Swagger

- 강경미 (carami@nate.com)

Swagger란?

Swagger란 서버로 요청되는 URL 리스트를 HTML화면으로 문서화 및 테스트 할 수 있는 라이브러리이다. 간단하게 설명하면 Swagger는 API Spec 문서이다. API를 엑셀이나 가이드 문서를 통해 관리하는 방법은 주기적인 업데이트가 필요하기 때문에 관리가 쉽지 않고 시간이 오래 걸린다. 그래서 Swagger를 사용해 API Spec 문서를 자동화해주어 간편하게 API문서를 관리하면서 테스트할 수 있다.

Swagger 설정하기

```
        <dependency>
            <groupId>io.springfox</groupId>
            <artifactId>springfox-swagger2</artifactId>
            <version>2.9.2</version>
        </dependency>
         <dependency>
            <groupId>io.springfox</groupId>
            <artifactId>springfox-swagger-ui</artifactId>
            <version>2.9.2</version>
        </dependency>
```

Swagger 설정하기

```
@Configuration
@EnableSwagger2
@SuppressWarnings("unchecked")  // warning밑줄 제거를 위한 태그
public class SwaggerConfig {


    @Bean
    public Docket todosApi() {
        return getDocket("todos", Predicates.or(
                PathSelectors.regex("/api/todos/.*")));
    }
```

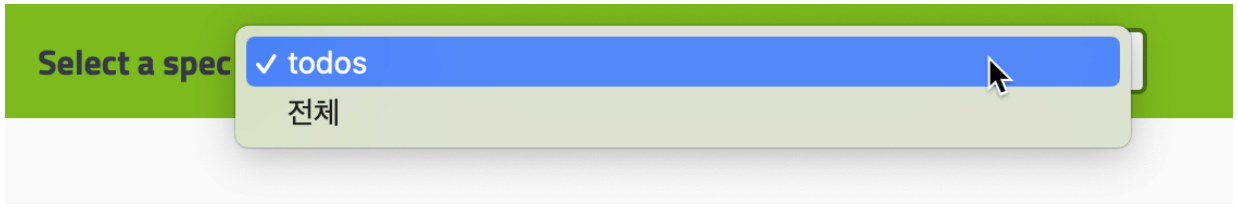
```
    @Bean
    public Docket allApi() {
        return getDocket("전체", Predicates.or(
                        PathSelectors.any()));
    }


    //swagger 설정.
    public Docket getDocket(String groupName, Predicate<String> predicate) {
        return new Docket(DocumentationType.SWAGGER_2)
                        .groupName(groupName)
                .select()
                .apis(RequestHandlerSelectors.basePackage("examples.springmvc"))
                .paths(predicate)
                .apis(RequestHandlerSelectors.any())
                .build();
    }
}
```

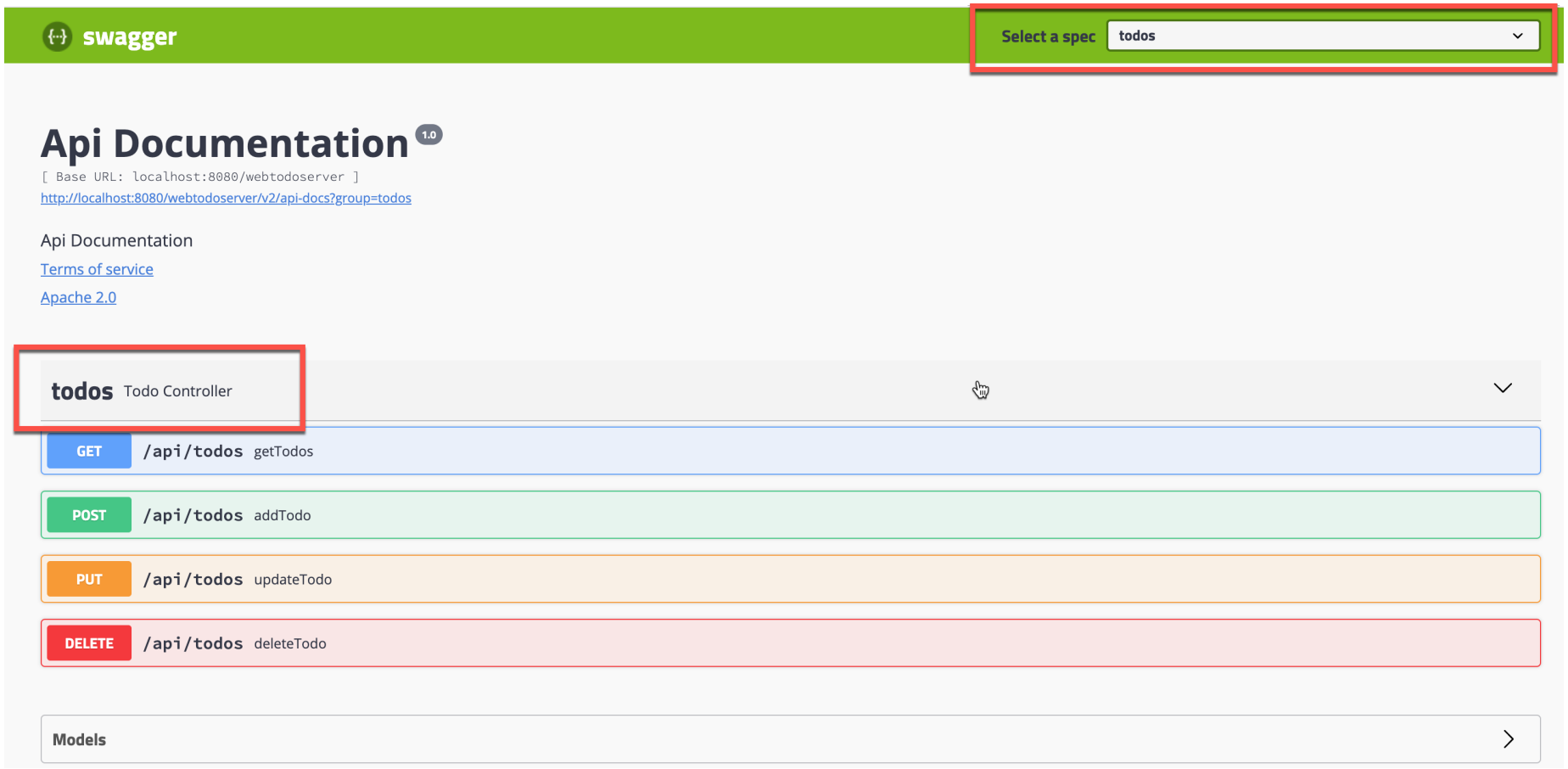
Swagger Config 설명

- .consume()과 .produces()는 각각 Request Content-Type, Response Content-Type에 대한 설정. (선택)
- .apiInfo()는 Swagger API 문서에 대한 설명을 표기하는 메소드. (선택)
- .apis()는 Swagger API 문서로 만들기 원하는 BasePackage 경로. (필수)
- .path()는 URL 경로를 지정하여 해당 URL에 해당하는 요청만 Swagger API 문서로 만든다. (필수)
- /api/todos/.* : 정규표현식 . 은 문자하나를 말한다. .*은 어떤 문자든 올 수 있다는 것을 의미한다.
    - /api/todos/100 , api/todos/list 등

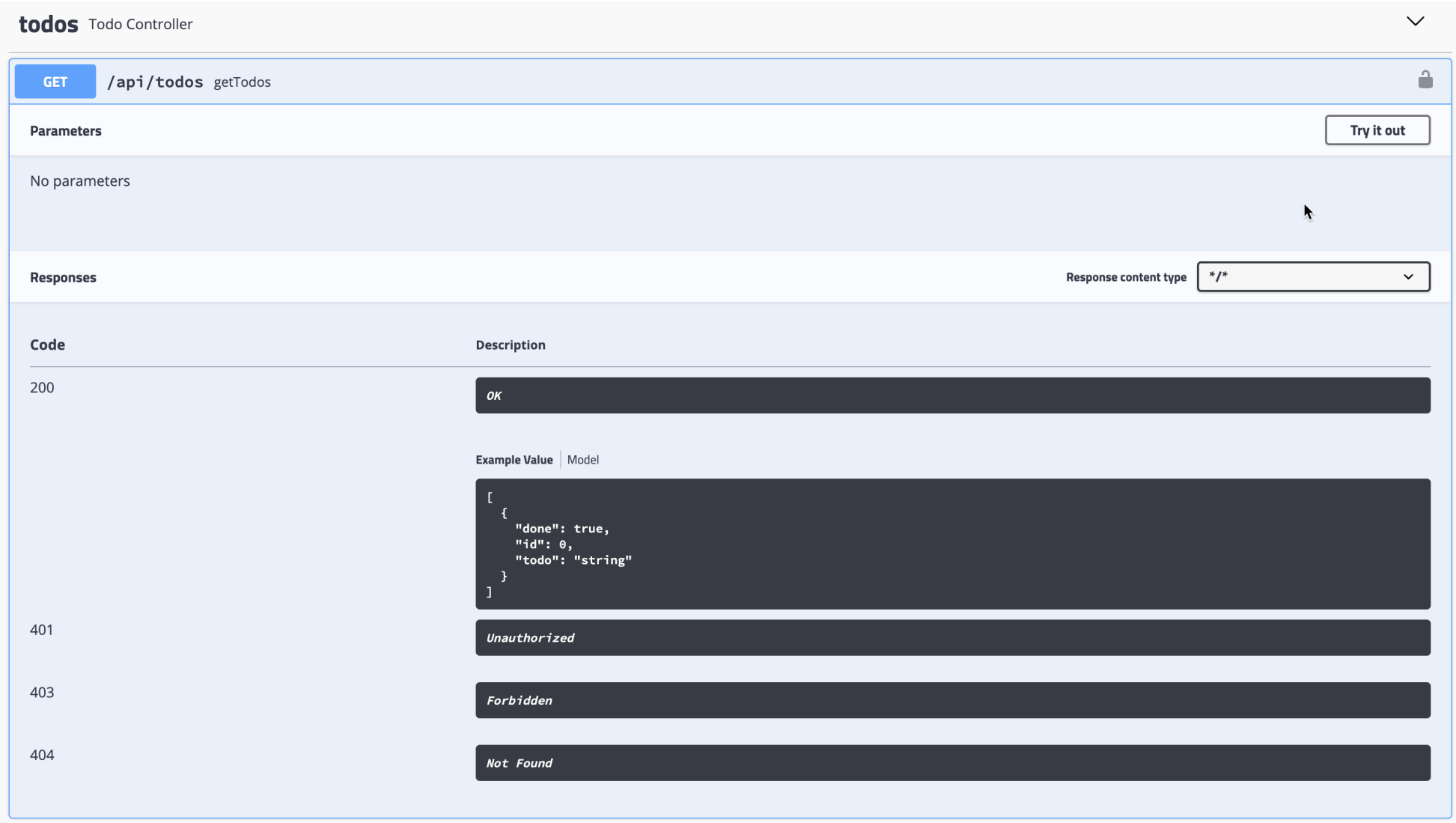
Swagger 확인하기

http://localhost:8080/webtodoserver/swagger-ui.html#/todo-controller

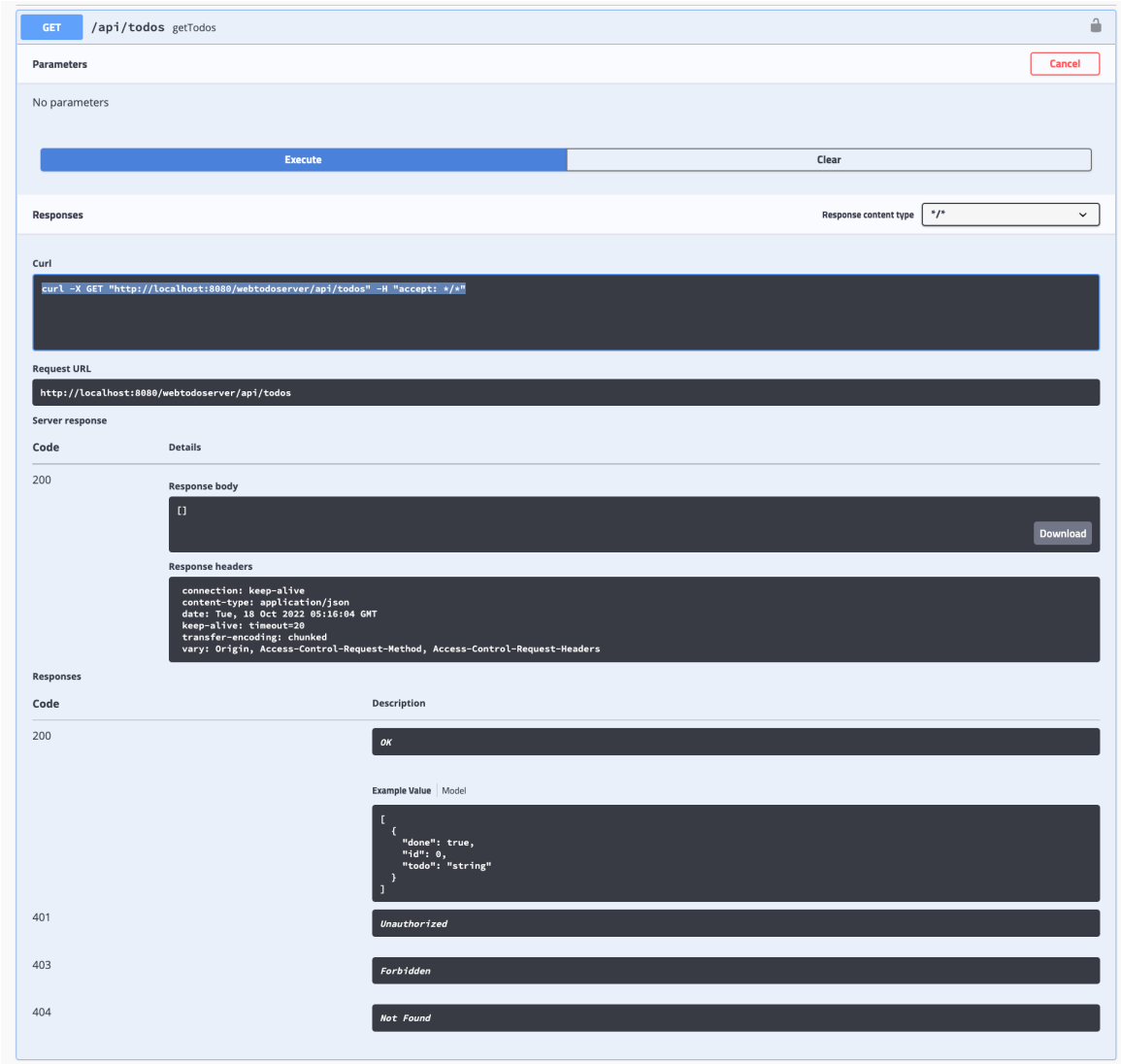




- 실행시키고 싶은 API를 선택하고 "Try it out" 버튼을 클릭한다.
- Execute 버튼을 클릭한다.



- 결과가 출력된다.


GET
/api/todos getTodos
Parameters
Cancel
No parameters
Execute
Clear
Responses
Response content type
/
Curl
curl -X GET "http://localhost:8080/webtodoserver/api/todos" -H "accept: */*"
Request URL
http://localhost:8080/webtodoserver/api/todos
Server response
Code
Details
200
Response body
[]
Download
Response headers
connection: keep-alive
content-type: application/json
date: Tue, 18 Oct 2022 05:16:04 GMT
keep-alive: timeout=20
transfer-encoding: chunked
vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
Responses
Code
Description
200
OK
Example Value
Model
[
{
"done": true,
"id": 0,
"todo": "string"
}
]
401
Unauthorized
403
Forbidden
404
Not Found

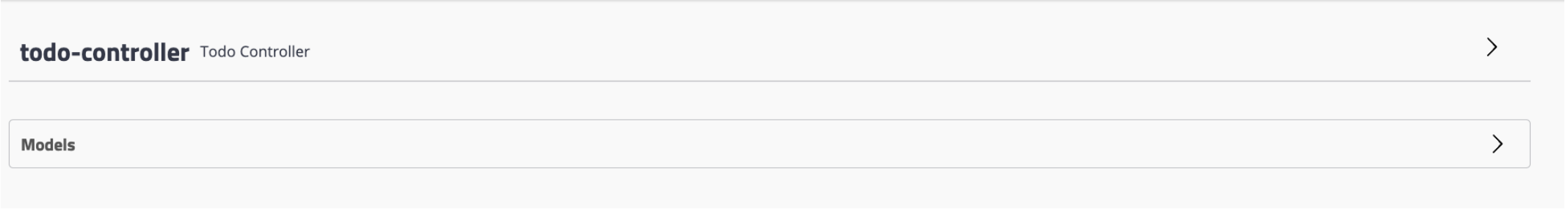
Swagger 핵심 Annontation

- RestController에 Annontation을 설정하여 Swagger-UI에서 원하는 값들이 출력하도록 할 수 있다.

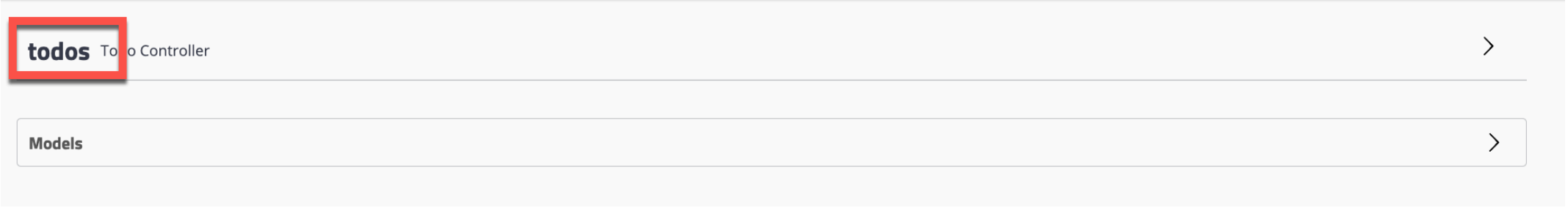
@Api = Api태그설정

```
@Api(tags = "todos")
@RestController
@RequestMapping(path = "/api/todos")
public class TodoController {
```

- @Api 안붙였을 때



- @Api 붙였을 때

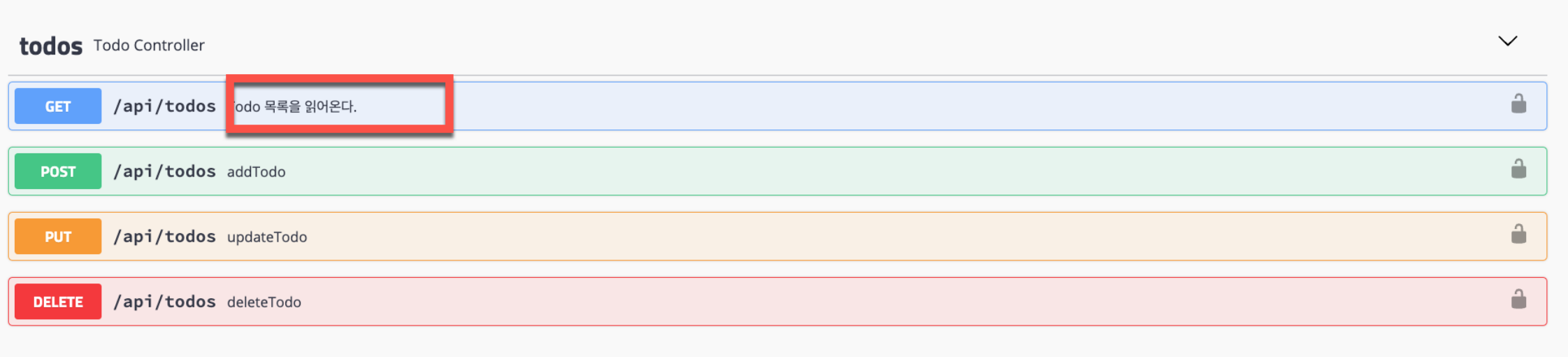


@ApiOperation = Method 설명

- @ApiOperation으로 해당 Controller 안의 method의 설명을 추가할 수 있다.

```
@ApiOperation(
        value = "Todo 목록을 읽어온다."
        , notes = "모든 Todo 목록을 읽어온다.")
@GetMapping
public List<Todo> getTodos(){
    return todoService.getTotos();
}
```

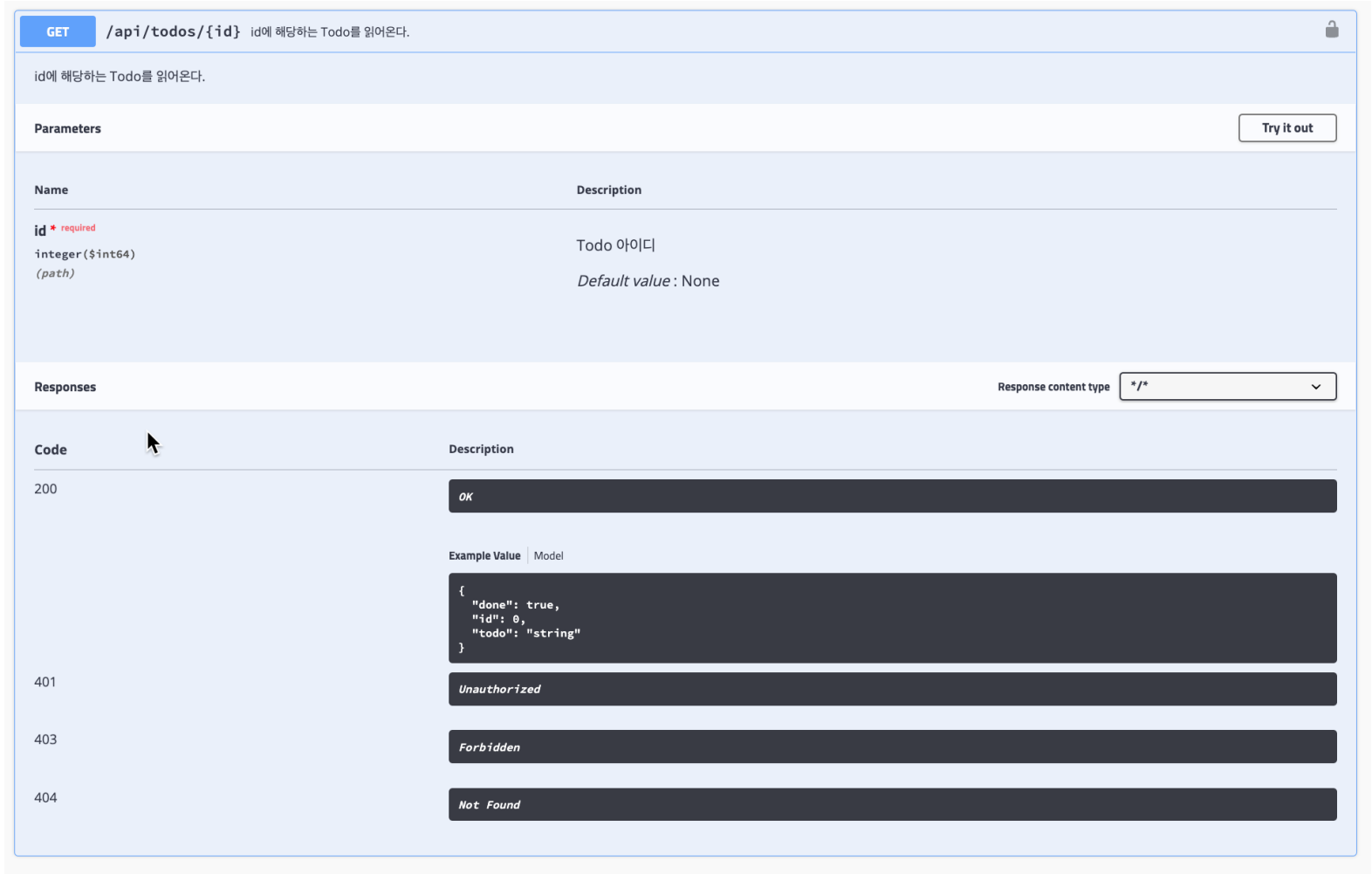
- API에 설명이 표시된다.



@ApiImplicitParam = Request Parameter 설명

- @ApiImplicitParam Annotation으로 해당 API Method 호출에 필요한 Parameter들의 설명을 추가할 수 있다.
- dataType, paramType, required의 경우 해당 name의 정보가 자동으로 채워지므로 생략 할 수 있다. paramType의 경우 @RequestParam은 query를, @PathVariable은 path를 명시해주면 된다. 만약 해당 Method의 Parameter가 복수일 경우, @ApiImplictParams로 @ApiImplictParam을 복수개 사용할 수 있다.

```
    @ApiOperation(
            value = "id에 해당하는 Todo를 읽어온다."
            , notes = "id에 해당하는 Todo를 읽어온다.")
    @ApiImplicitParam(
            name = "id"
            , value = "Todo 아이디"
            , required = true
            , dataType = "Long"
            , paramType = "path"
            , defaultValue = "None")
    @GetMapping("/{id}")
    public Todo getTodo(@PathVariable(name = "id")Long id){
        return todoService.getToto(id);
    }
```


GET
/api/todos/{id}
id에 해당하는 Todo를 읽어온다.
id에 해당하는 Todo를 읽어온다.
Parameters
Try it out
Name
Description
id * required
integer($int64)
(path)
Todo 아이디
Default value : None
Responses
Response content type
/
Code
Description
200
OK
Example Value
Model
{
"done": true,
"id": 0,
"todo": "string"
}
401
Unauthorized
403
Forbidden
404
Not Found

@ApiResponse = Reponse 설명

- @ApiResponse Annotation으로 해당 method의 Response에 대한 설명을 작성할 수 있다.
- 복수개의 Response에 대한 설명을 추가 하고 싶다면, @ApiResponses를 사용하면 된다.

```
    @ApiOperation(
            value = "Todo 목록을 읽어온다."
            , notes = "모든 Todo 목록을 읽어온다.")
    @ApiResponse(
            code = 200
            , message = "성공입니다."
        )
    @GetMapping
    public List<Todo> getTodos(){
        return todoService.getTotos();
    }
```

- 복수개를 설정하고 싶을 경우

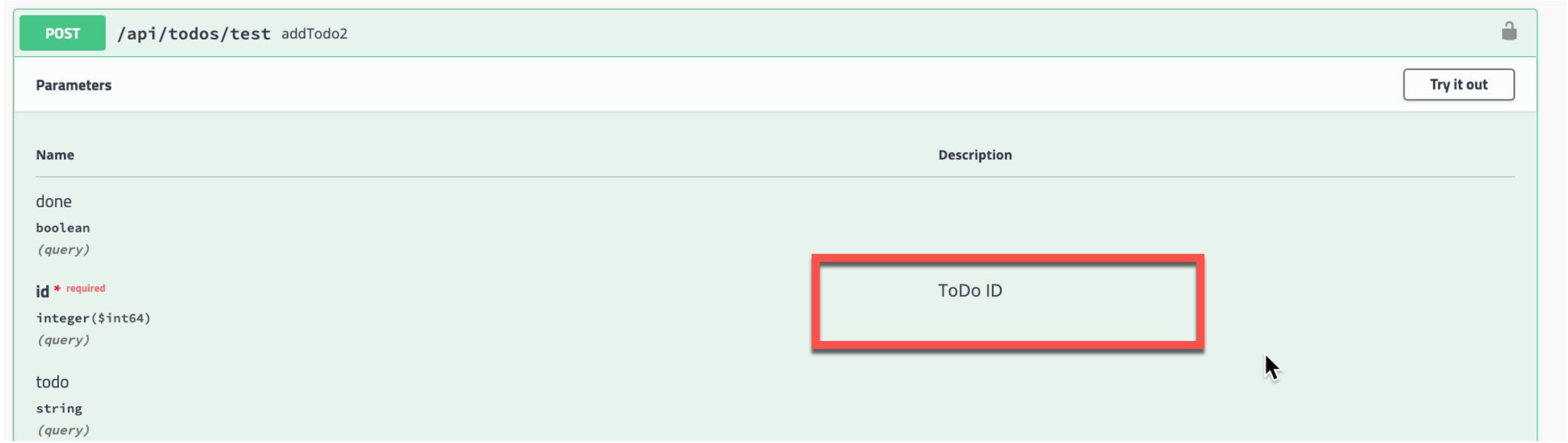
```
@ApiResponses({
      @ApiResponse(code = 200, message = "성공입니다.")
      , @ApiResponse(code = 400, message = "접근이 올바르지 않습니다.")
  })
```

- Default Response Message들을 삭제하고 싶다면, Swagger Config에 Docket에 useDefaultResponseMessages(false)를 설정해주면 된다. @ApiResponse로 설명하지 않은 401, 403, 404 응답들이 사라진다.

```
    public Docket getDocket(String groupName, Predicate<String> predicate) {
        return new Docket(DocumentationType.SWAGGER_2)
                        .useDefaultResponseMessages(false) // 지정한 응답 코드만 보여지도록 한다.
                .securityContexts(Arrays.asList(securityContext())) // bearer인증 설정
                .securitySchemes(Arrays.asList(apiKey())) // bearer 인증 설정
                        .groupName(groupName)
                .select()
                .apis(RequestHandlerSelectors.basePackage("examples.springmvc"))
                .paths(predicate)
                .apis(RequestHandlerSelectors.any())
                .build();
    }
```

@ApiParam = DTO field 설명

- @ApiParam Annotation으로 DTO의 field에 대한 설명을 추가할 수 있다.
- @ModelAttribute와 함께 사용될 때 설명에 표시가 된다.


POST
/api/todos/test
addTodo2
Parameters
Try it out
Name
Description
done
boolean
(query)
id * required
integer($int64)
(query)
ToDo ID
todo
string
(query)

@ApiModelProperty = DTO 예제 설명

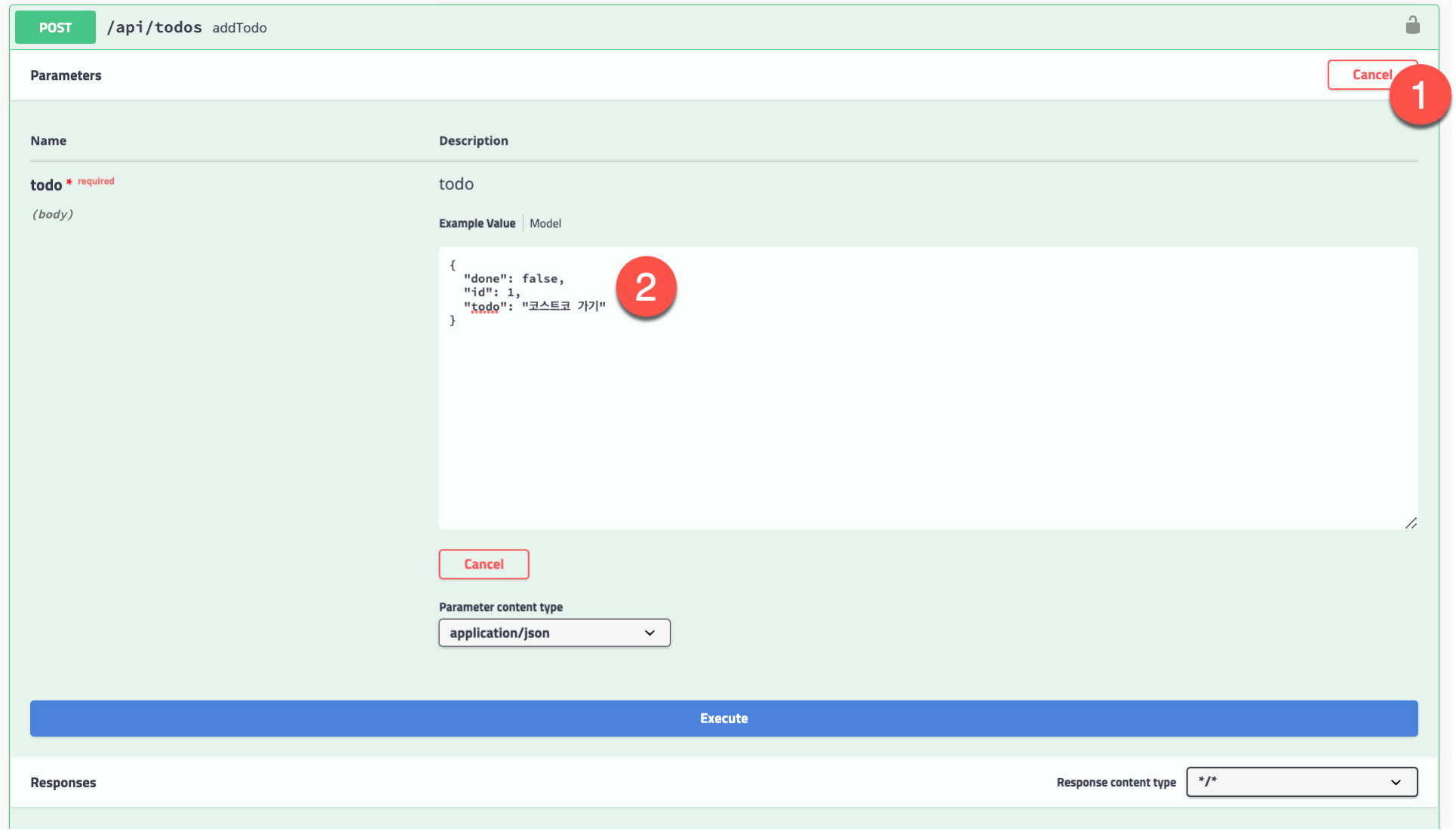
- @ApiModelProperty Annotation을 사용하는 DTO Class Field에 추가하면, 해당 DTO Field의 예제 Data를 추가할 수 있다.

```
package examples.springmvc.domain;

import java.util.Objects;

import io.swagger.annotations.ApiModelProperty;
import io.swagger.annotations.ApiParam;

public class Todo {
    private Long id;
    private String todo;
    private boolean done;

    public Todo(){
        this.done = false;
    }

    @ApiModelProperty(
            name = "id"
            , example = "1"
        )
    @ApiParam(value = "ToDo ID", required = true)
    public Long getId() {
        return id;
    }
```

```
    public void setId(Long id) {
        this.id = id;
    }

    public String getTodo() {
        return todo;
    }


    @ApiModelProperty(
            name = "todo"
            , example = "코스트코 가기"
        )
    public void setTodo(String todo) {
        this.todo = todo;
    }

    public boolean isDone() {
        return done;
    }
```

```
    @ApiModelProperty(
            name = "done"
            , example = "false"
        )
    public void setDone(boolean done) {
        this.done = done;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Todo todo = (Todo) o;
        return Objects.equals(id, todo.id);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }
}
```

- Todo를 @RequestBody로 입력받는데 어떤 경우는 example에 하나의 값만 보여지고 어떤 경우엔 3개보여지고 하는 방법은 없다. (있으면 알려주세요.)



@ApiIgnore = Swagger UI 상 무시

- 위 같은 코드에서 해당 method getNotice에서는 UserDTO의 id 만 필요한 상황인데, Parameter에 DTO를 삽입하여 모든 Parameter 정보가 노출되었다. @ApiIgnore을 추가함으로써, @ApiImplictParam으로 선언하지않은 parameter 정보들을 무시할 수 있다.
- 또한 method의 return type 앞에 명시해 해당 method를 아예 Swagger UI 에서 노출되지 않게 바꿀 수도 있다.

```
    @PostMapping("/test")
    public Todo addTodo2(@ModelAttribute Todo todo) {


        return todo;
    }
```

POST /api/todos/test addTodo2

Parameters Cancel

Name	Description
done boolean *(query)*	--
id * required integer($int64) *(query)*	ToDo ID id - ToDo ID
todo string *(query)*	todo

Execute

Responses Response content type */*

Code	Description

200	*OK*
	
	Example Value | Model
	{
	"done": false,
	"id": 1,
	"todo": "코스트코 가기"
	}

- @ApiIgnore 를 이용해 ModelAttribute로 입력받는 파라미터들을 무시하고, id만 입력받도록 설정한다.

```
    @ApiImplicitParams (
            @ApiImplicitParam(
                    name = "id"
                    , value = "사용자 아이디"
                    , required = true
                    , dataType = "string"
                    , paramType = "path"
                    , defaultValue = "None")
        )
    @PostMapping("/test")
    public Todo addTodo2(@ApiIgnore @ModelAttribute Todo todo) {


        return todo;
    }
```

POST /api/todos/test addTodo2

Parameters Cancel

Name	Description
id * required string *(path)*	사용자 아이디 None

Execute

Responses Response content type */*

Code	Description
200	*OK*

Example Value | Model

```
{
  "done": false,
  "id": 1,
  "todo": "코스트코 가기"
}
```

Swagger UI API 화면 커스텀

- Swagger Config Class에서 .apiInfo(ApiInfo Type)을 작성하면, Swagger UI 기본 화면의 API 설명 문구등을 커스텀할 수 있다.

```
    private static final String API_NAME = "ToDo API";
    private static final String API_VERSION = "0.0.1";
    private static final String API_DESCRIPTION = "ToDo API 명세서";

    ......

    //swagger 설정.
    public Docket getDocket(String groupName, Predicate<String> predicate) {
        return new Docket(DocumentationType.SWAGGER_2)
                        .useDefaultResponseMessages(false) // 지정한 응답 코드만 보여지도록 한다.
                .securityContexts(Arrays.asList(securityContext())) // bearer인증 설정
                .securitySchemes(Arrays.asList(apiKey())) // bearer 인증 설정
                        .groupName(groupName)
                        .apiInfo(apiInfo()) // <--- apiInfo를 추가한다.
......
    }


    public ApiInfo apiInfo() {
        return new ApiInfoBuilder()
                .title(API_NAME)
                .version(API_VERSION)
                .description(API_DESCRIPTION)
                .build();
    }
```

ToDo API 0.0.1

[Base URL: localhost:8080/webtodoserver]

http://localhost:8080/webtodoserver/v2/api-docs?group=todos

ToDo API 명세서

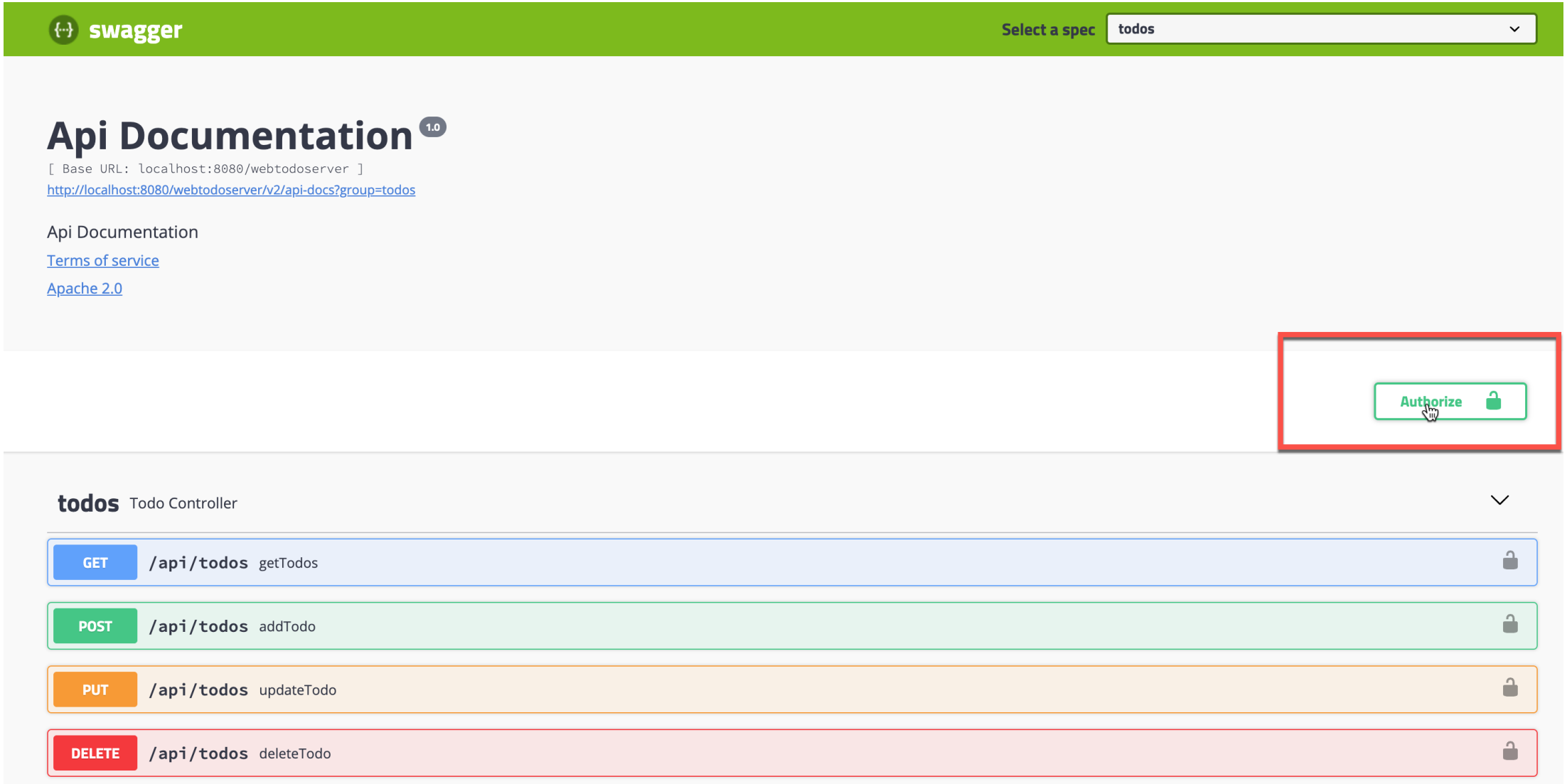
Bearer 인증 관련 설정

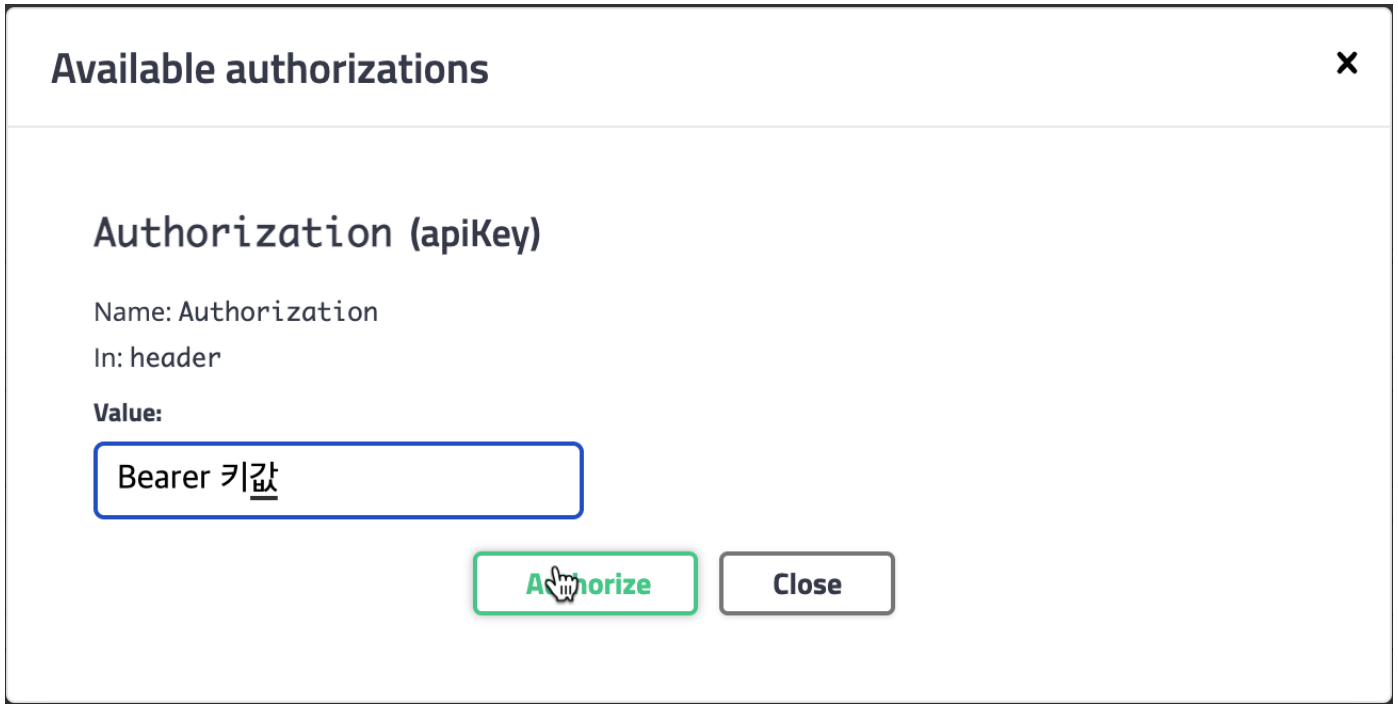
- Bearer 인증이란?
    - Bearer 토큰은 토큰을 소유한 사람에게 액세스 권한을 부여하는 일반적인 토큰 클래스입니다. 액세스 토큰, ID 토큰, 자체 서명 JWT는 모두 Bearer 토큰입니다. 인증에 Bearer 토큰을 사용하려면 HTTPS 와 같은 암호화된 프로토콜로 제공되는 보안이 필요합니다.

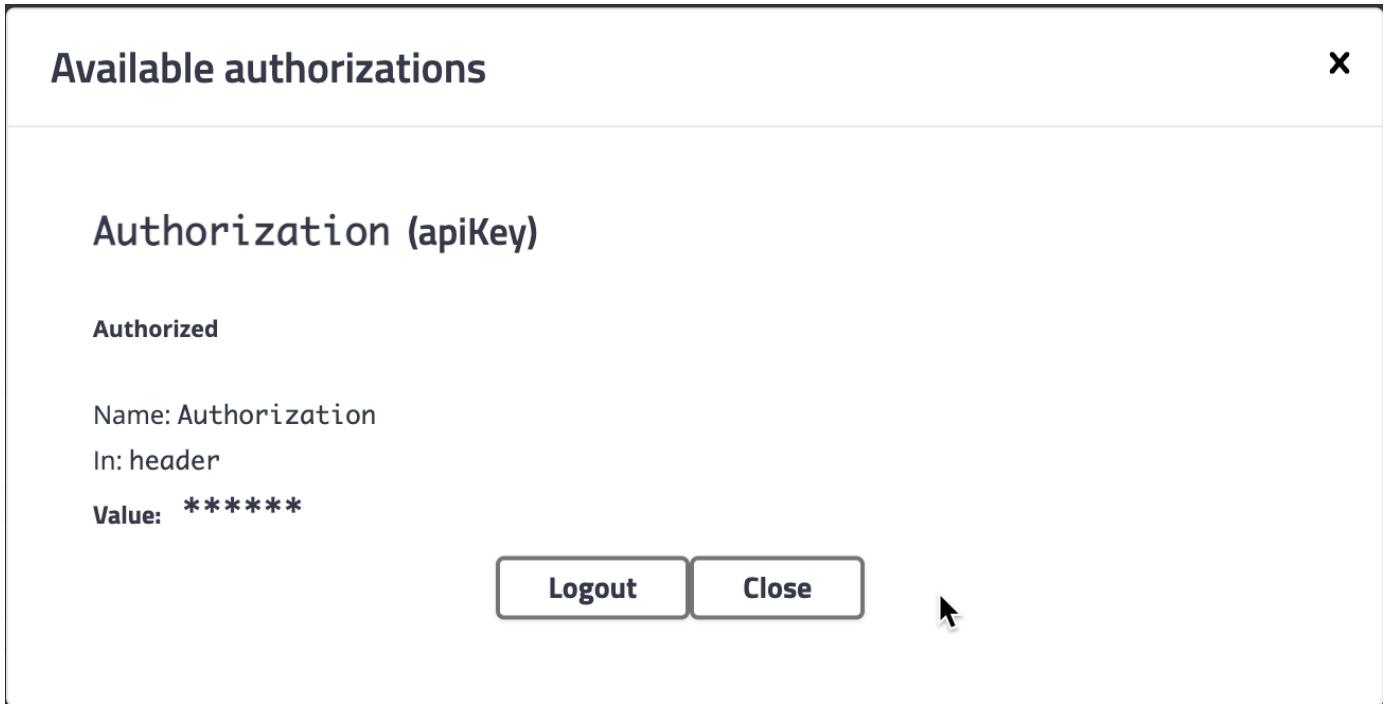
Bearer 인증 관련 설정

```
    //swagger 설정.
    public Docket getDocket(String groupName, Predicate<String> predicate) {
        return new Docket(DocumentationType.SWAGGER_2)
                .securityContexts(Arrays.asList(securityContext())) // bearer인증 설정
                .securitySchemes(Arrays.asList(apiKey())) // bearer 인증 설정
                        .groupName(groupName)
                .select()
                .apis(RequestHandlerSelectors.basePackage("examples.springmvc"))
                .paths(predicate)
                .apis(RequestHandlerSelectors.any())
                .build();
    }
```

```
    private SecurityContext securityContext() {
        return SecurityContext.builder()
            .securityReferences(defaultAuth())
            .build();
    }

    private List<SecurityReference> defaultAuth() {
        AuthorizationScope authorizationScope = new AuthorizationScope("global", "accessEverything");
        AuthorizationScope[] authorizationScopes = new AuthorizationScope[1];
        authorizationScopes[0] = authorizationScope;
        return Arrays.asList(new SecurityReference("Authorization", authorizationScopes));
    }

    private ApiKey apiKey() {
        return new ApiKey("Authorization", "Authorization", "header");
    }
```


swagger
Select a spec
todos
Api Documentation
1.0
[Base URL: localhost:8080/webtodoserver]
http://localhost:8080/webtodoserver/v2/api-docs?group=todos
Api Documentation
Terms of service
Apache 2.0
Authorize
todos Todo Controller
GET /api/todos getTodos
POST /api/todos addTodo
PUT /api/todos updateTodo
DELETE /api/todos deleteTodo


Available authorizations
Authorization (apiKey)
Name: Authorization
In: header
Value:
Bearer 키값
Authorize
Close


Available authorizations
Authorization (apiKey)
Authorized
Name: Authorization
In: header
Value: ******
Logout
Close

- Berarer 인증이란 Http 헤더이름은 Authorization, 헤더 값은 "Bearer 키값"으로 오는 형태이다. 이 값을 필터 등으로 공통으로 처리하도록 보통 프로그래밍 한다.

```
    @GetMapping("/bearerTest")
    public String bearerTest(@ApiIgnore @RequestHeader("Authorization") String authorization) {
        return authorization;
    }
```

GET /api/todos/bearerTest bearerTest

Parameters Cancel

No parameters

Execute | Clear

Responses Response content type */*

Curl

```
curl -X GET "http://localhost:8080/webtodoserver/api/todos/bearerTest" -H "accept: */*" -H "Authorization: Bearer asdfasdf"
```

Request URL

```
http://localhost:8080/webtodoserver/api/todos/bearerTest
```

Server response

Code	Details
200	**Response body** `"Bearer asdfasdf"` Download

감사합니다.