

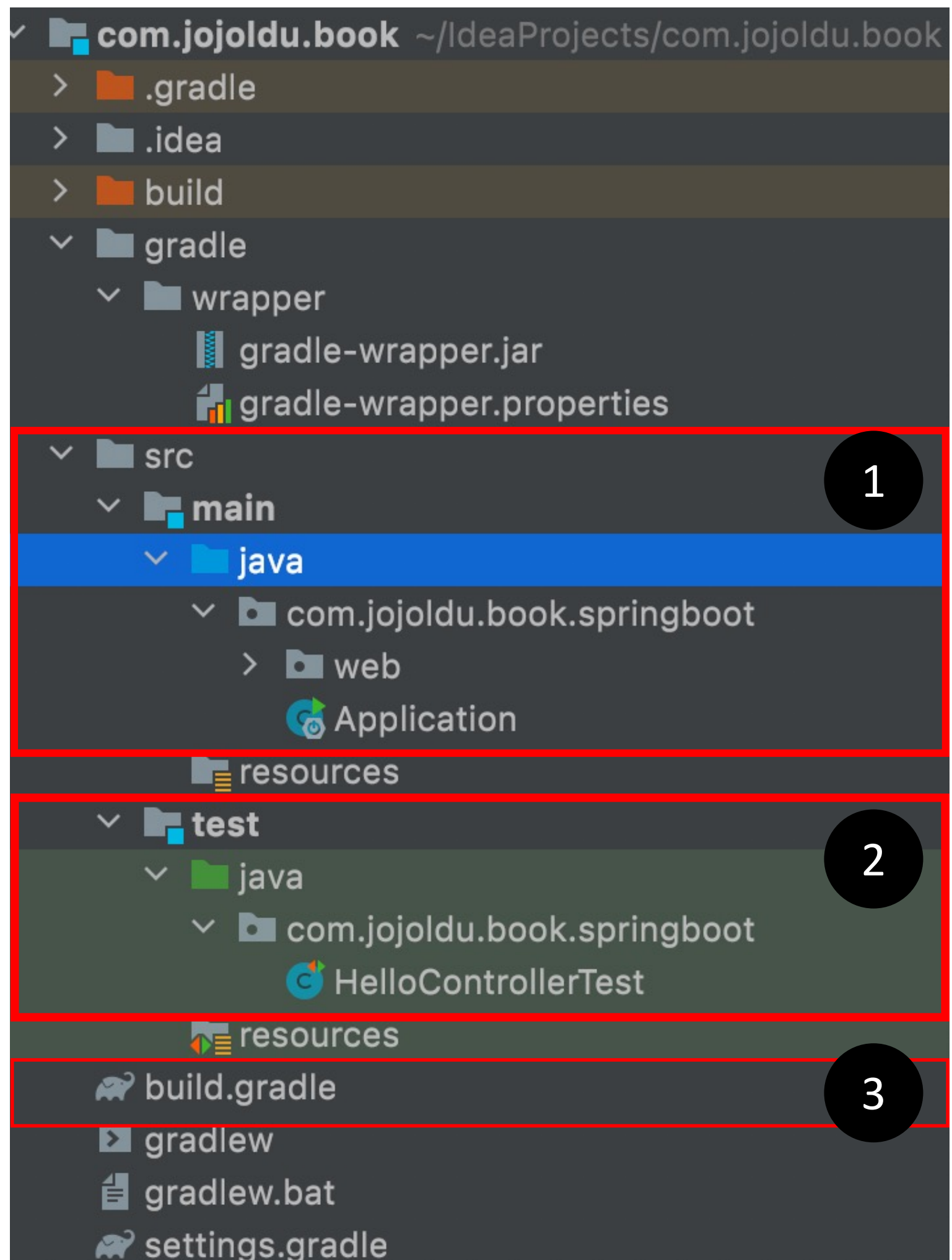
# Spring Boot

스프링 부트에서 테스트 코드를 작성하자 (2)

신온유  
정보보호학과  
2017111683



# 프로젝트 구조



1. 메인클래스 코드를 작성하는 부분

2. 메인클래스에 작성한 코드를 테스트하는 테스트 코드를 작성하는 부분

3. Grandle 설정을 변경하는 파일

# 메인 클래스

```
package com.jojoldu.book.springboot;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
//스프링부트의 자동설정 및 스프링 Bean 일기와 생성을 모두 자동으로 설정해준다.
//이 명령어가 있는 위치부터 설정을 읽어가기 때문에 항상 프로젝트의 최상단에 위치해야합니다.
public class Application {
    public static void main(String[] args){
        SpringApplication.run(Application.class,args);
        //main에서 실행하는 SpringApplication으로 인해 내장 WAS(web application server) 실행할수있게됨
        //이것 덕분에 별도의 was를 두지않고 어플리케이션을 실행할 때 내부에서 was가 실행된다.
        //항상 톰캣을 설치할 필요가 없으며 스프링 부트로 만들어진 jar 파일로 실행하면 된다.
        //언제 어디서나 같은 환경에서 스프링부트를 배포 할 수 있기 때문에 내장 WAS를 사용하는 것을 권장

        //내장 WAS를 쓰지 않을 경우 -> 모든 서버가 WAS의 종류와 버전 설정을 일치시켜야만 함!
    }
}
```

# 컨트롤러 클래스

```
package com.jojoldu.book.springboot.web;

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

@RestController
//컨트롤러를 json으로 반환하는 컨트롤러로 만들어주는 역할을 합니다.
public class HelloController {
    @GetMapping("/hello")
    //HTTP Method인 Get의 요청을 받을 수 있는 API를 만들어 줍니다.
    public String hello(){
        return "hello";
    }
    // /hello로 요청이 오면 문자열 hello를 반환하는 기능을 가지게 되었음.
    // 이 코드를 테스트 코드를 이용하여 검증해볼 것
}
```



# 메인 클래스 테스트 코드

```
@RunWith(SpringRunner.class)
//여기서는 SpringRunner라는 스프링 실행자를 사용함
//스프링부트 테스트와 JUnit(자바의 단위테스트 도구) 사이에 연결자 역할을 함

@WebMvcTest(controllers = HelloController.class)
//Web에 집중 할 수 있는 스프링부트 어노테이션
//해당 어노테이션의 컨트롤러를 사용할것이기 때문에 작성한 코드

public class HelloControllerTest {

    @Autowired
    //스프링이 관리하는 Bean을 주입받음
    private MockMvc mvc;

    //웹 API를 테스트할 때 사용함
    //스프링 MVC 테스트의 시작점임
    //해당 클래스를 이용하여 HTTP GET,POST등에 대한 API 테스트를 할 수 있음

    @Test
    @WithMockUser
    public void hello가_리턴된다() throws Exception{
        String hello = "hello";
        mvc.perform(get(urlTemplate: "/hello"))
            .andExpect(status().isOk())
            .andExpect(content().string(hello));

        //MockMvc를 통해 /hello 주소로 HTTP GET 요청을 합니다.
        //체이닝을 이용하여 여러가지 기능을 이어서 선언할 수 있다
        //Http header의 status를 검증합니다. 여기에서는 200상태인지 체크합니다
        //mvc.perform의 결과를 검증함 -> 응답 본문의 내용을 검증한다.
        //Controller 에서 "hello"를 리턴하기 때문에 이 값이 맞는지 검증함.
    }
}
```

# 테스트 코드 실행

✓ Tests passed: 1 of 1 test – 134 ms

✓ Test Results134 ms

```
.  ---- -  ----
Δ /  _ _ _ _ ( ) _ _ _ _ \ \ \ \
( ( ) \ _ _ | ' | ' | | ' _ \ / _ ' | \ \ \ \
\ / _ _ ) | | ) | | | | | | ( | | ) ) ) )
' | _ _ _ | . _ | | | | | | \ _ _ , | / / / /
=====|_|=====|_ _ / = / _ / _ /
:: Spring Boot ::      (v2.1.9.RELEASE)

2021-10-28 16:41:31.778 INFO 5566 --- [    Test worker] c.j.book.springboot.HelloControllerTest : Starting HelloControllerTest on ony
2021-10-28 16:41:31.779 INFO 5566 --- [    Test worker] c.j.book.springboot.HelloControllerTest : No active profile set, falling back
2021-10-28 16:41:32.450 INFO 5566 --- [    Test worker] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'appli
2021-10-28 16:41:32.531 WARN 5566 --- [    Test worker] o.s.b.a.m.MustacheAutoConfiguration      : Cannot find template location: clas
2021-10-28 16:41:32.646 INFO 5566 --- [    Test worker] .s.s.UserDetailsServiceAutoConfiguration :

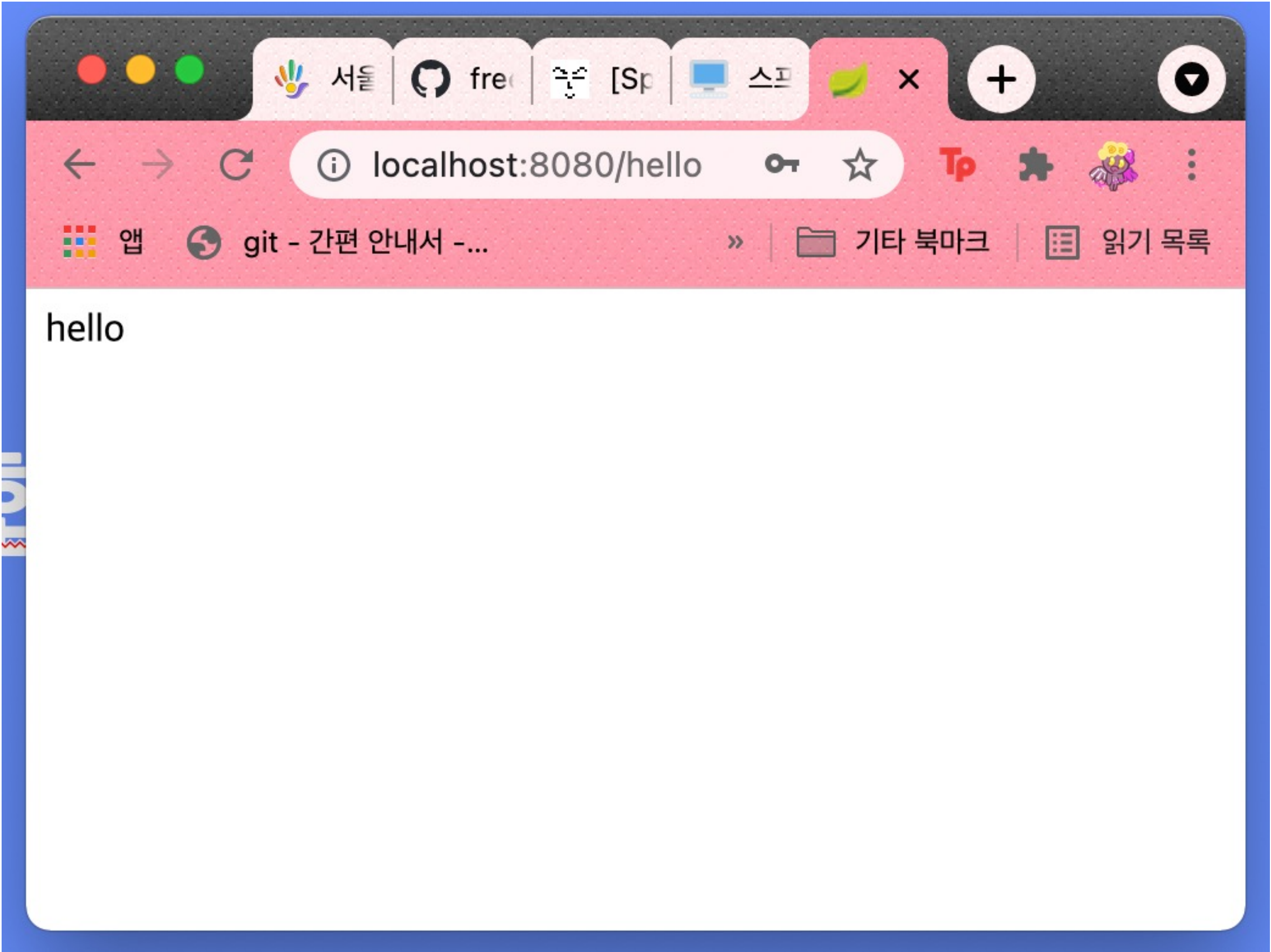
Using generated security password: 753acf6a-ebc8-41ac-b0b8-114d7e0fe6f0

2021-10-28 16:41:32.780 INFO 5566 --- [    Test worker] o.s.s.web.DefaultSecurityFilterChain     : Creating filter chain: any request,
2021-10-28 16:41:32.809 INFO 5566 --- [    Test worker] o.s.b.t.m.w.SpringBootMockServletContext : Initializing Spring TestDispatcherS
2021-10-28 16:41:32.810 INFO 5566 --- [    Test worker] o.s.t.web.servlet.TestDispatcherServlet : Initializing Servlet ''
2021-10-28 16:41:32.818 INFO 5566 --- [    Test worker] o.s.t.web.servlet.TestDispatcherServlet : Completed initialization in 8 ms
2021-10-28 16:41:32.849 INFO 5566 --- [    Test worker] c.j.book.springboot.HelloControllerTest : Started HelloControllerTest in 1.19
2021-10-28 16:41:32.992 INFO 5566 --- [Thread-5] o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'appli

Deprecated Gradle features were used in this build, making it incompatible with Gradle 5.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/4.10.2/userguide/command\_line\_interface.html#sec:command\_line\_warnings
BUILD SUCCESSFUL in 2s
3 actionable tasks: 1 executed, 2 up-to-date
4:41:33 오후: Task execution finished ':test --tests "com.jojoldu.book.springboot.HelloControllerTest.hello가_리턴된다"'.

```

# 실행화면





감사합니다

