

# Chapter 3-2 Controller와 RestController

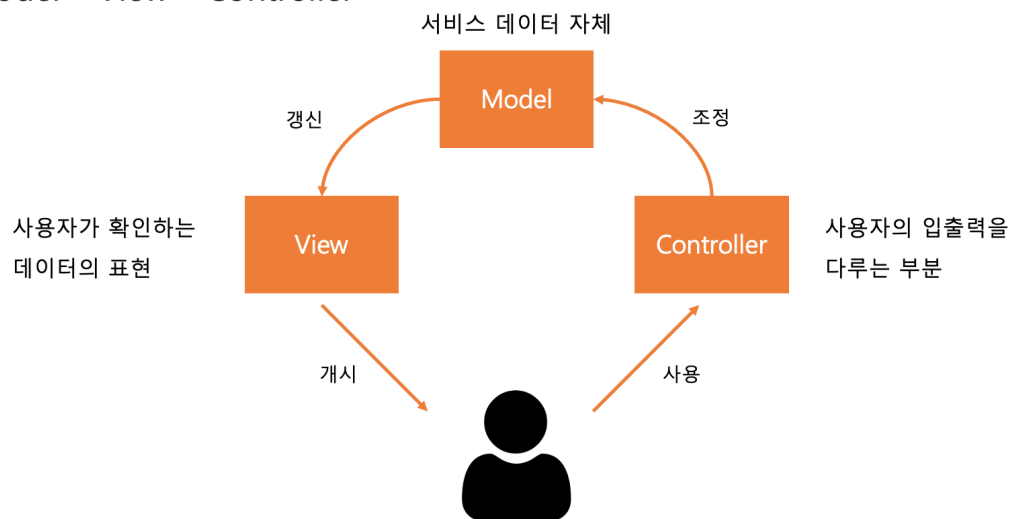
Spring MVC

Controller와 RestController 실습

Controller와 RequestMapping

## Spring MVC

### Model - View - Controller



Model : 서비스 데이터 자체

MVC 패턴의 작동 방식

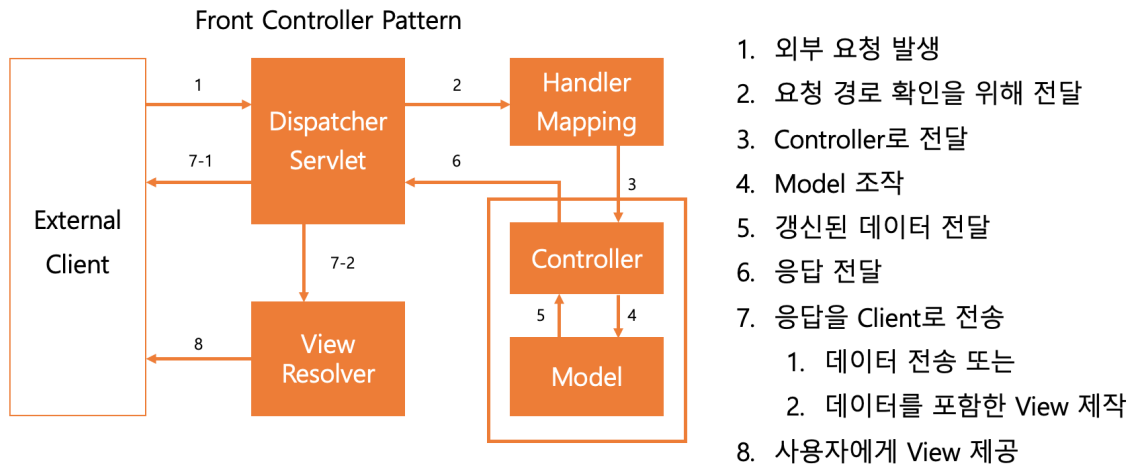
사용자가 View에서 애플리케이션의 작동 모습을 보게 됨

자신이 할 수 있는 행동들을 Controller를 사용해 전달

Controller는 데이터를 조작(추가, 제거, 상태 변화 등)함

Model은 Controller에서 지시받은 걸 기준으로 View를 갱신함

- 코드를 어떠한 형태로 구성할 것인지 생각하는 ..!



실제 Spring MVC의 모습(?)

앞 쪽은 대부분 자동으로 되고(?)

Spring Boot로 조작하는 부분은 Controller, Model 부분

## Controller와 RestController 실습

```

package dev.aquashdw.controllerdemo;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;

@Controller
public class SampleController {
    private static final Logger logger
        = LoggerFactory.getLogger(SampleController.class);

    @RequestMapping(
        value = "/hello",
        method = RequestMethod.GET
    )
    public String hello(@RequestParam(
        name = "id", required = false, defaultValue = "") String id
    ){
        logger.info("Path: Hello");
        logger.info("Query Param id: " + id);
        return "/hello.html";
    }
}

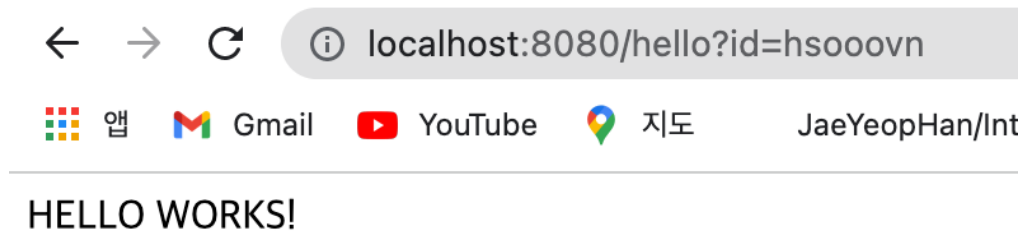
```

```

@GetMapping(
    value = "/hello/{id}"
)
public String helloPath(@PathVariable("id") String id){
    logger.info("Path Variable is: " + id);
    return "/hello.html";
}
}

```

쿼리 확인

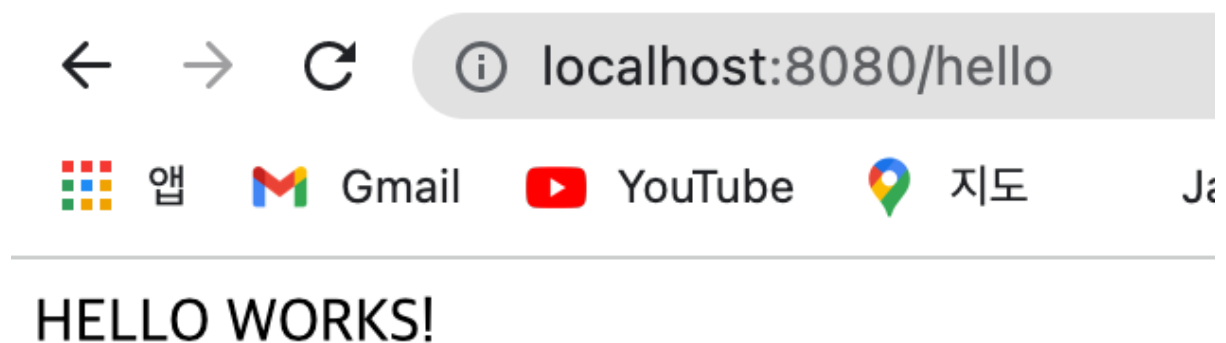


```

-8080-exec-2] d.a.controllerdemo.SampleController : Path: Hello
-8080-exec-2] d.a.controllerdemo.SampleController : Query Param id: hsooovn

```

id 없어도 괜찮음 !



```

-5] d.a.controllerdemo.SampleController : Path: Hello
-5] d.a.controllerdemo.SampleController : Query Param id:

```

## 함수 추가

```
@GetMapping(
    "/get-profile"
)
public @ResponseBody SamplePayload getProfile(){
    return new SamplePayload("hsooovn", 24, "Developer");
}
```

## SamplePayload.java 생성

```
package dev.aquashdw.controllerdemo;

public class SamplePayload {
    private String name;
    private int age;
    private String Occupation;

    public SamplePayload() {
    }

    public SamplePayload(String name, int age, String occupation) {
        this.name = name;
        this.age = age;
        Occupation = occupation;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getOccupation() {
        return Occupation;
    }

    public void setOccupation(String occupation) {
        Occupation = occupation;
    }

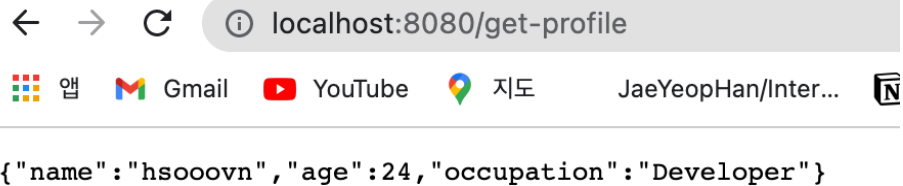
    @Override
    public String toString() {
        return "SamplePayload{" +
            "name='" + name + '\'' +
            ", age=" + age +

```

```

        ", Occupation='" + Occupation + '\'' +
        '}}';
    }
}

```



← → ↻ ⓘ localhost:8080/get-profile

앱 Gmail YouTube 지도 JaeYeopHan/Inter...

```

{"name": "hsooovn", "age": 24, "occupation": "Developer"}

```

ResponseBody를 사용함으로써 이것 자체로 데이터를 Body로 !

### SampleRestController.java 생성

```

package dev.aquashdw.controllerdemo;

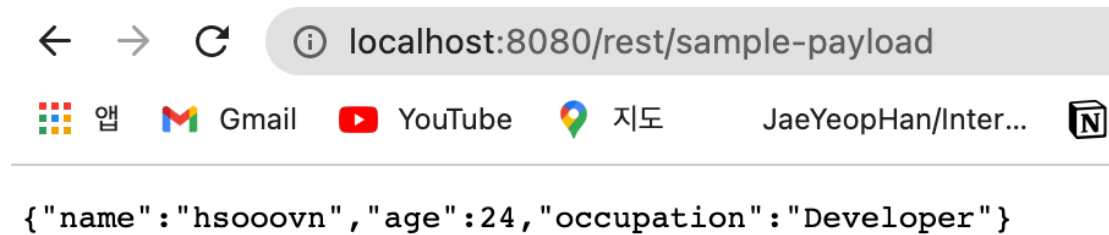
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/rest")
public class SampleRestController {
    private static final Logger logger = LoggerFactory.getLogger(SampleRestController.class);

    @GetMapping("/sample-payload")
    public SamplePayload samplePayload(){
        return new SamplePayload("hsooovn", 24, "Developer");
    }
}

```

동일한 결과



Controller는 뷰나 데이터를 기본적으로 제공하는 좀 더 넓은 의미 ??

그래서 ResponseBody를 명확히 표시해줘야 원하는 곳에 데이터를 보냄

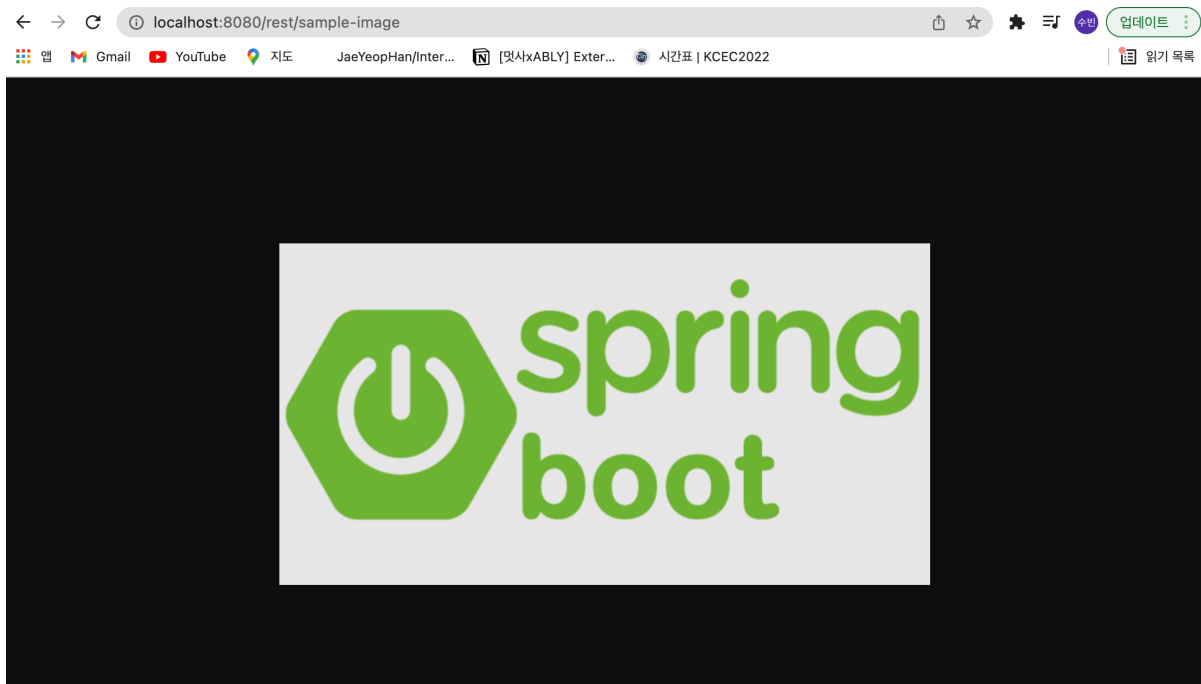
RestController는 데이터를 주고 받는데 주력하기 때문에 ResponseBody를 하지 않아도 자동으로 붙은 것과 같은 효과가 있음

함수 추가

```
@GetMapping(
    value = "/sample-image",
    produces = MediaType.IMAGE_PNG_VALUE
)
public byte[] sampleImage() throws IOException {
    InputStream inputStream = getClass().getResourceAsStream("/static/img.png");

    return inputStream.readAllBytes();
}
```

이미지든 동영상이든 byte를 돌려준다 ?



## Controller와 RequestMapping

```
@Controller  
public class SampleController {
```

요청 경로를 설정하기 위해  
Controller Annotation을  
사용하자.

```
    @RequestMapping("/profile")  
    public String profile(){  
        logger.info("in profile");  
        return "profile.html";  
    }  
}
```

RequestMapping을 이용해  
경로에 따라 실행될 함수를  
지정할 수 있다.

```

@GetMapping("/sample-profile")
public String sampleProfile(){
    logger.info("in sample profile");
    return "sample/profile.html";
}

```

Method 별로 별도의 Annotation이 존재한다.  
(GetMapping, PostMapping 등)

```

@GetMapping(
    value = "/sample-payload",
    produces = MediaType.APPLICATION_JSON_VALUE
)
public @ResponseBody SamplePayload samplePayload(){
    return new SamplePayload(

```

HTML 외에 데이터 전송을 위해 Body와  
MediaType을 설정할 수 있다.

```

@RestController
@RequestMapping("rest")
public class SampleRestController {
    private static final Logger logger = L

    @GetMapping(
        value = "sample-payload"
    )
    public SamplePayload samplePayload(){

```

RestController  
간단하게, Controller + ResponseBody



```
@GetMapping(  
    value = "sample-image",  
    produces = MediaType.IMAGE_PNG_VALUE  
)  
public byte[] sampleImage() throws IOException {
```

기본적으로 어떠한 형태의 응답이든 데이터의 일종이다.