섹션 2. 스프링 웹 개발 기초

Created Time	@2022년 5월 22일 오후 6:26
Updated Time	@2022년 5월 28일 오후 3:50
날짜	@2022년 5월 23일

@eeeyooon

강의 수강 날짜 : 05.23

섹션2. 스프링 웹 개발 기초

정적 컨텐츠

: 파일을 그대로 웹 브라우저에 내려주는거 (웰컴페이지처럼)

MVC와 템플릿 엔진

-JSP, PHP > 템플릿 엔진 (HTML을 서버에서 프로그래밍해서 브라우저로 보내는것 = 동적으로 바꿔서)

-컨트롤러, 모델, 템플릿 엔진 화면 > 이 세가지를 모델뷰컨트롤러라고 해서 MVC라고 함.

-정적 콘텐츠는 파일을 그대로 전달, MVC와 템플릿엔진은 서버에서 변형을 해서 HTML을 바꿔서 전달하는 방식

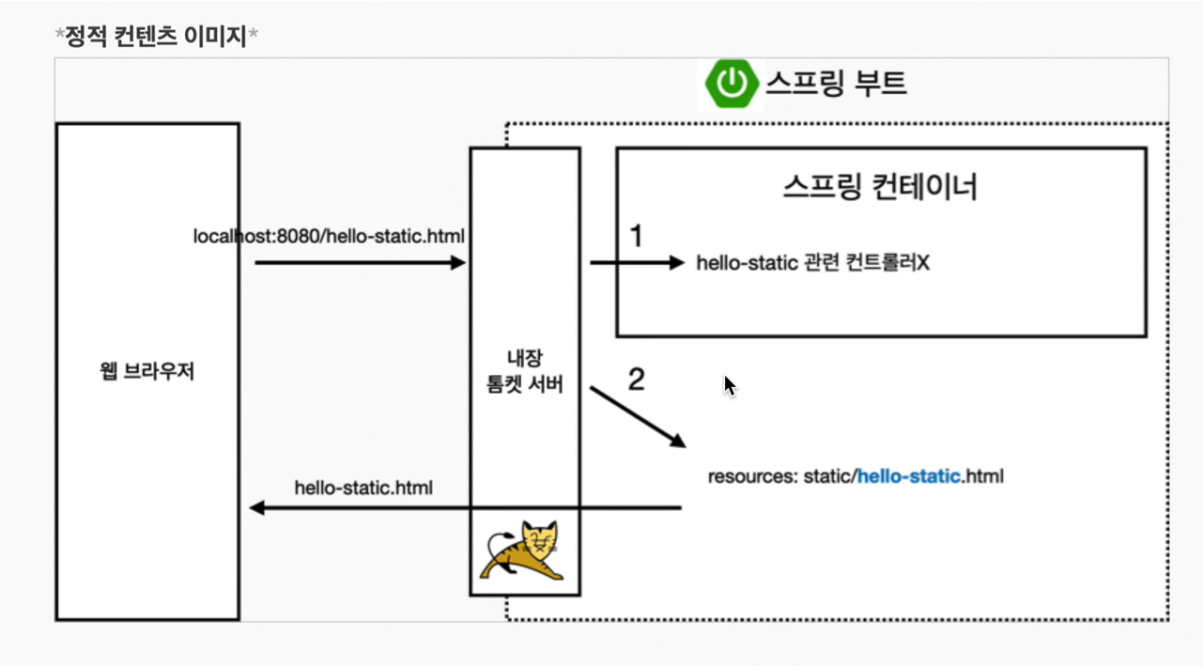
API

-요즘엔 json 데이터 구조 포맷으로 클라이언트한테 데이터를 전달함. > 이걸 API 방식이라고 함.

정적 컨텐츠

resources/static/hello-static.html

```
<!DOCTYPE HTML>
<html>
<head>
    <title>static content</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
정적 컨텐츠 입니다.
</body>
</html>
```



MVC와 템플릿 엔진

- MVC : Model, View, Controller
- 원래는 뷰와 컨트롤러를 구분하지 않았지만 뷰는 화면을 그리는데, 컨트롤러는 비지니스 로직이나 서버를 조작하는데 집중해야함. (본인 역할에) → 뷰와 컨트롤러를 구분하는건 이제 기본

Controller

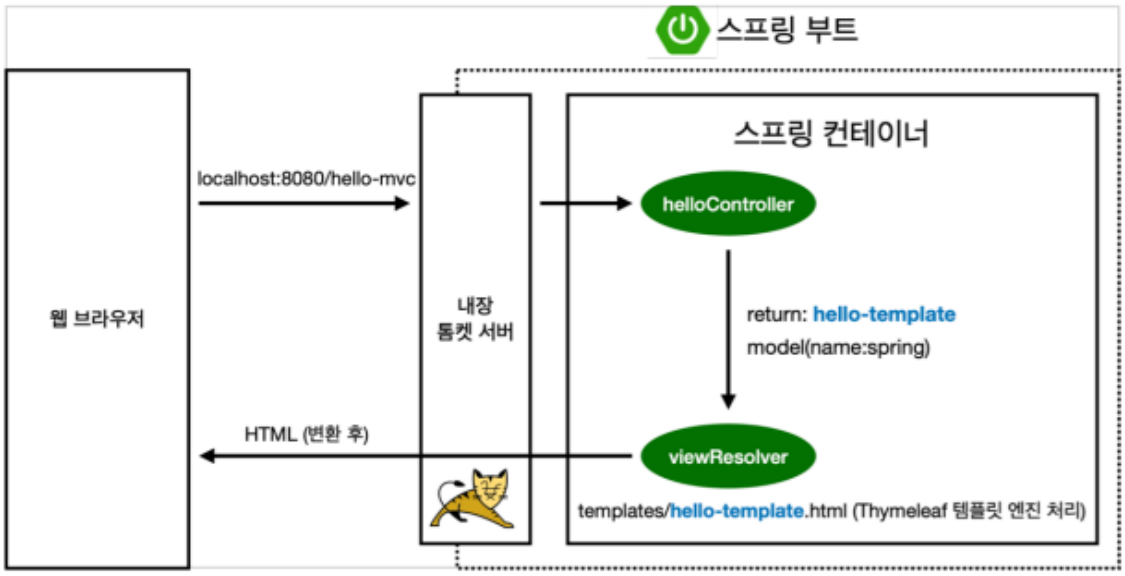
```
@Controller
public class HelloController {
 @GetMapping("hello-mvc")
 public String helloMvc(@RequestParam("name") String name, Model model) {
 model.addAttribute("name", name);
 return "hello-template";
 }
}
```

View

```
<html xmlns:th="http://www.thymeleaf.org">
<body>
<p th:text="'hello ' + ${name}">hello! empty</p>
</body>
</html>
```

http://localhost:9090/hello-mvc?name=spring 실행 과정

MVC, 템플릿 엔진 이미지



- 웹 브라우저에서 localhost:9090에 hello-mvc를 넘김 → 내장 톰캣 서버에서 hello-mvc를 스프링에게 넘김 → 스프링에서는 얜 helloController에 매핑이 되어있네? 하고 그 메소드를 호출해줌 → return해줄때 hello-template으로 → viewResolver가 templates/hello-template.html, 리턴의 스트링 네임과 똑같은 애를 찾아서 템플릿 엔진이 랜더링을 해서 변환을 한 HTML을 웹 브라우저로 보냄.

API

-HTML로 내리냐, API방식으로 내리냐의 차이

@ResopnseBody 객체 반환

```
@Controller
public class HelloController {
 @GetMapping("hello-api")
 @ResponseBody
 public Hello helloApi(@RequestParam("name") String name) {
 Hello hello = new Hello();
 hello.setName(name);
```

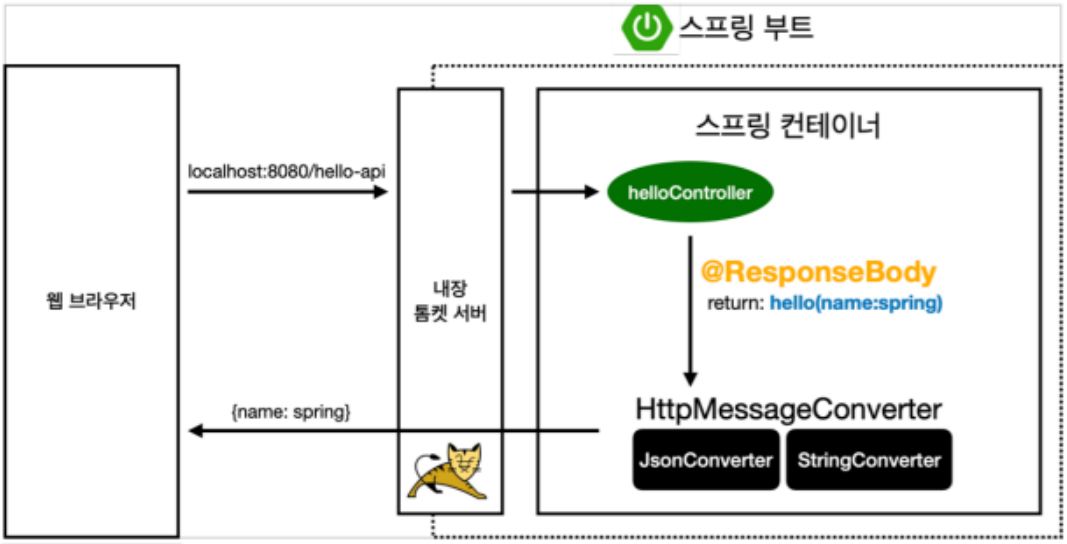
```
  return hello;
  }
  static class Hello {
  private String name;
  public String getName() {
  return name;
  }
  public void setName(String name) {
  this.name = name;
  }
  }
 }
```

- @ResopnsBody를 사용하고, 객체를 반환하면 객체가 JSON으로 변환됨.
- json 형식 찾아보기 {key:value}
- xml 방식은 되게 무겁고 열고 닫고 두번 해야함. json은 심플함

  → 최근에는 거의 json방식으로 통일 (기본으로 json반환하는걸로 세팅하는게 좋음)

실행 : http://localhost:9090/hello-api?name-spring

05/28 이어서 들음.



- `@ResponseBody` 를 사용
  - HTTP의 BODY에 문자 내용을 직접 반환
  - `viewResolver` 대신에 `HttpMessageConverter` 가 동작
  - 기본 문자처리: `StringHttpMessageConverter`
  - 기본 객체처리: `MappingJackson2HttpMessageConverter`
  - byte 처리 등등 기타 여러 HttpMessageConverter가 기본으로 등록되어 있음

> 참고: 클라이언트의 HTTP Accept 해더와 서버의 컨트롤러 반환 타입 정보 둘을 조합해서 `HttpMessageConverter` 가 선택된다. 더 자세한 내용은 스프링 MVC 강의에서 설명하겠다.

- 웹브라우저에서 localhost:9090/hello-api 입력하면 내장 톰켓 서버에서 왔다가 스프링 알려줌.
- 스프링은 hello-api를 찾는데, @ResponseBody가 있으면 http 응답에 그걸 그대로 전달함. 근데 얘가 문자가 아니고 객체 (문자면 그대로 줬음). 객체를 받으면 기본(디폴트)이 json 방식으로 데이터를 만들어서 htpp 응답에 반환하겠다는게 기본정책임.
- @ResponseBody라고 오면 hello 객체를 넘기고 HttpMessageConverter가 동작을 함. 얘가 전달받은게 단순 문자면 StringConverter를 동작하고 객체면 JsonConverter를 동작됨. 그 다음 이 객체를 json방식으로 바꿈. 바꾼 json을 요청한 웹브라우저나 서버에 보내줌.
- @ResponseBody 원래 이런게 안붙어있으면 템플릿에선 viewResolver에 던져서 맞는 템플릿 달라고 함. (이전 수업 내용 mvc 템플릿 엔진 이미지.png)

- @ResponseBody를 사용하면 HTTP의 BODY에 문자 내용을 직접 반환함. 이때 객체라면 MappingJackson2HttpMessageConverter가 동작하여 json방식으로 바꿈.

*객체를 json으로 바꿔주는 대표적인 라이브러리 : Jackson 라이브러리 > 스프링은 기본적으로 탑재

요약

1. 정적 컨텐츠
    - 파일을 그대로 내려준다.
2. MVC와 템플릿 엔진
    - 템플릿 엔진을 모델뷰 컨트롤러 방식으로 쪼개서 뷰를 템플릿 엔진으로 html을 좀 더 랜더링해서 랜더링이 된 html을 클라이언트에게 전달해준다.
3. API
    - API 방식은 객체를 반환하는 것.
    - HttpMessageConverter로 객체를 json 방식으로 반환함.