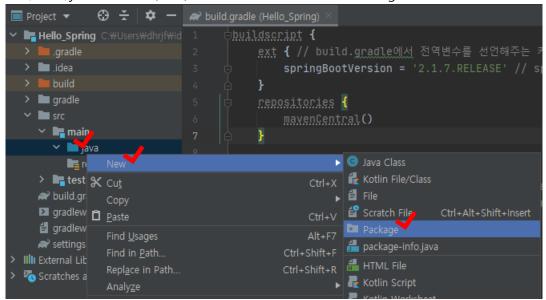
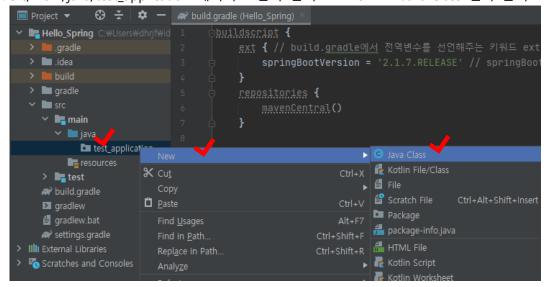
Simple Server

실습1) Spring Boot Simple Server

- 1. Controller 클래스 생성
 - 1.1. Package 생성
 - : src/main/java 디렉터리(폴더) 오른쪽 클릭 => New => Package 왼쪽 클릭

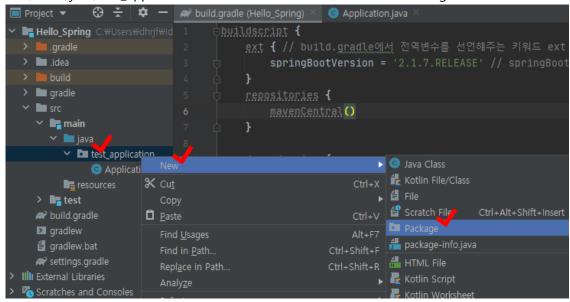


- 1.2. 패키지 이름은 test_application으로 작성.
- 1.3. test_application 패키지 내에 Java Class 생성
- : src/main/java/test_application 패키지 오른쪽 클릭 => New => Java Class 왼쪽 클릭

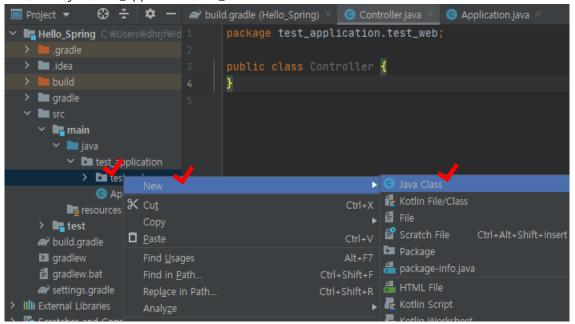


- 1.4. Java Class 이름은 Application으로 작성
- 1.5. test_application 패키지 하위에 새로운 패키지 생성

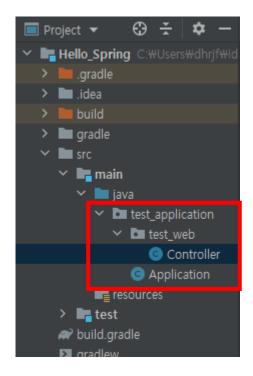
: src/main/java/test_application 패키지 오른쪽 클릭 => New => Package 왼쪽 클릭



- 1.6. 패키지 이름은 test_web으로 작성
- 1.7. test_web 패키지 내에 Java Class 생성
- : src/main/java/test_application/test_web 패키지 오른쪽 클릭 => New => Java Class 왼쪽 클릭



- 1.8. Java Class 이름은 Controller로 지정.
- 1.9. { 1.1. ~ 1.8. } 문항까지 완료하였다면 아래 사진과 같이 프로젝트가 형성됐을 것이다.



2. Application, Controller 코드 작성

2.1. Application 코드

```
package test_application;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

// @SpringBootApplication 어노테이션으로 설정된 클래스는 반드시 프로젝트의 최상단에 위치하여야 한다.
@SpringBootApplication // 스프링 부트를 자동으로 설정하기 위해 사용 (해당 위치부터 설정을 읽기 때문에 최상단에 위치)
public class Application { // 메인 클래스
   public static void main(String[] args){
        SpringApplication.run(Application.class, args); // 내장 WAS 실행
   }
}
```

2.2. Controller 코드

```
package test_application.test_web;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController // Client에게 view(*.js, *html 등)가 아닌 data자체를 넘겨주기 위한 방법.
public class Controller {

@GetMapping("/") // GET요청에 대한 응답을 매핑 시켜준다.
public String hello() { // 응답
    return "Hello, Spring!!";
}
```

3. Application클래스의 main메소드로 Spring 실행

• 3.1. Application 클래스의 main메소드옆에 있는 2번째 초록색 화살표 모양 클릭

```
■ Project ▼
               🛟 🛨 💠 — 🔊 build.gradle (Hello_Spring) × 💿 Controller.java × 🜀 Application.java
                                       package test_application;

➤ Hello_Spring C:\Users\Users\Underst\text{dhrjf\Understand}

  > 🖿 .idea
                                        import org.springframework.boot.SpringApplication;
                                        import org.springframework.boot.autoconfigure.SpringBootApplicatio
  > gradle

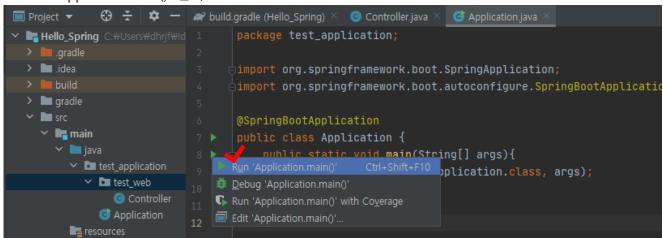
✓ Image: Src

                                       @SpringBootApplication
     🗸 📭 main
                                       public class Application {
                                            public static void main(String[] args){
          SpringApplication.run(Application.class, args);

✓ Image: Very very test web.

                 🧐 Controller
                Application
          resources
```

3.2. Run 'Application.main()' 클릭

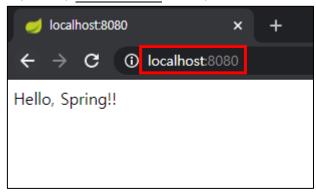


3.3. IntelliJ IDEA 하단의 Run탭을 클릭 시 Spring이 동작하는 것을 확인가능

```
> Task :compileJava
> Task :classes
 :: Spring Boot ::
2020-09-02 20:15:49.939 INFO 31412 --- [
                                                   main] test_application.Application
                                                                                                  : Starting Application on DESKTOP-DBP1SQF
2020-09-02 20:15:49.939 INFO 31412 --- [
                                                                                                  : No active profile set, falling back to
                                                   main] test_application.Application
2020-09-02 20:15:53.087 INFO 31412 ---
                                                   main] o.apache.catalina.core.StandardService
2020-09-02 20:15:53.087 INFO 31412 ---
                                                   main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat,
2020-09-02 20:15:53.238 INFO 31412 --- [
                                                   mainl o.a.c.c.C.[Tomcat].[localhost].[/]
                                                                                                  : Initializing Spring embedded WebApplica
2020-09-02 20:15:53.238 INFO 31412 ---
                                                                                                  : Root WebApplicationContext: initializat
2020-09-02 20:15:53.684 INFO 31412 ---
2020-09-02 20:15:53.684 INFO 31412 ---
                                                                                                  : Started Application in 4.194 seconds (3
```

(사진에서 Tomcat 포트가 8080번으로 열린 것을 확인할 수 있다.)

- 3.4. 클라이언트(웹 브라우저)에서 실행결과 확인
- : 주소창에 <u>localhost:8080</u>을 입력



(본 실습에서는 크롬 웹 브라우저를 사용하였다.)

참고문헌

• 이동욱 (2019). 스프링 부트와 AWS로 혼자 구현하는 웹 서비스. 54-66.