA ORM을 사용하기 위한 인터페이스를 모아놓은 API

- 장점 : 데이터 접근을 위한 기본적인 method가 정의되어 있음

- 단점 : 복잡한 쿼리가 필요한 경우 부적합

**MyBatis**

임의의 SQL 쿼리문, 그리고 그 쿼리의 결과를 객체와 맵핑할 수 있도록 지원하는 SQL Mapper

동작 과정

DB <-> Mapper <-> DAO(Data Access Object) <-> Controller <-> UI

- 장점 : 복잡한 쿼리가 필요한 경우 적합, 가공된 데이터 또는 특정 컬럼만 추출하기 적합

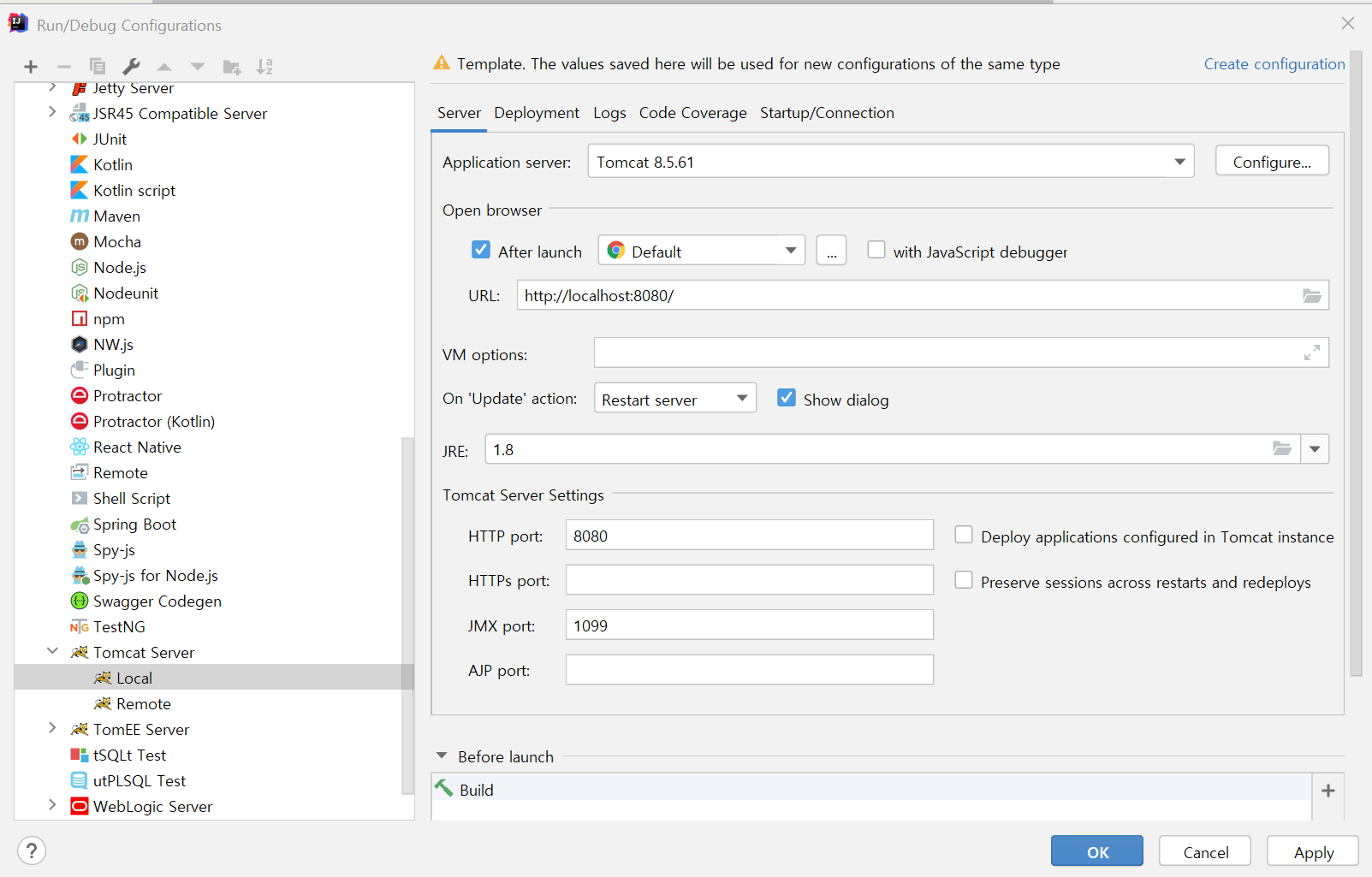
- 단점 : 반복적인 작업이 필요, 일일히 SQL문과 데이터를 맵핑해줘야 한다.

**IntelliJ 개발 환경 구축**

1.IntelliJ, JDK, Tomcat, Gradle 설치

2. 새 프로젝트 생성 후, 상단에

testApplication -> Edit configurations -> Templates -> Tomcat Server -> Local

에서 톰캣 설치 경로 입력 

3. Mysql driver 라이브러리 추가

<https://dev.mysql.com/downloads/connector/j/> 에서 zip 다운 후, C 드라이브에 저장

file -> project structure -> Libraries 에서 +, java -> mysql-connecter .jar 파일 선택 후 저장

4. 한글 설정

- IntelliJ VM 설정

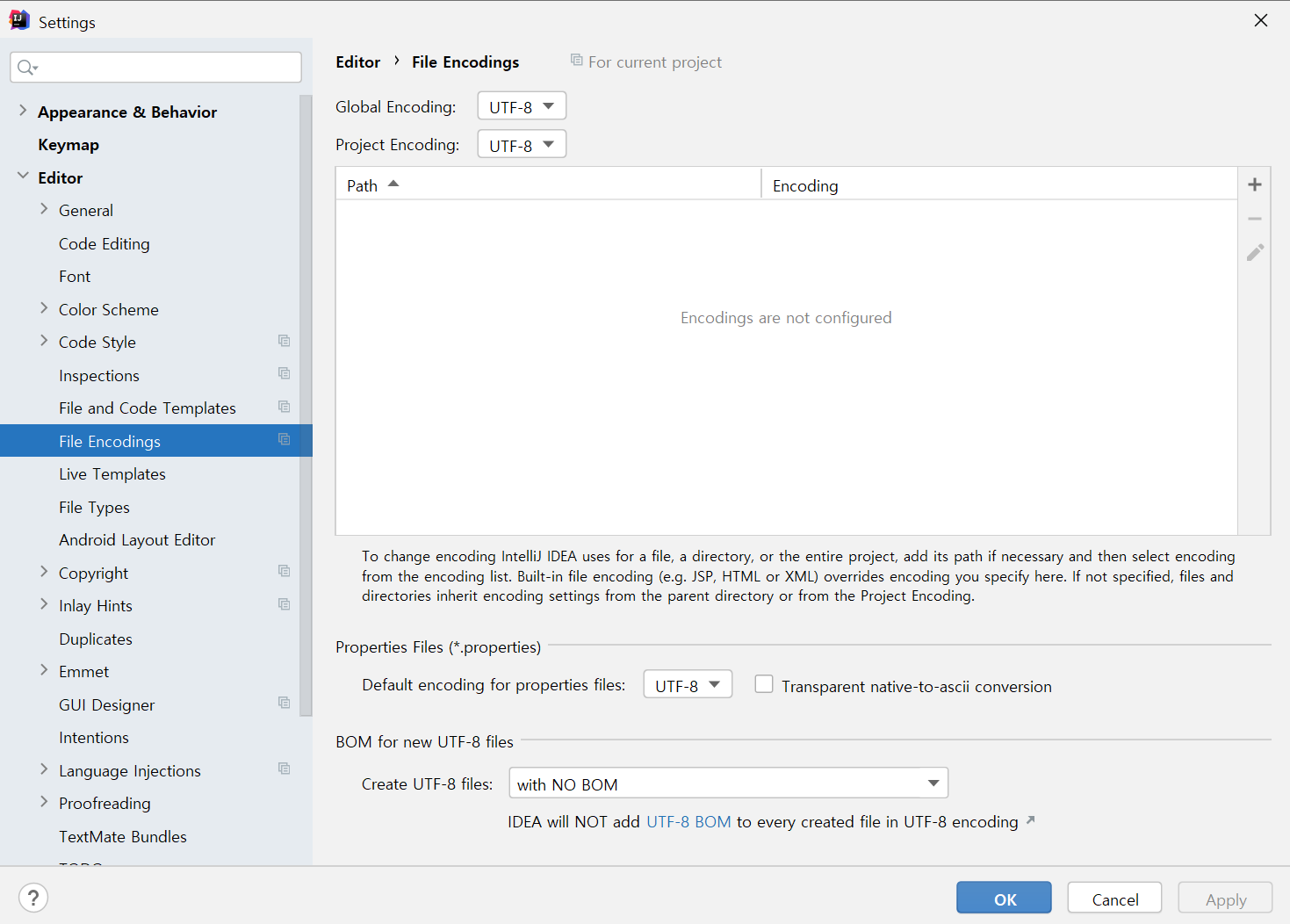
IntelliJ 설치 파일의 bin 파일로 이동 후, vmoptions 파일 열기

C:\Program Files\IntelliJ IDEA 2020.2.2\bin -> idea64.exe.vmoptions

해당 파일 백업본 하나 생성 후, 맨 마지막에 “-Dfile.encoding=UTF-8” 추가, 후 재시작

-Intellij Editor File Encoding 설정

File -> settings -> Editor -> file encoding -> 모든 설정 “UTF-8”로 변경



-톰캣 인코딩 설정

Run -> edit configurations -> Tomcat Server VM option에 "-Dfile.encoding=UTF-8" 추가

**HelloWorld**

1.기본 설정 파일

- Build.gradle파일

dependencies **{** implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
 implementation 'org.springframework.boot:spring-boot-starter-web'  
 implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:2.1.4'  
 compileOnly 'org.projectlombok:lombok'  
 annotationProcessor 'org.projectlombok:lombok'  
 providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'  
 testImplementation 'org.springframework.boot:spring-boot-starter-test'  
 implementation "mysql:mysql-connector-java:5.1.34"  
**}**

- application.properties

spring.datasource.url=jdbc:mysql://localhost/testdb  
spring.datasource.username=root  
spring.datasource.password=mylee1  
spring.datasource.driver-class-name=com.mysql.jdbc.Driver  
  
spring.datasource.tomcat.testOnBorrow=true  
spring.datasource.tomcat.validationQuery=SELECT 1

2. HelloController.java 파일   
import lombok.Getter;  
import lombok.Setter;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
@Controller *// Spring에 해당 클래스가 Controller라는 것을 알려주기 위함*public class HelloController {  
 */\*  
 1. 화면에 helloworld가 출력됩니다.  
 \*/* @GetMapping(value = "/helloworld/string") *// 해당 주소의 Resource를 이용하기 위해 Get method로 호출할 수 있게 하는 어노테이션* @ResponseBody *// 그대로 결과를 출력한다는 뜻* public String helloworldString() {  
 return "helloworld";  
 }  
 */\*  
 2. 화면에 {message:"helloworld"} 라고 출력됩니다.  
 \*/* @GetMapping(value = "/helloworld/json")  
 @ResponseBody  
 public Hello helloworldJson() {  
 Hello hello = new Hello();  
 hello.message = "helloworld";  
 return hello;  
 }  
  
 @Setter  
 @Getter  
 public static class Hello {  
 private String message;  
 }  
   
}

**발생 오류**

1. com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure

->Application.properties 파일에 추가

spring.datasource.tomcat.testOnBorrow=true  
spring.datasource.tomcat.validationQuery=SELECT 1

: DB Pool이 생성이 되더라도 특정 시간 호출되지 않으면 DBMS 설정에 따라 연결이 끊어질 수 있음, 이 설정을 추가하면 연결을 유지시켜줌

2. Web server failed to start. Port 8080 was already in use.

-> 8080포트가 사용 중임

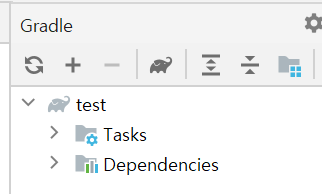
Cmd 에서 8080 포트 사용 중인거 죽이기

Netstat -ano (사용 중인 포트 확인)

Taskkill /pid 11024 /f (pid가 11024인 포트 죽이기)

3. build.gradle re-import

라이브러리를 추가했는데 인식하지 못할 때, 오른쪽의 Gradle 밑 재시작 버전을 누르면됌



Gradle과 Maven의 차이

Spring과 Spring Boot?

Spring boot는 spring안에 tomcat이 내장되어 있어 web개발에 수월하게 함

스프링부트는 스트링 프레임워크라는 큰 틀에 속하는 도구

MVC 모델 모델, 뷰, 컨트롤러란?

Model View Controller

뷰처리는 view에서, 데이터는 model, url에 따른 페이지 이동은 controller에서 처리

Html과 JSP의 차이

Java의 데이터를 이용해 뷰에서의 화면을 구성하는 것 j네

Controller, Repository, Service, DAO, VO 구조에 대해 알아보기

@Controller는 웹어플리케이션에서 컨트롤러 역할을 하는 클래스

@service 로직을 수행할 클래스

@repository DAO같은 데이터베이스와 과련된 클래스

DAO, DAOImple 또는 VO, VOImpl 알아보기