



나는 왜 코프링 컨트롤러를 더 이상 만들지 않게 되었나?

백엔드 개발자(대학생) 김경환



목차

01 발표자 소개

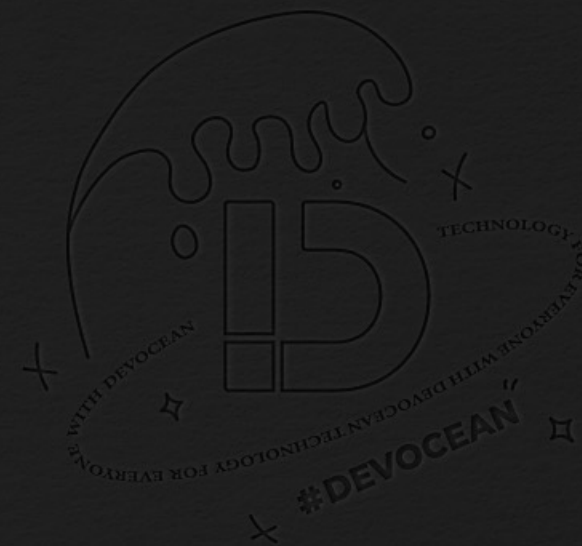
02 코틀린 스터디 소개

03 OpenAPI란?

04 OpenAPI Generator?

05 실제 프로젝트 적용 후기

06 Q&A



01 발표자 소개



김 경 환

게으르고 싶어서 부지런히 개발하는 개발자 입니다

- 세종대학교 전자정보통신공학과 3학년 재학 중
- DEVOCEAN YOUNG 3기 활동 중! ☺
- 과학기술정보통신부 청년 정책 자문단원
- (전) 데이터 플랫폼 스타트업 플랫폼 엔지니어



02 코틀린 스터디 소개



Kotlin 웹 어플리케이션 개발

스터디 목표

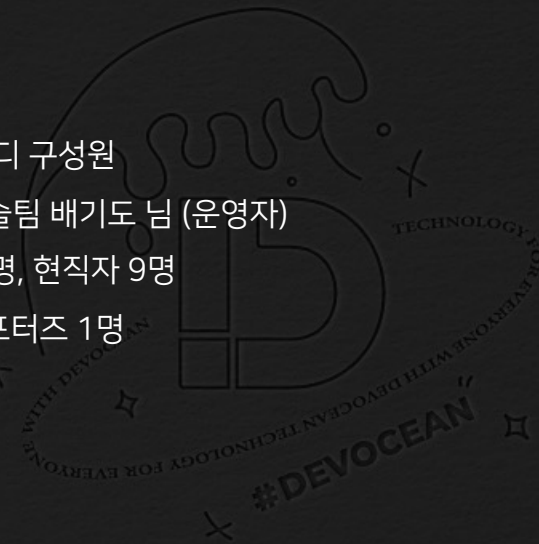
Kotlin 프로그래밍 언어와 프레임워크를 이용하여 간단한
MVP REST API 개발하기

스터디 구성원

SK텔레콤 구독기술팀 배기도 님 (운영자)

대학생 2명, 현직자 9명

+ 서포터즈 1명



02 코틀린 스터디 소개



이용한 프로젝트 진행

Kotlin 웹 어플리케이션 개발
+ OpenAPI

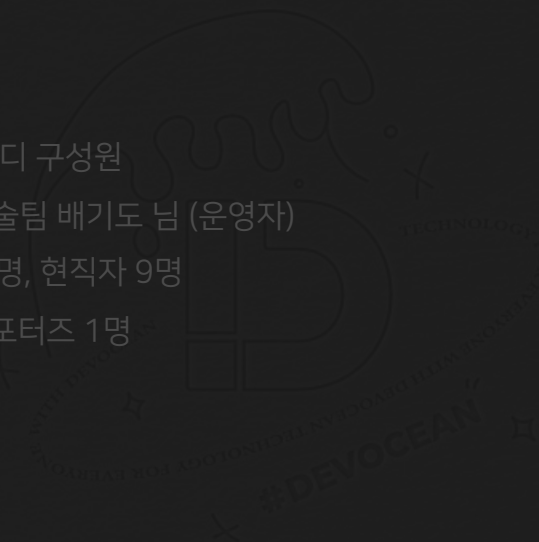
Kotlin 프로그래밍 언어와 프레임워크를 이용하여 간단한 MVP REST API 개발하기

스터디 구성원

SK텔레콤 구독기술팀 배기도 님 (운영자)

대학생 2명, 현직자 9명

+ 서포터즈 1명



03 OpenAPI란? - 기존의 프로젝트 커뮤니케이션

저희 이번에 API 새로 개발한 거랑 Parameter 일부 수정해서 배포 했습니다!

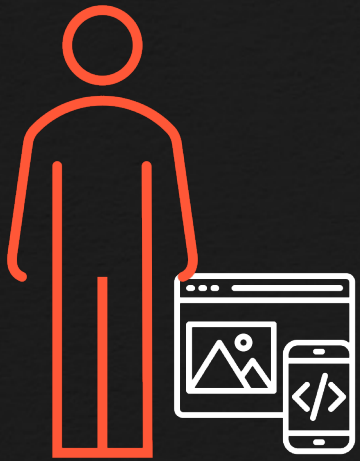
네 알겠습니다~! 개발 서버에 붙여서 개발 진행해보겠습니다!

어 ... 이거 제대로 요청한 거 같은데 **에러** 발생하는데요 ..?

네 ..? 어 TC 돌렸을 때는 잘됐는데.. 잠시만요 ..

아 .. Wiki에 **API 명세서 최신화**가 안되었네요 ..

아하 .. 그럼 최신화 하고 알려주세요 ..



프론트엔드 개발자



백엔드 개발자

03 OpenAPI란? - 기존의 프로젝트 커뮤니케이션

저희 이번에 API 새로 개발한 거랑 Parameter 일부 수정해서 배포 했습니다!

네 알겠습니다~! 개발 서버에 붙여서 개발 진행해보겠습니다!

API 명세서 언제 수정하지..?

네 ..? 어 TC 돌렸을 때는 잘됐는데.. 잠시만요 ..
쉽게 수정하는 방법 없나?

아 .. Wiki에 API 명세서 최신화가 안되었네요 ..

아하 .. 그럼 최신화 하고 알려주세요 ..



프론트엔드 개발자



백엔드 개발자

03 OpenAPI란? – Swagger

Swagger

- API 개발자들이 API를 설계, 빌드, 문서화, 개발할 수 있도록 도와주는 오픈 소스 소프트웨어 프레임워크
- 코드에 Swagger Annotation을 삽입하여 코드와 함께 API 문서를 관리, 생성할 수 있음.
- Swagger UI를 통해 API 문서를 확인하고, API를 테스트 할 수 있음.

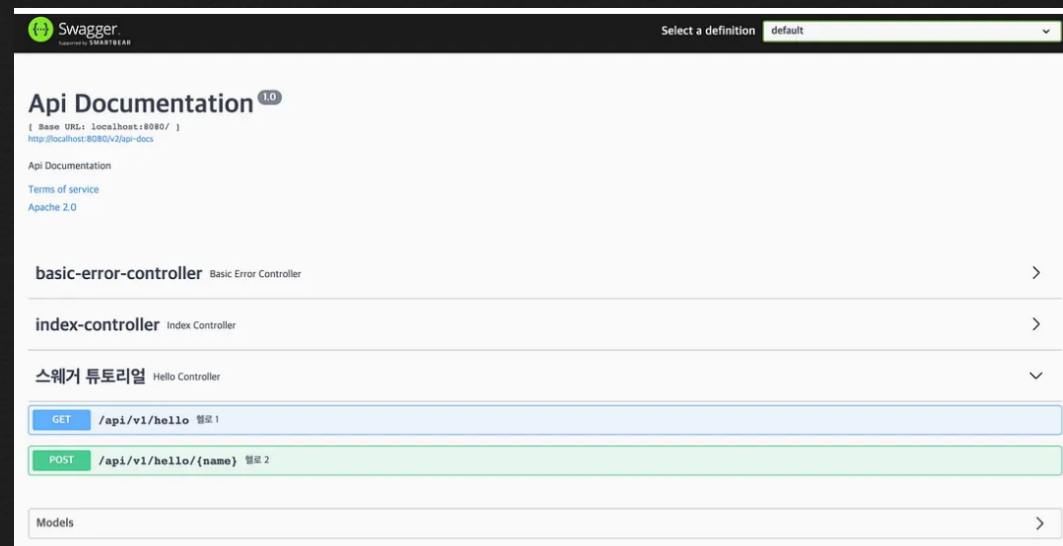
```
import io.swagger.annotations.ApiParam;

@Api(tags = "스웨거 테스트용 API")
@RestController
@RequestMapping("/api/v1")
public class HelloController {

    @ApiOperation(value = "헬로 1")
    @GetMapping("/hello")
    public ResponseEntity<String> hello1() {
        return ResponseEntity.ok("Hello World!");
    }

    @ApiOperation(value = "헬로 2")
    @ApiResponses(value = {
        @ApiResponse(code = 200, message = "정상 성공", response = String.class),
        @ApiResponse(code = 500, message = "서버 에러")
    })
    @PostMapping("/hello/{name}")
    public ResponseEntity<String> hello2(
        @ApiParam(value = "이름") @PathVariable String name) {
        if (name.equals("aa")) {
            throw new IllegalArgumentException("유효하지 않은 이름입니다.");
        }
        String body = "Hello, " + name;
        return ResponseEntity.ok(body);
    }
}
```

코드에 Swagger 관련 Annotation 삽입



Swagger UI 화면

03 OpenAPI란? – OpenAPI Specification (OAS)



- 프로그래밍 언어에 종속되지 않은 HTTP API를 위한 표준 인터페이스 정의
- 기존의 Swagger Specification으로 부터 시작.

표준화된 포맷

OpenAPI Specification은
JSON 또는 YAML 형식을 사용하여
API를 정의함.
API의 구조와 동작을
명확하고 일관되게 기술.

유연성

특정 언어나 프레임워크에 종속되지 않으며,
다양한 언어와 환경에서 사용할 수 있음. API의
인터페이스를 정의하는 데 있어 유연성을 제공

코드 생성



03 OpenAPI란? – OpenAPI Specification (OAS)



- 프로그래밍 언어에 종속되지 않은 HTTP API를 위한 표준 인터페이스 정의
- 기존의 Swagger Specification으로 부터 시작.

표준화된 포맷

OpenAPI Specification은
JSON 또는 YAML 형식을 사용하여
API를 정의함.
API의 구조와 동작을
명확하고 일관되게 기술.

유연성

특정 언어나 프레임워크에 종속되지 않으며,
다양한 언어와 환경에서 사용할 수 있음. API의
인터페이스를 정의하는 데 있어 유연성을 제공

코드 생성



03 OpenAPI란? – OpenAPI Specification (OAS)



- 프로그래밍 언어에 종속되지 않은 HTTP API
- 기존의 Swagger Specification으로 부터

표준화된 포맷

OpenAPI Specification은
JSON 또는 YAML 형식을 사용하여
API를 정의함.
API의 구조와 동작을
명확하고 일관되게 기술.

코드 생성

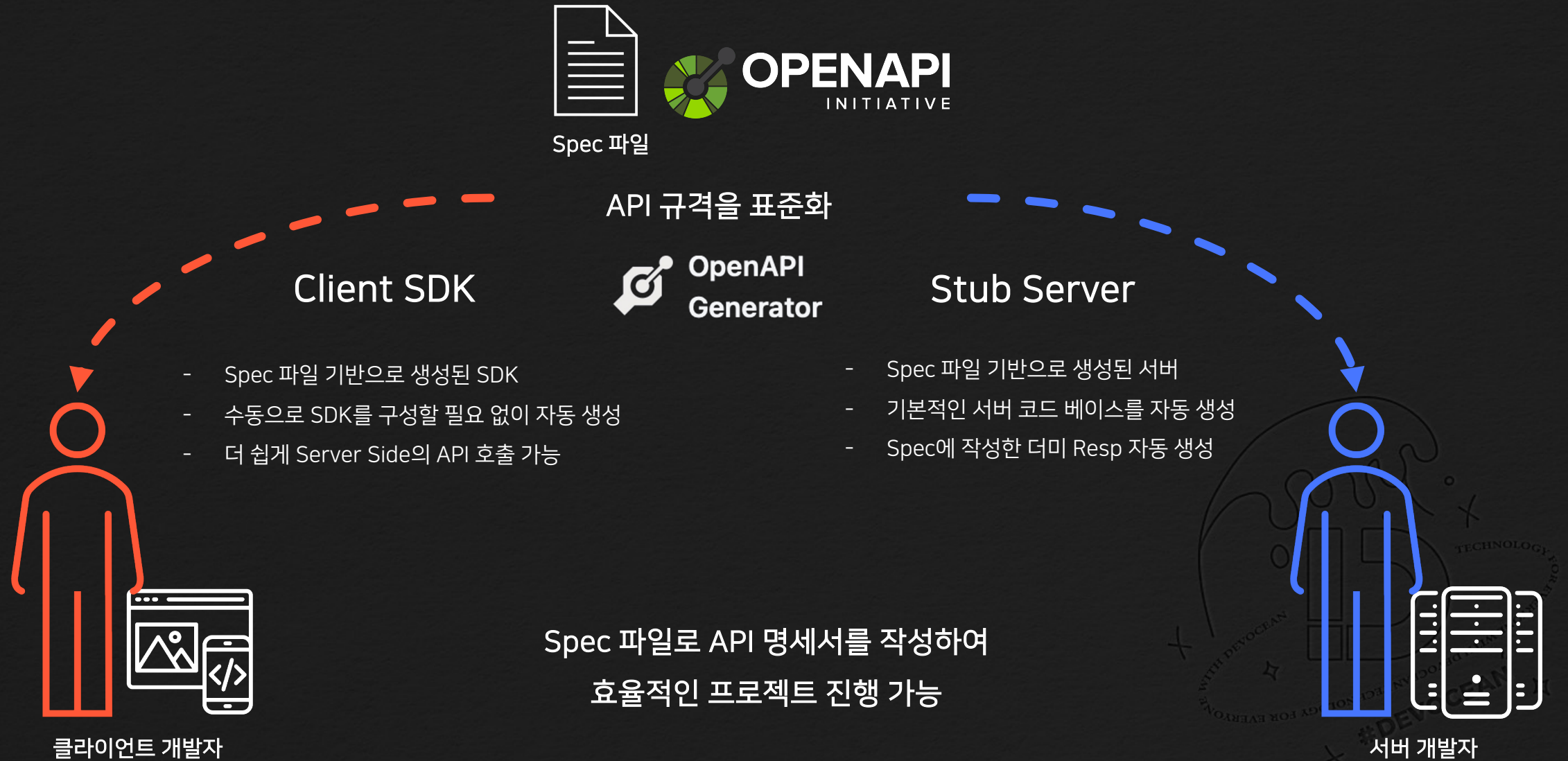


OpenAPI
Generator

특정 언어나 프레임워크에 종속되지 않으며,
다양한 언어와 환경에서 사용할 수 있음. API의
인터페이스를 정의하는 데 있어 유연성을 제공



04 OpenAPI Generator?



04 OpenAPI Generator? – Spec 파일 (w. Swagger Editor)

```
Swagger Editor
1 openapi: 3.0.3
2 info:
3   title: Swagger Petstore - OpenAPI 3.0
4   termsOfService: http://swagger.io/terms/
5   contact:
6     email: apiteam@swagger.io
7   license:
8     name: Apache 2.0
9     url: http://www.apache.org/licenses/LICENSE-2.0.html
10  version: 1.0.11
11  externalDocs:
12    description: Find out more about Swagger
13    url: http://swagger.io
14  servers:
15    - url: https://petstore3.swagger.io/api/v3
16  tags:
17    - name: pet
18      description: Everything about your Pets
19      externalDocs:
20        description: Find out more
21        url: http://swagger.io
22    - name: store
23      description: Access to Petstore orders
24      externalDocs:
25        description: Find out more about our store
26        url: http://swagger.io
27    - name: user
28      description: Operations about user
29  paths:
30    /pet:
31      put:
32        tags:
33          - pet
34        summary: Update an existing pet
35        description: Update an existing pet by Id
36        operationId: updatePet
37        requestBody:
38          description: Update an existent pet in the store
39          content:
40            application/json:
41              schema:
42                $ref: '#/components/schemas/Pet'
43            application/xml:
44              schema:
45                $ref: '#/components/schemas/Pet'
46            application/x-www-form-urlencoded:
47              schema:
48                $ref: '#/components/schemas/Pet'
49          required: true
50        responses:
51          '200':
52            description: Successful operation
53            content:
54              application/json:
55                schema:
56                  $ref: '#/components/schemas/Pet'
57              application/xml:
58                schema:
59                  $ref: '#/components/schemas/Pet'
60          '400':
61            description: Invalid ID supplied
```

- editor.swagger.io 에서 petstore Spec 파일 예시 제공

Spec 파일 YAML 기본 구조

Key	설명
openapi	OpenAPI 사양의 버전을 명시
info	API 문서의 기본 정보를 기술 API의 제목, 버전, 설명, 저자 및 라이선스 정보를 포함.
Server	API 서버들의 URL과 설명을 나열
tags	API의 엔드포인트들을 카테고리화하여 관련된 작업들을 그룹화하는 데 사용
paths	경로별로 묶음으로 API 엔드포인트 관련 정보를 기술. 각 경로에서의 HTTP 메서드, 요청, 응답 매개변수를 정의.
components	스키마, 응답, 매개변수, 예제, 요청 본문, 헤더, 보안 스키마 및 링크와 같은 재사용 가능한 API 구성 요소 를 정의.

04 OpenAPI Generator? – Spec 파일 (w. Swagger Editor)

```
Swagger Editor
1 openapi: 3.0.3
2 info:
3   title: Swagger Petstore - OpenAPI 3.0
4   termsOfService: http://swagger.io/terms/
5   contact:
6     email: apiteam@swagger.io
7   license:
8     name: Apache 2.0
9     url: http://www.apache.org/licenses/LICENSE-2.0.html
10  version: 1.0.11
11  externalDocs:
12    description: Find out more about Swagger
13    url: http://swagger.io
14  servers:
15    - url: https://petstore3.swagger.io/api/v3
16  tags:
17    - name: pet
18      description: Everything about your Pets
19    - name: store
20      description: Access to Petstore orders
21    - name: user
22      description: Operations about user
23  paths:
24    /pet:
25      put:
26        tags:
27          - pet
28        summary: Update an existing pet
29        description: Update an existing pet by Id
30        operationId: updatePet
31        requestBody:
32          description: Update an existent pet in the store
33          content:
34            application/json:
35              schema:
36                $ref: '#/components/schemas/Pet'
37            application/xml:
38              schema:
39                $ref: '#/components/schemas/Pet'
40            application/x-www-form-urlencoded:
41              schema:
42                $ref: '#/components/schemas/Pet'
43          required: true
44        responses:
45          '200':
46            description: Successful operation
47            content:
48              application/json:
49                schema:
50                  $ref: '#/components/schemas/Pet'
51              application/xml:
52                schema:
53                  $ref: '#/components/schemas/Pet'
54          '400':
55            description: Invalid ID supplied
```



생성된 Swagger UI

Swagger Petstore - OpenAPI 3.0

1.0.11 OAS 3.0

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

Servers
https://petstore3.swagger.io/api/v3 [Authorize](#)

pet

Everything about your Pets [Find out more](#)

PUT /pet Update an existing pet

Update an existing pet by Id

Parameters
No parameters [Try it out](#)

Request body **required** application/json

Update an existent pet in the store

Example Value | Schema

```
{
  "id": 10,
  "name": "doggie",
  "category": {
    "id": 1,
    "name": "Dogs"
  },
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ]
}
```

04 OpenAPI Generator? – Client Side Generate

Generate 방법 (OpenAPI Generator Install)

- npm, Homebrew, Scoop, Docker, JAR 등 지원
- <https://openapi-generator.tech/docs/installation>
- editor.swagger.io 에서 Spec 파일 Import 후 Generate Client 탭을 통해 생성

Swagger Editor
Sponsored by SMARTBEAR

File Edit Insert Generate Server Generate Client About

```
1 openapi: 3.0.3
2 info:
3   title: Swagger Petstore - OpenAPI 3.0
4   termsOfService: http://swagger.io/terms/
5   contact:
6     email: apiteam@swagger.io
7   license:
8     name: Apache 2.0
9     url: http://www.apache.org/licenses/LICENSE-2.0.html
10  version: 1.0.11
11 externalDocs:
12   description: Find out more about Swagger
13   url: http://swagger.io
14 servers:
15   - url: https://petstore3.swagger.io/api/v3
16 tags:
17   - name: pet
18     description: Everything about your Pets
19     externalDocs:
20       description: Find out more
21       url: http://swagger.io
22   - name: store
23     description: Access to Petstore orders
24     externalDocs:
25       description: Find out more about our store
26       url: http://swagger.io
27   - name: user
28     description: Operations about user
29 paths:
```

csharp	kotlin-client	swift5
csharp-dotnet2	openapi	typescript-angular
dart	openapi-yaml	typescript-axios
dynamic-html	php	typescript-fetch
go	python	
html	r	
html2	ruby	
java	scala	
javascript	swift3	
jaxrs-cxf-client	swift4	

- 다양한 언어, 프레임워크 지원



04 OpenAPI Generator? – Client Side Generate (typescript)

▼ TYPESCRIPT-AXIOS-CLIENT-GENERATED

> .swagger-codegen

▼ apis

TS pet-api.ts

TS store-api.ts

TS user-api.ts

▼ models

TS address.ts

TS api-response.ts

TS category.ts

TS customer.ts

TS index.ts

TS order.ts

TS pet.ts

TS tag.ts

TS user.ts

◆ .gitignore

npm .npmignore

≡ .swagger-codegen-ignore

TS api.ts

TS base.ts

TS configuration.ts

\$ git_push.sh

TS index.ts

{ } package-lock.json

{ } package.json

① README.md

tsconfig.json

/apis

- API 엔드포인트 별로 생성된 파일들이 포함된 폴더.
- 각 파일은 특정 API 엔드포인트에 대한 클라이언트 코드와 요청 메서드를 포함.

/models

- API 요청 및 응답에 사용되는 데이터 모델 클래스들이 포함된 폴더.
- 각 파일은 API에서 사용하는 데이터 구조를 정의.



04 OpenAPI Generator? – Client Side Generate (typescript)

▼ TYPESCRIPT-AXIOS-CLIENT-GENERATED

> .swagger-codegen

▼ apis

TS pet-api.ts

TS store-api.ts

TS user-api.ts

▼ models

TS address.ts

TS api-response.ts

TS category.ts

TS customer.ts

TS index.ts

TS order.ts

TS pet.ts

TS tag.ts

TS user.ts

◆ .gitignore

npm .npmignore

≡ .swagger-codegen-ignore

TS api.ts

TS base.ts

TS configuration.ts

\$ git_push.sh

TS index.ts

{ } package-lock.json

{ } package.json

① README.md

ts tsconfig.json

```
/**
 * StoreApi - object-oriented interface
 * @export
 * @class StoreApi
 * @extends {BaseAPI}
 */
export class StoreApi extends BaseAPI {
  /**
   * For valid response try integer IDs with value < 1000. Anything above 1000 or nonintegers will generate API errors
   * @summary Delete purchase order by ID
   * @param {number} orderId ID of the order that needs to be deleted
   * @param {*} [options] Override http request option.
   * @throws {RequiredError}
   * @memberof StoreApi
   */
  public async deleteOrder(orderId: number, options?: AxiosRequestConfig) : Promise<AxiosResponse<void>> {
    return StoreApiFp(this.configuration).deleteOrder(orderId, options).then((request) => request(this.axios, this.basePath));
  }

  /**
   * Returns a map of status codes to quantities
   * @summary Returns pet inventories by status
   * @param {*} [options] Override http request option.
   * @throws {RequiredError}
   * @memberof StoreApi
   */
  public async getInventory(options?: AxiosRequestConfig) : Promise<AxiosResponse<{ [key: string]: number; }>> {
    return StoreApiFp(this.configuration).getInventory(options).then((request) => request(this.axios, this.basePath));
  }
}
```

- Factory 패턴을 기반으로 SDK 생성.

클라이언트 개발자는 SDK를 이용해 Method 호출로
간편한 API 연동 가능



04 OpenAPI Generator? – Server Side Generate (Kotlin Spring)

Generate 방법 (의존성 추가)

- openapi-generator: openapi를 이용하여 기존 엔드포인트를 생성할 수 있는 라이브러리.
- <https://devocean.sk.com/blog/techBoardDetail.do?ID=164961&boardType=techBlog>
- Spec 파일을 기반으로 프로젝트 빌드 시, stub server 코드가 생성됨.

```
import javax.annotation.Generated;

@Generated(value = "org.openapitools.codegen.languages.SpringCodegen", date = "2023-05-15T16:32:18.367315+09:00[Asia/Seoul]")
@Validated
@Tag(name = "pet", description = "Everything about your Pets")
public interface PetApi {

    default Optional<NativeWebRequest> getRequest() {
        return Optional.empty();
    }

    /**
     * POST /pet : Add a new pet to the store
     *
     * @param pet Pet object that needs to be added to the store (required)
     * @return successful operation (status code 200)
     *         or Invalid input (status code 405)
     */
    @Operation(
        operationId = "addPet",
        summary = "Add a new pet to the store",
        description = "",
        tags = { "pet" },
        responses = {
            @ApiResponse(responseCode = "200", description = "successful operation", content = {
                @Content(mediaType = "application/xml", schema = @Schema(implementation = Pet.class)),
                @Content(mediaType = "application/json", schema = @Schema(implementation = Pet.class))
            }),
            @ApiResponse(responseCode = "405", description = "Invalid input")
        },
        security = {
            @SecurityRequirement(name = "petstore_auth", scopes = { "write:pets", "read:pets" })
        }
    )
```

```
@RequestMapping(
    method = RequestMethod.POST,
    value = "/pet",
    produces = { "application/xml", "application/json" },
    consumes = { "application/json", "application/xml" }
)
default ResponseEntity<Pet> addPet(
    @Parameter(name = "Pet", description = "Pet object that needs to be added to the store", required = true) )
) {
    getRequest().ifPresent(request -> {
        for (MediaType mediaType: MediaType.parseMediaTypes(request.getHeader("Accept"))) {
            if (mediaType.isCompatibleWith(MediaType.valueOf("application/json"))) {
                String exampleString = "{ \"photoUrls\" : [ \"photoUrls\", \"photoUrls\" ], \"name\" : \"doggie\", \"id\" }";
                ApiUtil.setExampleResponse(request, "application/json", exampleString);
                break;
            }
            if (mediaType.isCompatibleWith(MediaType.valueOf("application/xml"))) {
                String exampleString = "<Pet> <id>123456789</id> <Category> <id>123456789</id> <name>aeiou</name> </Category>";
                ApiUtil.setExampleResponse(request, "application/xml", exampleString);
                break;
            }
        }
    });
    return new ResponseEntity<>(HttpStatus.NOT_IMPLEMENTED);
}

/ ... 생략
```

생성된 서버 코드

04 OpenAPI Generator? – Server Side Generate (Kotlin Spring)

Generate 방법 (의존성 추가)

- openapi-generator: openapi를 이용하여 기존 엔드포인트를 생성할 수 있는 라이브러리.
- <https://devocean.sk.com/blog/techBoardDetail.do?ID=164961&boardType=techBlog>
- Spec 파일을 기반으로 프로젝트 빌드 시, stub server 코드가 생성됨.

```
import javax.annotation.Generated;

@Generated(value = "org.openapitools.codegen.languages.SpringCodegen", date = "2023-05-15T16:32:18.367315+09:00[Asia/Seoul]")
@Validated
@Tag(name = "pet", description = "Everything about your Pets")
public interface PetApi {
```

```
    default Optional<NativeWebRequest> getRequest() {
        return Optional.empty();
    }

    /**
     * POST /pet : Add a new pet to the store
     *
     * @param pet Pet object that needs to be added to the store (required)
     * @return successful operation (status code 200)
     *         or Invalid input (status code 405)
     */
```

Swagger 관련 Annotation

```
@Operation(
    operationId = "addPet",
    summary = "Add a new pet to the store",
    description = "",
    tags = { "pet" },
    responses = {
        @ApiResponse(responseCode = "200", description = "successful operation", content = {
            @Content(mediaType = "application/xml", schema = @Schema(implementation = Pet.class)),
            @Content(mediaType = "application/json", schema = @Schema(implementation = Pet.class))
        }),
        @ApiResponse(responseCode = "405", description = "Invalid input")
    },
    security = {
        @SecurityRequirement(name = "petstore_auth", scopes = { "write:pets", "read:pets" })
    }
)
```

```
@RequestMapping(
    method = RequestMethod.POST,
    value = "/pet",
    produces = { "application/xml", "application/json" },
    consumes = { "application/json", "application/xml" }
)
```

default 컨트롤러 interface 생성

```
default ResponseEntity<Pet> addPet(
    @Parameter(name = "Pet", description = "Pet object that needs to be added to the store", required = true) (
) {
```

```
    getRequest().ifPresent(request -> {
        for (MediaType mediaType: MediaType.parseMediaTypes(request.getHeader("Accept"))) {
            if (mediaType.isCompatibleWith(MediaType.valueOf("application/json"))) {
                String exampleString = "{ \"photoUrls\" : [ \"photoUrls\", \"photoUrls\" ], \"name\" : \"doggie\", \"id\" : 123456789 }";
                ApiUtil.setExampleResponse(request, "application/json", exampleString);
                break;
            }
        }
    });
```

예시 응답 Return

```
        if (mediaType.isCompatibleWith(MediaType.valueOf("application/xml"))) {
            String exampleString = "<Pet> <id>123456789</id> <Category> <id>123456789</id> <name>aeiou</name> </Category>";
            ApiUtil.setExampleResponse(request, "application/xml", exampleString);
            break;
        }
    });
    return new ResponseEntity<>(HttpStatus.NOT_IMPLEMENTED);
}
```

/ ... 생략

04 OpenAPI Generator? – Server Side Generate

생성된 코드

```
@RequestMapping(
    method = RequestMethod.POST,
    value = "/pet",
    produces = { "application/xml", "application/json" },
    consumes = { "application/json", "application/xml" }
)
default ResponseEntity<Pet> addPet(
    @Parameter(name = "Pet", description = "Pet object that needs to be added"
) {
    getRequest().ifPresent(request -> {
        for (MediaType mediaType: MediaType.parseMediaTypes(request.getHeader("Accept"
        if (mediaType.isCompatibleWith(MediaType.valueOf("application/json")) {
            String exampleString = "{ \"photoUrls\" : [ \"photoUrls\", \"photoUrls\"
            ApiUtil.setExampleResponse(request, "application/json", exampleString);
```

Override 후 비즈니스 로직 삽입

작성하는 코드

```
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class PetControllerImpl implements PetApi {

    @Override
    public ResponseEntity<Pet> addPet(Pet pet) {
        // TODO.. 생성된 pet Rest api 에 해당하는 비즈니스 로직을 작성하자.
        return PetApi.super.addPet(pet);
    }

    @Override
    public ResponseEntity<Void> deletePet(Long petId, String apiKey) {
        // TODO.. 삭제된 pet Rest api 에 해당하는 비즈니스 로직을 작성하자.
        return PetApi.super.deletePet(petId, apiKey);
    }

    @Override
    public ResponseEntity<Pet> findPetById(Long petId) {
        // TODO.. 조회된 pet Rest api 에 해당하는 비즈니스 로직을 작성하자.
        return PetApi.super.findPetById(petId);
    }
}
```



04 OpenAPI Generator? - 코프링 컨트롤러가 필요하지 않는 이유

①

Stub Server 코드 생성으로
직접적인 **컨트롤러 개발이 필요 없음.**

②

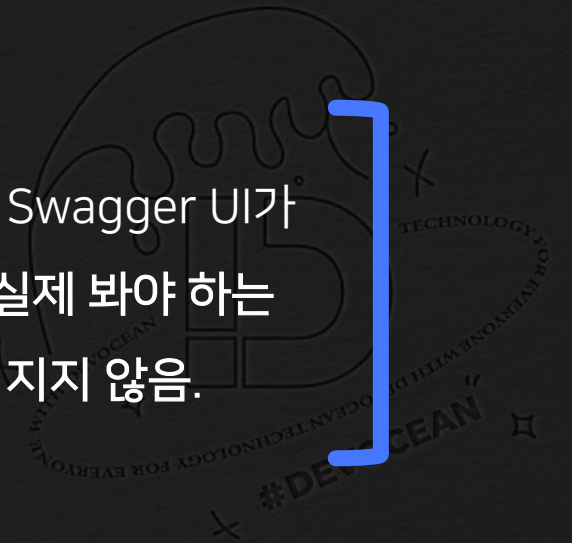
생성된 컨트롤러 부분의 코드를
Override 받아 작성

③

컨트롤러 부분을 Override 받아 구성하기
때문에 **비즈니스 로직에 집중**할 수 있음.

④

Stub Server 코드 내에 Swagger UI가
적용되어 있기 때문에 실제 봐야 하는
소스코드가 더러워 지지 않음.

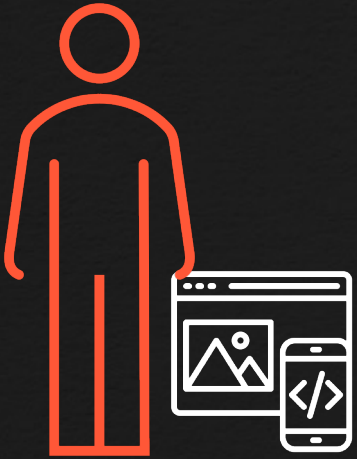


04 OpenAPI Generator - 변화한 프로젝트 커뮤니케이션

저희 이번에 API 수정된 부분 Spec 파일 업데이트 했습니다!

네 알겠습니다~! Spec 파일로 SDK 생성해서 작업할게요!

정상 동작 하네요 ! 저희 테스트 서버에 넘기시죠 ~ 고생하셨습니다!



프론트엔드 개발자



휴 .. 야근은 안 해도 되겠다..



백엔드 개발자

05 실제 프로젝트 적용 후기(1) - 리팩토링 적용



기존 개발된 플랫폼

레거시 코드 자바 버전 업그레이드
도메인 별 각기 다른 디자인 패턴 적용



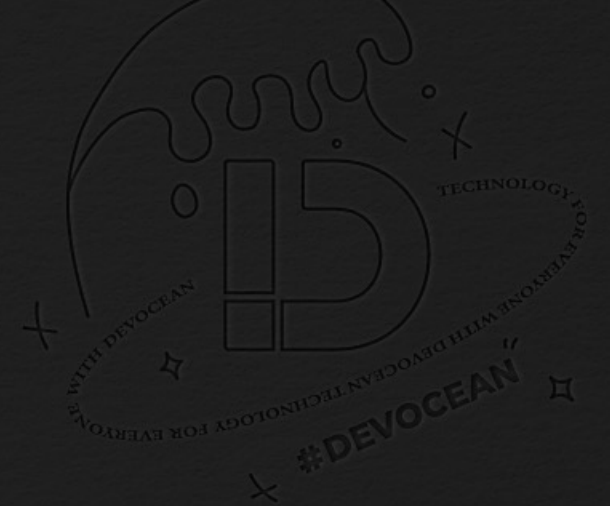
이미 작성된
API 명세서 존재



OpenAPI
Spec으로 수정



OpenAPI Generator를
이용하여 빠르게
리팩토링 수행 진행



05 실제 프로젝트 적용 후기(2) - 오픈소스 연동

서비스 개발 중, 특정 오픈 소스와 연동하고자 하는 의견.

하지만, Official SDK는 아무리 찾아도 없고.. 단지 REST API를 지원함을 확인

오픈 소스의 Repo 구경 중 .. OpenAPI 규격의 Swagger Spec 파일 존재 확인

발견한 Spec 파일을 통해 Client SDK 생성하여 개발 진행

HTTP Client 객체 직접 생성 필요 없이 SDK Method 호출로 간편 연동 ☺

05 실제 프로젝트 적용 후기(2) - 오픈소스 연동

오픈소스 연동 파이프라인



Q&A 

