

Back-End Study

Spring REST Docs

2020. 8. 15

백엔드팀 서동현

01

Spring REST Docs

RESTful 서비스에 대한 문서를 생성해주는데 도움을 주는 도구

기본적으로 AsciiDoc을 사용하며 테스트 코드를 통해 html 파일을 생성

Swagger vs Spring REST Docs

API를 테스트 해 볼 수 있는 화면을 제공
적용하기 쉽다

제품 코드에 영향은 없지만 가독성이 떨어짐
애노테이션은 주석일뿐이라서 최신화가 안됨

제품 코드에 영향이 없다
테스트 코드가 성공해야 문서가 생성

적용하기 어렵다

```
@Api(description = "약", tags = {"drug"})
@RestController
@RequestMapping(value = "/products")
public class ProductController {

    @Autowired
    private ProductService productService;

    @ApiOperation(value = "약 검색", notes = "약을 검색한다.")
    @ApiResponses(value = {
        @ApiResponse(code = 200, message = "Success"),
        @ApiResponse(code = 400, message = "Bad Request"),
        @ApiResponse(code = 500, message = "Server Error")
    })
    @GetMapping(value = "")
    public ProductLiteDto getProductsByNameAndCompany(@RequestParam String name,
        @RequestParam(required = false) String company, @RequestParam int pageNo) {
        return productService.nameListByNameAndCompany(name, company, pageNo);
    }

    @ApiOperation(value = "약 정보 상세 조회", notes = "약 정보 상세를 조회한다.")
    @GetMapping(value = "{id}")
    public Product getProductById(@PathVariable int id) {
        return productService.getById(id);
    }
}
```

Yakgguk^{1.0}

[Base URL: localhost:8080/]
<http://localhost:8080/v2/api-docs>

2018 하반기 Mash-Up 프로젝트

[Apache License Version 2.0](#)

barcode 바코드

POST /barcodes 바코드 번호, 제품 데이터 저장

GET /barcodes/{barcodeNumber} 바코드 번호로 검색

drug 약

GET /products 약 검색

GET /products/{id} 약 정보 상세 조회

barcode 바코드

POST /barcodes 바코드 번호, 제품 데이터 저장

입력한 바코드 번호와 선택한 제품의 데이터를 저장한다.

Parameters

Cancel

| Name | Description |
|------------------------------------------------------------|------------------------------------------------|
| barcodeNumber * required string (query) | barcodeNumber barcodeNumber - barcodeNumber |
| productId * required integer(\$int32) (query) | productId productId - productId |

Execute

Responses

Response content type

/

| Code | Description |
|------|-----------------------|
| 200 | OK |
| 500 | Internal Server Error |

Spring REST Docs를 사용하기 위한 최소 요구사항

- Java 8
- Spring Framework 5 (5.0.2 or later)

```
plugins {
    id 'org.asciidoctor.convert' version '1.5.9.2'
}

ext {
    snippetsDir = file('build/generated-snippets')
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-web'

    testImplementation('org.springframework.boot:spring-boot-starter-test') {
        exclude group: 'org.junit.vintage', module: 'junit-vintage-engine'
    }

    testCompile 'org.springframework.restdocs:spring-restdocs-mockmvc:2.0.4.RELEASE'
    asciidoctor 'org.springframework.restdocs:spring-restdocs-asciidoctor:2.0.4.RELEASE'
}

test {
    outputs.dir snippetsDir
    useJUnitPlatform()
}

asciidoctor {
    inputs.dir snippetsDir
    dependsOn test
}

bootJar {
    dependsOn asciidoctor
    from ("${asciidoctor.outputDir}/html5") {
        into 'static/docs'
    }
}
```



```
plugins {  
    id 'org.asciidoctor.convert' version '1.5.9.2'  
}
```

AsciiDoc 파일을 컨버팅하고 Build 폴더에 복사하기 위한 플러그인

```
test {  
    outputs.dir snippetsDir  
    useJUnitPlatform()  
}  
  
asciidoctor {  
    inputs.dir snippetsDir  
    dependsOn test  
}  
  
bootJar {  
    dependsOn asciidoctor  
    from ("${asciidoctor.outputDir}/html5") {  
        into 'static/docs'  
    }  
}
```

Grade build를 하면 test -> asciidoctor -> bootJar 순으로 실행이 되도록 설정
(JUnit5부터 test에 useJUnitPlatform()을 적어줘야 정상 작동함!!)

```
@RequiredArgsConstructor
@RestController
@RequestMapping("/api/styles")
public class StyleController {

    private final StyleService styleService;

    @GetMapping
    public ResponseEntity<OtdResponse<List<StyleResponse>>> list() {
        List<StyleResponse> response = styleService.list();

        return ResponseEntity
            .status(HttpStatus.OK)
            .body(OtdResponse.<List<StyleResponse>>builder()
                .code(HttpStatus.OK.value())
                .msg("스타일 정보 반환")
                .data(response)
                .build());
    }
}
```

```
@ExtendWith(SpringExtension.class)
@WebMvcTest(StyleController.class)
@AutoConfigureRestDocs
@Import(JsonConfig.class)
public class StyleControllerTest {
```

Junit5부터 @RunWith(SpringRunner.class) 대신 @ExtendWith(SpringExtension.class) 사용

RESTDocs를 사용하려면 @AutoConfigureRestDocs 애노테이션을 달아줘야 함

```
BDDMockito.given(styleService.list()).willReturn(response);

// when
ResultActions result = mockMvc.perform(get("/api/styles")
    .contentType(MediaType.APPLICATION_JSON)
    .accept(MediaType.APPLICATION_JSON));

// then
result.andExpect(status().isOk())
    .andDo(document("style-list",
        getDocumentRequest(),
        getDocumentResponse(),
        responseFields(
            fieldWithPath("code").type(JsonFieldType.NUMBER).description("상태 코드"),
            fieldWithPath("msg").type(JsonFieldType.STRING).description("상태 메시지"),
            fieldWithPath("data[].id").type(JsonFieldType.NUMBER).description("id"),
            fieldWithPath("data[].name").type(JsonFieldType.STRING).description("이름")
        )
    ));
```

responseFields에 response 값들에 대한 문서 정의를 적어주면 된다.

```
GET /api/styles HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: localhost:8080
```

HTTP

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 159

{
  "code" : 200,
  "msg" : "스타일 정보 반환",
  "data" : [ {
    "id" : 1,
    "name" : "댄디"
  }, {
    "id" : 2,
    "name" : "캐주얼"
  } ]
}
```

| Path | Type | Description |
|-------------|--------|-------------|
| code | Number | 상태 코드 |
| msg | String | 상태 메시지 |
| data[].id | Number | id |
| data[].name | String | 이름 |

```
@RequiredArgsConstructor
@RestController
@RequestMapping("/api/posts")
public class PostController {

    private final PostService postService;

    @PostMapping
    public ResponseEntity<OtdResponse<Void>> create(PostCreateRequest dto) {
        postService.create(dto);

        return ResponseEntity
            .status(HttpStatus.CREATED)
            .body(OtdResponse.<Void>builder()
                .code(HttpStatus.CREATED.value())
                .msg("업로드 성공")
                .build());
    }
}
```

```
@Test
public void test_create() throws Exception {
    ResultActions result = mockMvc.perform(fileUpload("/api/posts")
        .file("uploadFile", "image".getBytes())
        .param("userId", "1")
        .param("message", "메세지")
        .param("address", "주소")
        .param("weather", "날씨")
        .param("temperature", "온도")
        .param("styleIds", "1,2,3"))
        .andExpect(status().isCreated());

    result.andDo(print())
        .andExpect(status().isCreated())
        .andDo(document("upload",
            getDocumentResponse(),
            requestParts(
                partWithName("uploadFile").description("업로드 할 이미지")
            ),
            requestParameters(
                parameterWithName("userId").description("유저 id 입력"),
                parameterWithName("message").description("주소 입력"),
                parameterWithName("address").description("주소 입력"),
                parameterWithName("weather").description("날씨 입력"),
                parameterWithName("temperature").description("온도 입력"),
                parameterWithName("styleIds").description("스타일 id 입력")
            )
        ));
}
```

requestParts

- multipart 타입에 해당하는 값 정의

requestParameters

- request에 해당하는 파라미터들 정의

| Part | Description |
|------------|-------------|
| uploadFile | 업로드 할 이미지 |

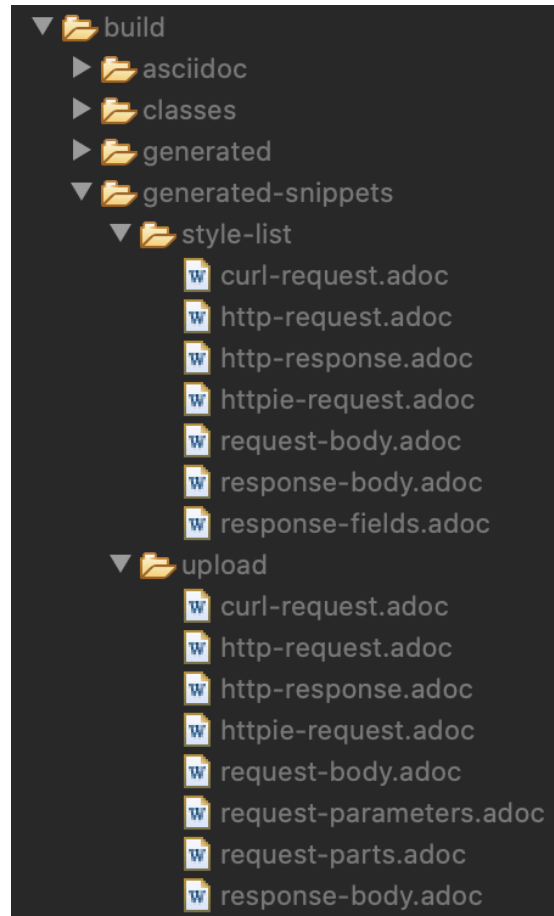
| Parameter | Description |
|-------------|-------------|
| userId | 유저 id 입력 |
| message | 주소 입력 |
| address | 주소 입력 |
| weather | 날씨 입력 |
| temperature | 온도 입력 |
| styleIds | 스타일 id 입력 |

Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 48
```

HTTP

```
{
  "code" : 201,
  "msg" : "업로드 성공"
}
```

grade build를 하면 build/generated-snippets에 adoc파일들이 생성된다.
(maven, gradle에 따라서 다르다.)

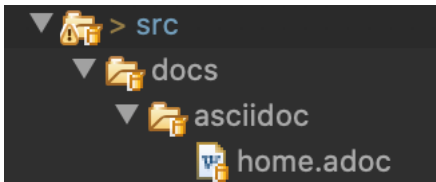
만들어진 snippets을 사용하려면 이걸 가져다 쓰는 곳이 있어야 한다.

gradle의 경우 (maven의 경우 다름)

- source 파일 : src/docs/asciidoc/*.adoc
- generated 파일 : build/asciidoc/html5/*.html

adoc파일의 이름과 동일한 이름으로 html파일이 생성된다.

Spring REST Docs



```
= 00TD API Document
:doctype: book
:icons: font
:source-highlighter: highlightjs
:toc: left
:toclevels: 3
:sectlinks:

[[Post]]
== 파일 업로드

=== Request

include::{snippets}/upload/curl-request.adoc[]
include::{snippets}/upload/http-request.adoc[]
include::{snippets}/upload/httpie-request.adoc[]
include::{snippets}/upload/request-parts.adoc[]
include::{snippets}/upload/request-parameters.adoc[]

=== Response

include::{snippets}/upload/http-response.adoc[]

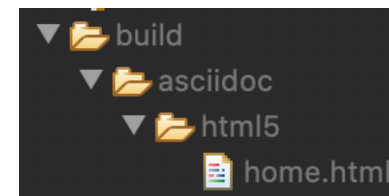
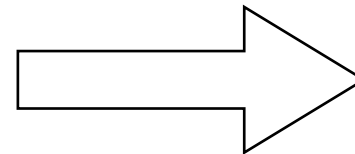
[[Style]]
== 스타일 정보

=== Request

include::{snippets}/style-list/curl-request.adoc[]
include::{snippets}/style-list/http-request.adoc[]

=== Response

include::{snippets}/style-list/http-response.adoc[]
include::{snippets}/style-list/response-fields.adoc[]
```



```
libs — zsh — 148x28
-rw-r--r--  2.0 unx  237826 b- stor 20-Jul-14 19:56 BOOT-INF/lib/jakarta.el-3.0.3.jar
-rw-r--r--  2.0 unx 1035660 b- stor 20-Jul-20 23:53 BOOT-INF/lib/jaxb-runtime-2.3.3.jar
-rw-r--r--  2.0 unx   71976 b- stor 20-Jul-20 23:53 BOOT-INF/lib/txw2-2.3.3.jar
-rw-r--r--  2.0 unx  328593 b- stor 20-Jul-14 19:55 BOOT-INF/lib/httpcore-4.4.13.jar
-rw-r--r--  2.0 unx   68453 b- stor 20-Jul-20 23:53 BOOT-INF/lib/jakarta.activation-1.2.2.jar
-rw-r--r--  2.0 unx   25058 b- stor 20-Jan-13 23:18 BOOT-INF/lib/jakarta.annotation-api-1.3.5.jar
-rw-r--r--  2.0 unx  115638 b- stor 20-Jul-14 19:56 BOOT-INF/lib/jakarta.xml.bind-api-2.3.3.jar
-rw-r--r--  2.0 unx   76204 b- stor 20-Jul-20 23:53 BOOT-INF/lib/hibernate-commons-annotations-5.1.0.Final.jar
-rw-r--r--  2.0 unx   60790 b- stor 20-Jul-15 22:34 BOOT-INF/lib/jboss-logging-3.4.1.Final.jar
-rw-r--r--  2.0 unx  471901 b- stor 20-Jan-04 15:46 BOOT-INF/lib/logback-core-1.2.3.jar
-rw-r--r--  2.0 unx   309001 b- stor 20-Jul-14 19:56 BOOT-INF/lib/snakeyaml-1.26.jar
-rw-r--r--  2.0 unx  268491 b- stor 20-Jul-14 19:56 BOOT-INF/lib/tomcat-embed-websocket-9.0.36.jar
-rw-r--r--  2.0 unx  3373581 b- stor 20-Jul-14 19:56 BOOT-INF/lib/tomcat-embed-core-9.0.36.jar
-rw-r--r--  2.0 unx   777669 b- stor 20-Jul-20 23:53 BOOT-INF/lib/javassist-3.24.0-GA.jar
-rw-r--r--  2.0 unx  195684 b- stor 20-Jul-20 23:53 BOOT-INF/lib/jandex-2.1.3.Final.jar
-rw-r--r--  2.0 unx   323630 b- stor 20-Jul-20 23:53 BOOT-INF/lib/dom4j-2.1.3.jar
-rw-r--r--  2.0 unx   29877 b- stor 20-Jul-20 23:53 BOOT-INF/lib/istack-commons-runtime-3.0.11.jar
-rw-r--r--  2.0 unx   565410 b- stor 20-Jul-14 22:00 BOOT-INF/lib/ion-java-1.0.2.jar
-rw-r--r--  2.0 unx   621931 b- stor 20-Jul-14 23:34 BOOT-INF/lib/joda-time-2.8.1.jar
-rwxr-xr-x  2.0 unx      0 bl defN 20-Aug-04 01:02 static/
-rw-r--r--  2.0 unx   39036 bl defN 20-Aug-04 00:00 static/docs/home.html
-rwxr-xr-x  2.0 unx      0 bl defN 20-Jul-20 00:00 BOOT-INF/classes/schema/
-rwxr-xr-x  2.0 unx      0 bl defN 20-Jul-24 01:03 BOOT-INF/classes/static/
-rwxr-xr-x  2.0 unx      0 bl defN 20-Jul-24 01:03 BOOT-INF/classes/templates/
-rw-r--r--  2.0 unx   2642 bl defN 20-Aug-04 01:02 BOOT-INF/classpath.idx
213 files, 47684856 bytes uncompressed, 47478396 bytes compressed: 0.4%
sdh@Appleui-MacBookPro libs %
```

Create new page

Title

Write

Preview

h1

h2

h3

B

i

<>

?

Edit mode:

✓ AsciiDoc

Creole

Markdown

MediaWiki

Org-mode

Pod

RDoc

Textile

reStructuredText

Github wiki에서도
AsciiDoc형식의 문서를 만들수 있음

THANKS FOR WATCHING

감사합니다

2020.8. 15

백엔드팀 서동현