

Spring Boot 02

[프로젝트 생성, 설정]

실전! 스프링 부트와 JPA 활용 1 [복습]

김영한

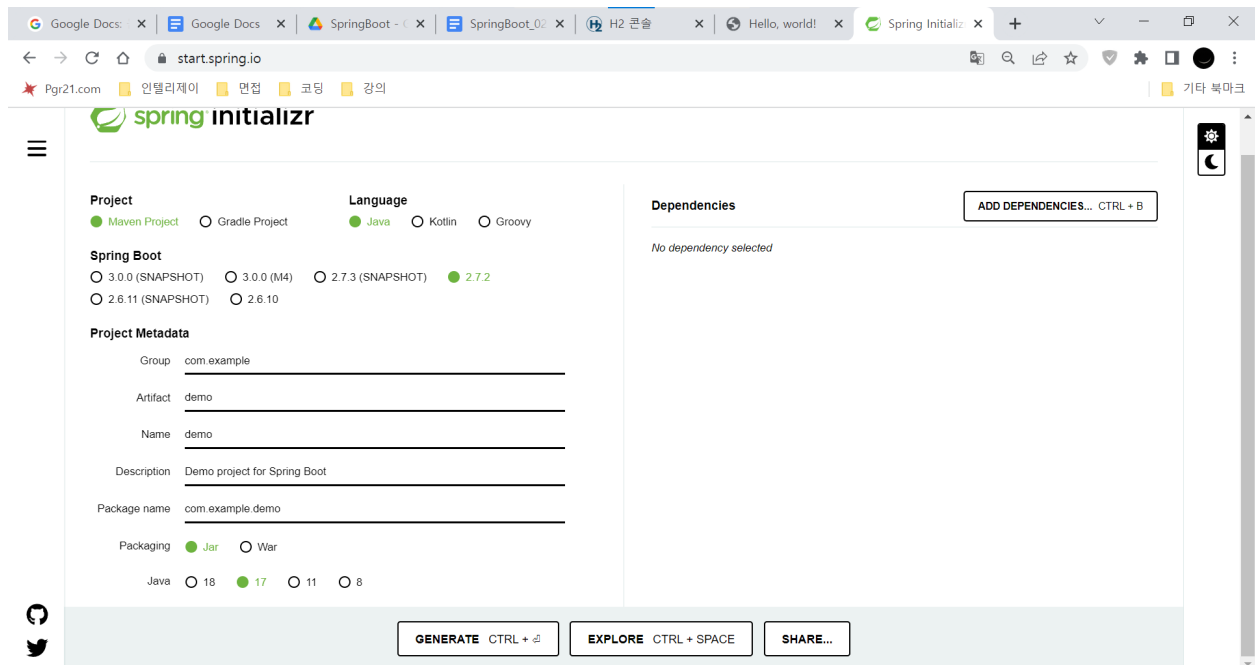
2022.08.10

[Spring Boot Reference Documentation](https://spring.io/reference)

프로젝트 생성

<https://start.spring.io>

에서 설정을 간편히 만들수 있다.



SpringBoot 버전은 M4, SNAPSHOT 은 아직 안정화가 덜 된 버전이다.

Dependencies에 원하는 툴을 추가하면 필요한 라이브러리까지 전부 가져온다.

롬복 설정

- IntelliJ plug-in에 Lombok 다운로드

-
- Annotation 검색후 - Enable annotation processing 체크 (필수)

resources

Static

-정적 페이지

template

-템플릿이 적용된 페이지 (동적)

```
implementation 'org.springframework.boot:spring-boot-devtools'
```

리로딩을 편하게 하는 라이브러리

JPA 설정 파일

Gradle

resources → application.yml 생성

```
spring:
  datasource:
    url: jdbc:h2:tcp://localhost/~/jpashop
    username: sa
    password:
    driver-class-name: org.h2.Driver

  jpa:
    hibernate:
      ddl-auto: create          →엔티티를 지우고 다시 생성
    properties:
      hibernate:
        #      show_sql: true
        format_sql: true

  logging:
    level:
      org.hibernate.SQL: debug
      org.hibernate.type: trace
```

H2 database시 설정

Maven

resources/ META-INF/ persistence.xml 생성 후 설정

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">
    <persistence-unit name="hello">
        <properties>
            <!-- 필수 속성 -->
            <property name="javax.persistence.jdbc.driver" value="org.h2.Driver"/>
            <property name="javax.persistence.jdbc.user" value="sa"/>
            <property name="javax.persistence.jdbc.password" value=""/>
            <property name="javax.persistence.jdbc.url"
value="jdbc:h2:tcp://localhost/~/test"/>
            <property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect"/>

            <!-- 옵션 -->
            <property name="hibernate.show_sql" value="true"/>
            <property name="hibernate.format_sql" value="true"/>
            <property name="hibernate.use_sql_comments" value="true"/>
            <property name="hibernate.jdbc.batch_size" value="10"/>
            <property name="hibernate.hbm2ddl.auto" value="create" />
        </properties>
    </persistence-unit>
</persistence>
```

테스트 코드 -Save

```
public void testMember() throws Exception {
    //given
    Member member = new Member();
    member.setName("memberA");

    //when
    Long savedId = memberRepository2.save(member);
    Member findMember = memberRepository2.find(savedId);

    //then
    Assertions.assertThat(findMember.getId()).isEqualTo(member.getId());
    Assertions.assertThat(findMember).isEqualTo(member);
}
```

→ 마지막 Assertions 테스트에서 findMember == member 는 같은 영속성 컨텍스트에서 가져왔기 때문에 == 을 보장해준다.

배포시

Gradle 빌드해서 올리는 것.

과정 → 빌드 설정 → 서버에서 실행 (IP, 포트 설정) → DNS 로 도메인 설정

외부에서 접근 가능하게 주소로 접근가능 . 서버에서 실행하는 부분이 배포이다.

스프링 부트 **Gradle** 설정시

-라이브러리 버전 ? 버전이 없는 것은 스프링 부트가 설정해 놓은 궁합맞는 버전이 설정되어있음

```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-validation'
    implementation 'org.springframework.boot:spring-boot-devtools'
    implementation 'com.github.gavlyukovskiy:p6spy-spring-boot-starter:1.5.6'
    compileOnly 'org.projectlombok:lombok'
    runtimeOnly 'com.h2database:h2'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}
```
