

# Rest Controller와 Feign Client 각각 Genrate 및 Swagger-UI 적용

## 필요성 [↗](#)

[OpenAPI Generator Gradle Plugin 시작하기](#) 에서 작성했던 `build.gradle` 는 `rest controller` 혹은 `feign client` 둘 중 하나만 추출이 가능했습니다. 왜냐하면 `openApiGenrate` task의 `configOptions` 중 `library` 옵션에는 세가지 옵션(`spring-boot`, `spring-mvc`, `spring-cloud`)을 넣을 수 있는데 `rest controller` 는 `spring-boot` 옵션에 의해, `feign client` 는 `spring-cloud` 옵션에 의해 추출되기 때문입니다.

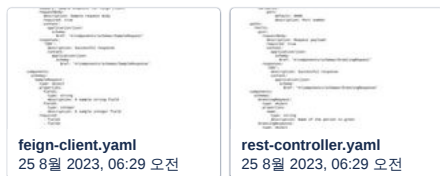
따라서 `rest controller` 와 `feign client` 를 추출하기 위한 별도의 task가 필요합니다!

또한 기존의 `application.properties` 설정을 따르면 하나의 명세서만 `swagger-ui` 를 적용할 수 있습니다.

따라서 `rest controller` 와 `feign client` 에 대한 `swagger-ui` 가 필요합니다!

## 준비 [↗](#)

- [OpenAPI Generator Gradle Plugin 시작하기](#) 숙지
- `rest controller` 명세서와 `feign client` 명세서



## build.gradle 수정 [↗](#)

### 전체 `build.gradle` [↗](#)

```
1  import org.openapitools.generator.gradle.plugin.tasks.GenerateTask
2
3  plugins {
4      id 'java'
5      id 'org.springframework.boot' version '3.1.2'
6      id 'io.spring.dependency-management' version '1.1.2'
7      // 추가
8      id "org.openapi.generator" version "6.6.0"
9  }
10
11  group = 'com.example'
12  version = '0.0.1-SNAPSHOT'
13
14  java {
15      sourceCompatibility = JavaVersion.VERSION_17
16  }
17
18  configurations {
19      compileOnly {
```

```

20     extendsFrom annotationProcessor
21 }
22 }
23
24 repositories {
25     mavenCentral()
26 }
27
28 dependencies {
29     // 추가
30     implementation 'org.springframework.boot:spring-boot-starter-web'
31     implementation 'org.springdoc:springdoc-openapi-starter-webmvc-ui:2.2.0'
32     implementation('org.openapitools:openapi-generator:6.6.0') {
33         exclude group: 'org.slf4j', module: 'slf4j-simple'
34     }
35     implementation 'org.openapitools:jackson-databind-nullable:0.2.6'
36     implementation 'org.springframework.boot:spring-boot-starter-validation'
37
38     compileOnly 'org.projectlombok:lombok'
39     developmentOnly 'org.springframework.boot:spring-boot-devtools'
40     annotationProcessor 'org.springframework.boot:spring-boot-configuration-processor'
41     annotationProcessor 'org.projectlombok:lombok'
42     testImplementation 'org.springframework.boot:spring-boot-starter-test'
43 }
44
45 tasks.named('test') {
46     useJUnitPlatform()
47 }
48
49 // 추가
50 tasks.register('openApiGenerateRestController', GenerateTask) {
51     generatorName.set("spring")
52     verbose.set(true)
53     inputSpec.set("$rootDir/src/main/resources/static/rest-controller.yaml")
54     outputDir.set("$buildDir/generate-resources")
55     apiPackage.set("com.example.openapigen.api")
56     invokerPackage.set("com.example.openapigen.invoker")
57     modelPackage.set("com.example.openapigen.model")
58     configOptions.set([
59         dateLibrary      : "java8",
60         library           : "spring-boot", //spring-boot, spring-mvc, spring-cloud
61         interfaceOnly     : "true",
62         unhandledException : "true",
63         useSpringBoot3     : "true",
64         useSpringController : "true",
65         skipDefaultInterface : "true"
66     ])
67 }
68
69 // 추가
70 tasks.register('openApiGenerateFeignClient', GenerateTask) {
71     generatorName.set("spring")
72     verbose.set(true)
73     inputSpec.set("$rootDir/src/main/resources/static/feign-client.yaml")
74     outputDir.set("$buildDir/generate-resources")
75     apiPackage.set("com.example.openapigen.fengin")
76     invokerPackage.set("com.example.openapigen.invoker")
77     modelPackage.set("com.example.openapigen.model")

```

```

78     configOptions.set([
79         dateLibrary      : "java8",
80         library          : "spring-cloud", //spring-boot, spring-mvc, spring-cloud
81         interfaceOnly    : "true",
82         unhandledException : "true",
83         useSpringBoot3    : "true",
84         //               useSpringController: "true",
85         //               skipDefaultInterface : "true"
86     ])
87 }
88
89 // 추가
90 compileJava.dependsOn tasks.named("openApiGenerateController")
91 compileJava.dependsOn tasks.named("openApiGenerateFeignClient")
92
93 // 추가
94 sourceSets {
95     main {
96         java {
97             srcDirs = ['src/main/java', 'build/generate-resources/src/main/java']
98         }
99     }
100 }
101

```

## 단계별 설명



### 단계별 설명

#### gradle task 분리

기존 `openApiGenerate` task를 지우고 아래 두 task를 추가한다.

```

1  // 추가
2  tasks.register('openApiGenerateRestController', GenerateTask) {
3      generatorName.set("spring")
4      verbose.set(true)
5      inputSpec.set("$rootDir/src/main/resources/static/rest-controller.yaml")
6      outputDir.set("$buildDir/generate-resources")
7      apiPackage.set("com.example.openapigen.api")
8      invokerPackage.set("com.example.openapigen.invoker")
9      modelPackage.set("com.example.openapigen.model")
10     configOptions.set([
11         dateLibrary      : "java8",
12         library          : "spring-boot", //spring-boot, spring-mvc, spring-cloud
13         interfaceOnly    : "true",
14         unhandledException : "true",
15         useSpringBoot3    : "true",
16         useSpringController : "true",
17         skipDefaultInterface: "true"
18     ])
19 }
20
21 // 추가
22 tasks.register('openApiGenerateFeignClient', GenerateTask) {
23     generatorName.set("spring")
24     verbose.set(true)

```

```

25     inputSpec.set("$rootDir/src/main/resources/static/feign-client.yaml")
26     outputDir.set("$buildDir/generate-resources")
27     apiPackage.set("com.example.openapigen.fengin")
28     invokerPackage.set("com.example.openapigen.invoker")
29     modelPackage.set("com.example.openapigen.model")
30     configOptions.set([
31         dateLibrary      : "java8",
32         library           : "spring-cloud", //spring-boot, spring-mvc, spring-cloud
33         interfaceOnly    : "true",
34         unhandledException : "true",
35         useSpringBoot3    : "true",
36     ])
37 }

```

#### compileJava task에 의존성 추가 [↗](#)

```

1 // 삭제
2 // compileJava.dependsOn tasks.named("openApiGenerate")
3 //추가
4 compileJava.dependsOn tasks.named("openApiGenerateController")
5 compileJava.dependsOn tasks.named("openApiGenerateFeignClient")

```

#### openApiGenerate task 실행 [↗](#)

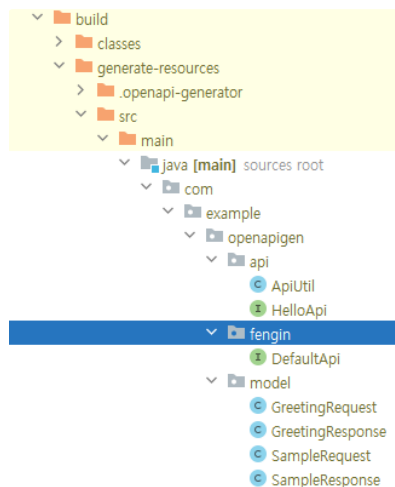
```

1 ./gradlew openApiGenerateRestController
2 ./gradlew openApiGenerateFeignClient

```

위의 명령어를 `terminal` 에서 실행해주세요

#### output 디렉토리(outputDir)로 추출된 소스 확인 [↗](#)



#### Swagger-UI 로 확인을 위해 application.properties 수정 [↗](#)

```

1 // 삭제
2 // springdoc.swagger-ui.url=/simple-api.yaml
3 // 추가
4 springdoc.swagger-ui.urls[0].name=rest-controller

```

```
5 springdoc.swagger-ui.urls[0].url=/rest-controller.yaml
6 springdoc.swagger-ui.urls[1].name=feign-client
7 springdoc.swagger-ui.urls[1].url=/feign-client.yaml
```

## Swagger - UI 확인 [↗](#)

{HOST}/swagger-ui/index.html 로 접속하여 확인

