

05. Git

컴퓨터공학부 201911228 홍지우

01. Spring Initializr[https://start.spring.io]에서 간단한 SpringBoot 프로젝트를 만들어서 IntelliJ로 열어보세요

Project

☐ Maven Project

☒ Gradle Project

Language

☐ Java

☒ Kotlin

☐ Groovy

Spring Boot

☐ 2.5.0 (SNAPSHOT)

☐ 2.5.0 (RC1)

☐ 2.4.6 (SNAPSHOT)

☒ 2.4.5

☐ 2.3.11 (SNAPSHOT)

☐ 2.3.10

Project Metadata

Group com.example

Artifact demo

Name demo

Description Demo project for Spring Boot

Package name com.example.demo

Packaging

☒ Jar

☐ War

Java

☐ 16

☒ 11

☐ 8

Dependencies

ADD ... CTRL + B

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Security

SECURITY

Highly customizable authentication and access-control framework for Spring applications.

Spring Data JPA

SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database

SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

•Project: 사용할 빌드 툴 선택 (Maven/Gradle)

Maven, Gradle은 프로젝트에 필요한 의존성을 관리하고 빌드 라이프사이클을 관리해주는 툴이다.

•Language: 언어 선택

•Spring Boot: 버전 선택

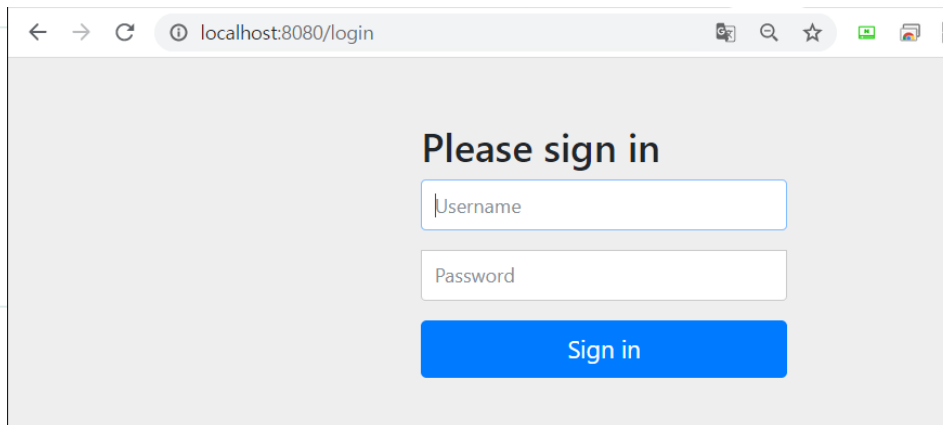
SNAPSHOT이 붙은 것은 현재 개발중인 버전, M(Minor)은 정식 릴리즈되지 않은 버전.

아무것도 붙어있지 않은 것이 정식 릴리즈된 버전이다.

•Project Metadata:

보통 Group은 기업의 도메인명을 적는다.

Artifact는 빌드 결과물 이름을 의미한다.



02. RestController - GetMapping[1]

- RestController 어노테이션과, GetMapping 어노테이션을 사용해서 "hello, world"를 반환하는 컨트롤러 클래스를 작성
- endpoint 정의는 "/api/v1/hello"

```
@RestController
@RequestMapping("api")
public class GetController{
    @GetMapping("/v1/hello")
    public String getParameter(){
        return "Hello World!";
    }
}
```

The screenshot shows a REST client interface with the following components:

- URL Bar:** `http://localhost:8080/api/v1/hello` with a `Save` button and a dropdown arrow.
- Method and URL:** `GET` method selected, with the same URL in the input field. A `Send` button is to the right.
- Tabs:** `Params`, `Auth`, `Headers (7)`, `Body`, `Pre-req.`, `Tests`, `Settings`, and `Cookies`.
- Query Params Table:**

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		
- Response Section:**
 - `Body` tab selected.
 - Status: `200 OK`, Time: `14 ms`, Size: `176 B`.
 - Buttons: `Save Response` with a dropdown arrow.
 - Format buttons: `Pretty` (selected), `Raw`, `Preview`, `Visualize`, `Text` with a dropdown arrow, and a `Copy` icon.
 - Response body: `1 Hello World!`

03. RestController - GetMapping[2]

- RestController 어노테이션과, GetMapping 어노테이션을 사용해서 특정 클래스를 반환하는 컨트롤러 클래스를 작성
- endpoint 정의는 `"/api/v1/class"`

```
@GetMapping("/v1/class")
public User getParameter2(){
    User user = new User(22,"jiwoo");
    return user;
}
```

http://localhost:8080/api/v1/class Save

GET ⌵ http://localhost:8080/api/v1/class

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body Cookies (1) Headers (5) Test Results ⌚ Status: 200 OK Time: 276 ms Size: 189 B

Pretty Raw Preview Visualize JSON ⌵ ≡

```
1 {
2   "age": 22,
3   "name": "jiwoo"
4 }
```

04. RestController - GetMapping - Service

- Service와 DI 개념을 사용해서 본인 프로필 클래스를 반환하는 컨트롤러 클래스를 작성
- endpoint 정의는 `"/api/v1/profile"`

```
package hello.hellospring;

public class User {
    int age;
    String name;

    public User(int age, String name) {
        this.age = age;
        this.name = name;
    }

    public User() {
    }
}
```

```
package hello.hellospring;

public interface UserService {
    public User getUsers();
}
```

```
package hello.hellospring.service;

import hello.hellospring.User;
import hello.hellospring.UserService;
import org.springframework.stereotype.Service;

@Service
public class UserServiceImpl implements UserService {
    @Override
    public User getUsers() {
        User user = new User(22, "Hong");
        return user;
    }
}
```

```
@GetMapping("/v1/profile")
public User getParameter3(Model model){
    return userService.getUsers();
}
```

http://localhost:8080/api/v1/profile

Save

GET

http://localhost:8080/api/v1/profile

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (5) Test Results

Status: 200 OK Time: 276 ms Size: 188 B

Pretty

Raw

Preview

Visualize

JSON

≡

```
1 {
2   "age": 22,
3   "name": "Hong"
4 }
```

05. "Hello World" 리턴하는 컨트롤러를 PostMan으로 테스트 하는 대신, 테스트 코드를 사용 / 실행해서 테스트

```
@RunWith(SpringRunner.class)
@WebMvcTest(controllers = GetController.class)
class GetControllerTest {
    @Autowired
    private MockMvc mvc;

    @Test
    public void hello가_리턴된다() throws Exception{
        String hello = "Hello World!";

        mvc.perform(get("/api/v1/hello"))
            .andExpect(status().isOk())
            .andExpect(content().string(hello));
    }
}
```

BUILD SUCCESSFUL in 8s

4 actionable tasks: 1 executed, 3 up-to-date

오후 6:37:46: Task execution finished ':test --tests "hello.hellospring.controller.GetControllerTest"'.
[INFO] BUILD SUCCESS

06. 미션 4에서 진행했던 컨트롤러, 서비스 모두 테스트 대상에 포함

```
@RunWith(SpringRunner.class)
@SpringBootTest
class UserServiceImplTest {
    @Autowired
    private UserServiceImpl userServiceImpl;
    @Test
    void getUsers() {
        User user1 = userServiceImpl.getUsers();
        User user2 = new User(22, "Hong");
        assertThat(user1.getAge(), isEqualTo(user2.getAge()));
    }
}
```

```
@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
class UserControllerTest {
    @LocalServerPort
    int port = 0;

    @Autowired
    TestRestTemplate restTemplate;

    @Test
    public void userController() throws Exception {
        User user = new User(22, "Hong");
        String str = "http://localhost:" + port + "/api/v1/profile";

        User user2 = restTemplate.getForObject(str, User.class);
        assertThat(user.getAge(), isEqualTo(user2.getAge()));
    }
}
```

```
BUILD SUCCESSFUL in 9s
4 actionable tasks: 1 executed, 3 up-to-date
오후 6:36:08: Task execution finished ':test --tests "hello.hellospring.service.UserServiceImplTest.getUsers"'.

```

```
BUILD SUCCESSFUL in 10s
4 actionable tasks: 1 executed, 3 up-to-date
오후 6:36:54: Task execution finished ':test --tests "hello.hellospring.controller.UserControllerTest"'.

```