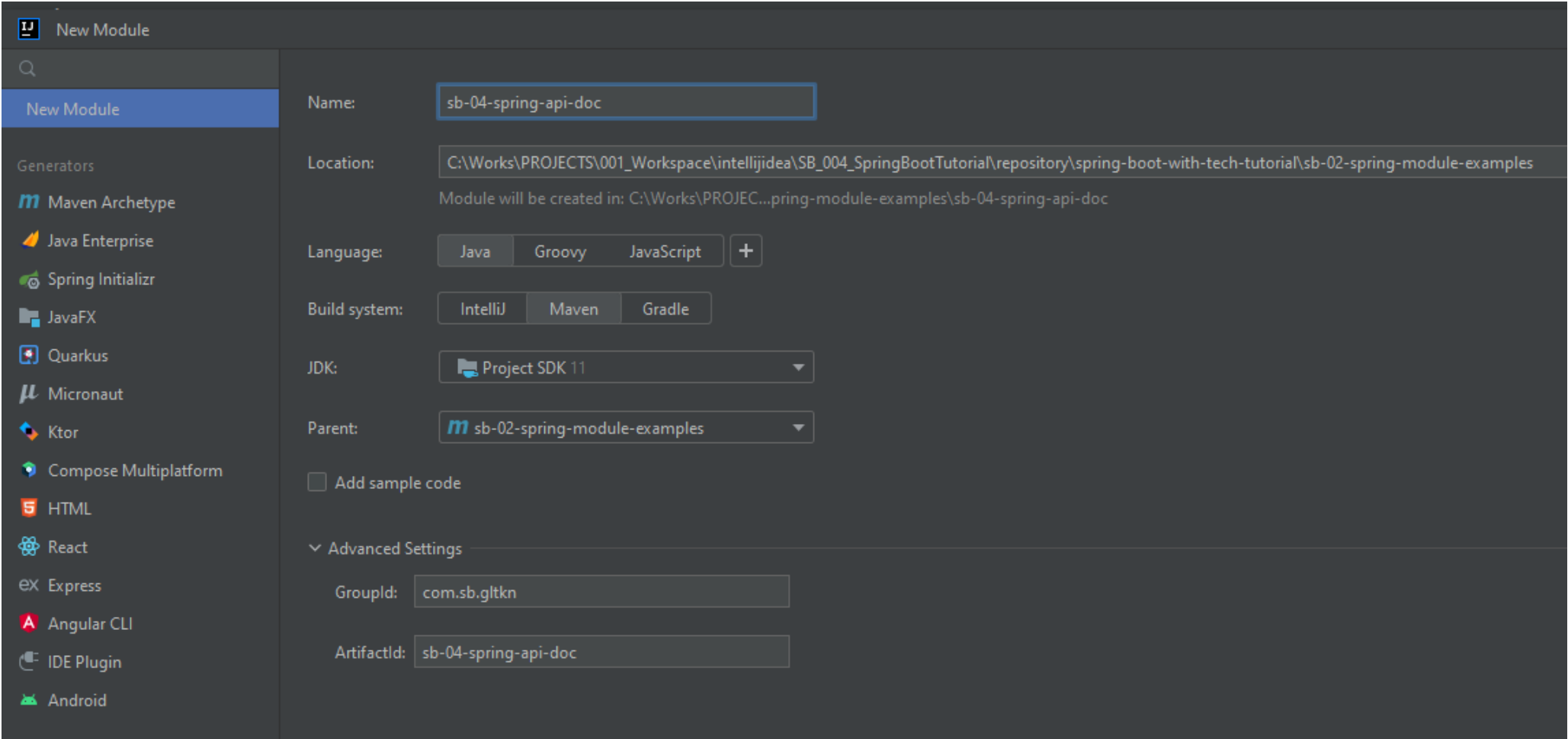
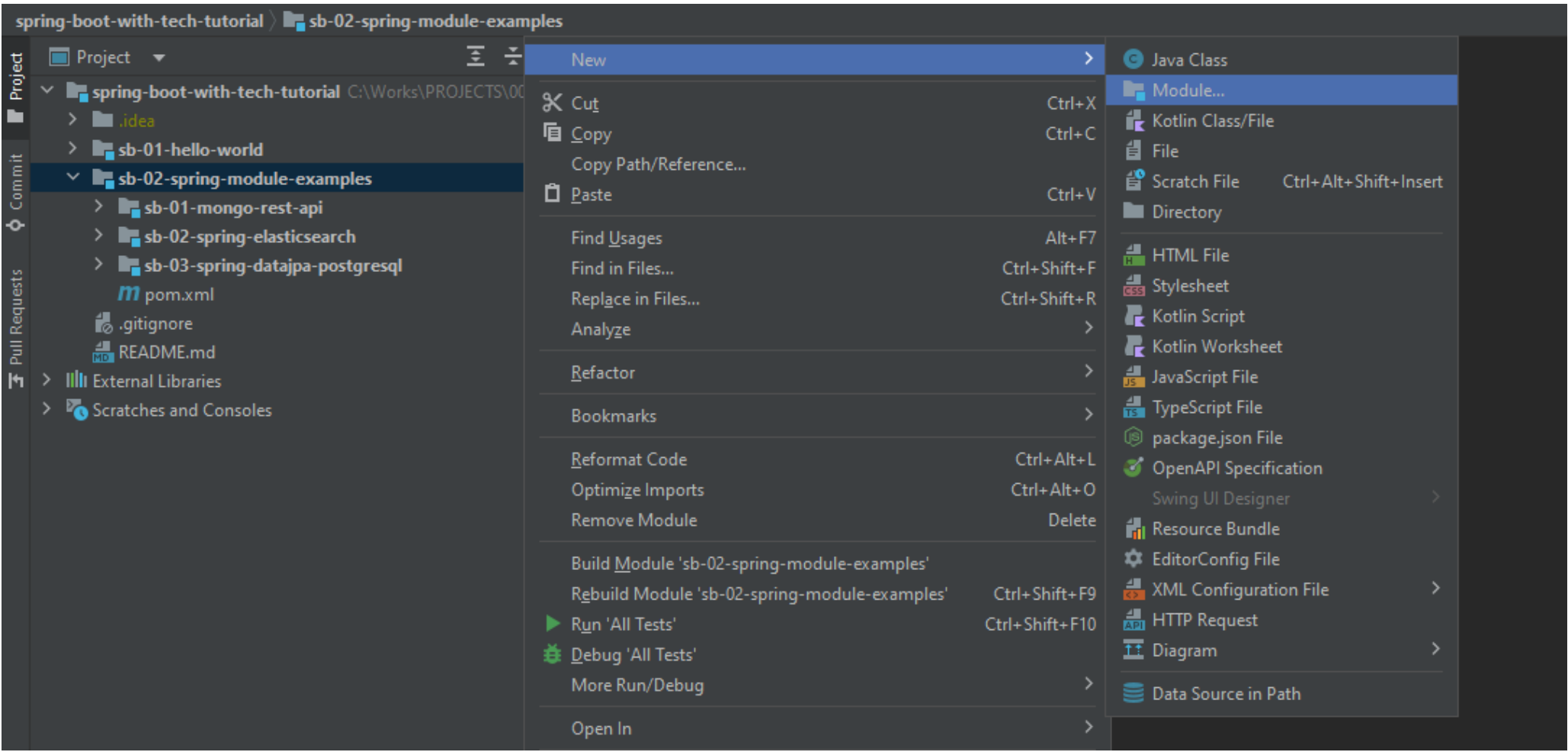
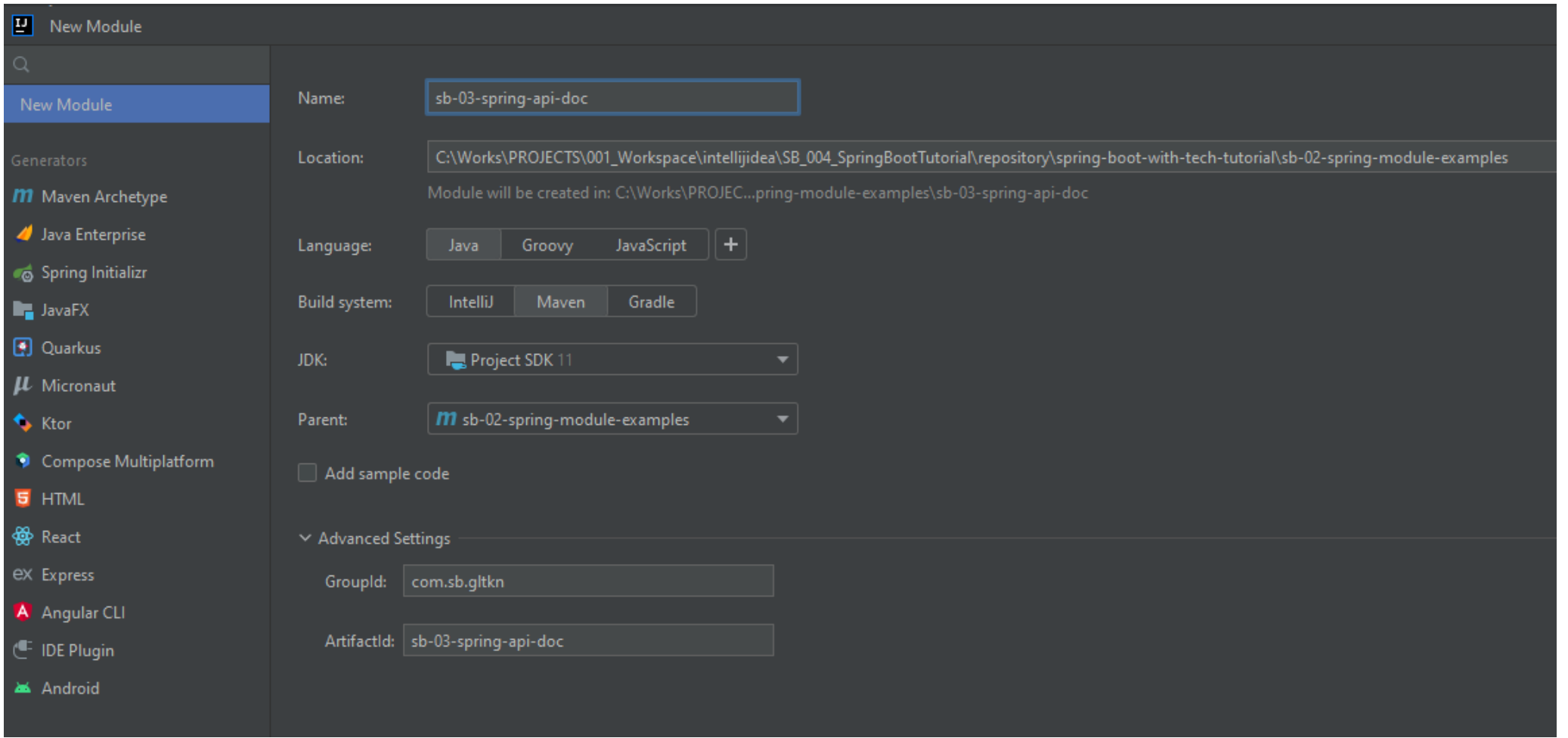


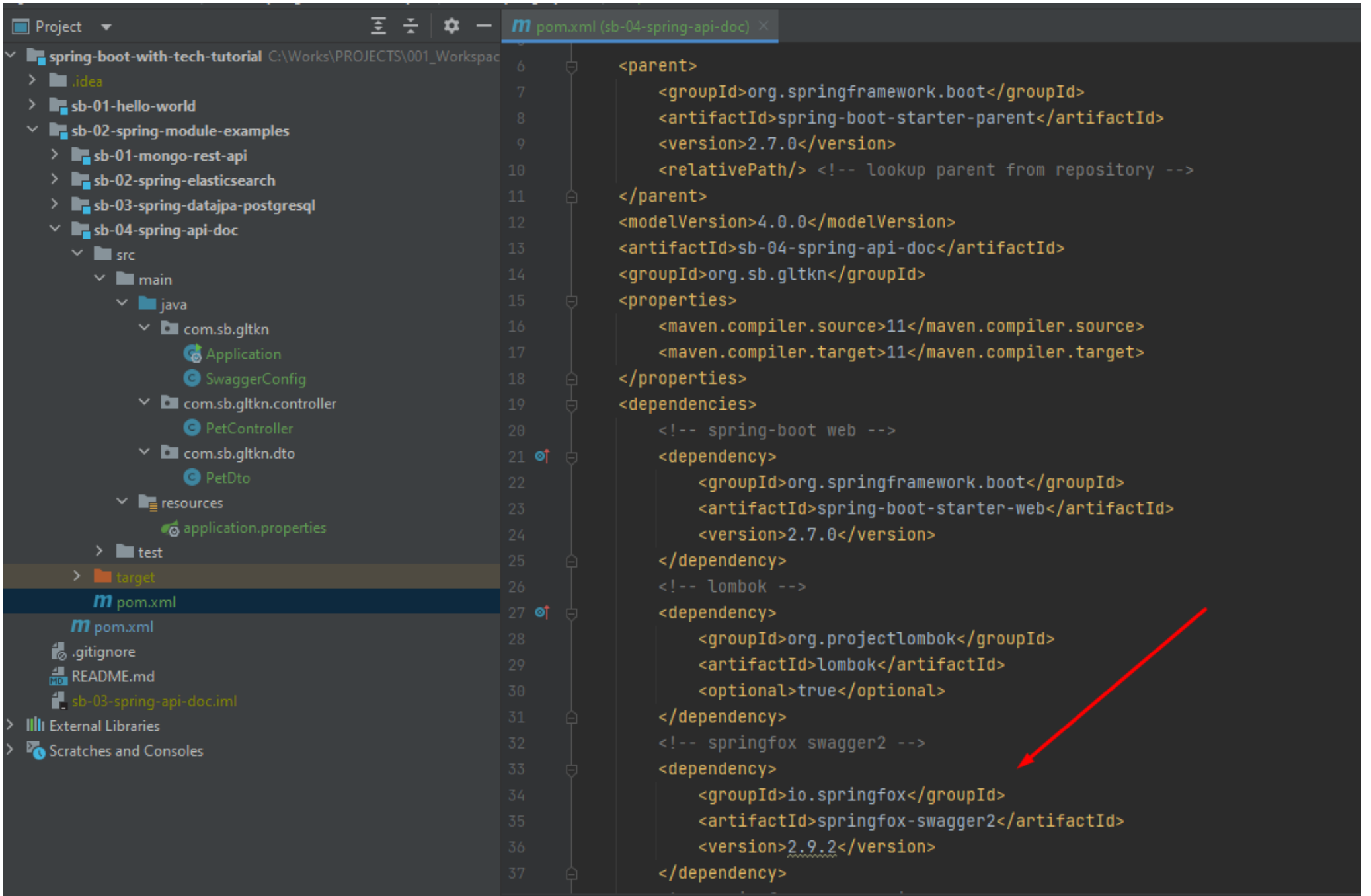
Projemizi aşağıdaki gibi oluşturalım.





Uygulamamızı yazdık.

Swagger dependency'lerini ve configürasyonlarını yaptık.



```
spring-boot-with-tech-tutorial > sb-02-spring-module-examples > sb-04-spring-api-doc > src > main > java > com > sb > gltkn > SwaggerConfig
pom.xml (sb-04-spring-api-doc) x SwaggerConfig.java x
9      import springfox.documentation.service.Contact;
10     import springfox.documentation.spi.DocumentationType;
11     import springfox.documentation.spring.web.plugins.Docket;
12     import springfox.documentation.swagger2.annotations.EnableSwagger2;
13
14     @Configuration
15     @EnableSwagger2
16     public class SwaggerConfig {
17
18         @Bean
19         public Docket api() {
20             return new Docket(DocumentationType.SWAGGER_2)
21                 .select()
22                 .apis(RequestHandlerSelectors.basePackage("com.sb.gltkn"))
23                 .paths(PathSelectors.regex(pathRegex: "/.*"))
24                 .build().apiInfo(apiEndpointsInfo());
25         }
26
27         1 usage
28         private ApiInfo apiEndpointsInfo() {
29             return new ApiInfoBuilder().title("Spring Boot Swagger Örneği")
30                 .description("Pet API Dokümantasyonu")
31                 .contact(new Contact(name: "Emre", url: "", email: ""))
32                 .licenseUrl("https://www.apache.org/licenses/LICENSE-2.0")
33                 .version("1.0.0")
34                 .build();
35         }
36     }
```

```
spring-boot-with-tech-tutorial > sb-02-spring-module-examples > sb-04-spring-api-doc > src > main > java > com > sb > gltkn > controller > PetController
pom.xml (sb-04-spring-api-doc) x SwaggerConfig.java x PetController.java x
4      import io.swagger.annotations.Api;
5      import io.swagger.annotations.ApiOperation;
6      import io.swagger.annotations.ApiParam;
7      import org.springframework.http.ResponseEntity;
8      import org.springframework.web.bind.annotation.*;
9
10     import javax.annotation.PostConstruct;
11     import java.util.ArrayList;
12     import java.util.Date;
13     import java.util.List;
14
15     @RestController
16     @RequestMapping("/pet")
17     @Api(value = "Pet API Dokümantasyonu")
18     public class PetController {
19
20         3 usages
21         private List<PetDto> petDtoList = new ArrayList<>();
22
23         @PostConstruct
24         public void init() { petDtoList.add(new PetDto(id: 1, name: "PetA", new Date())); }
25
26
27         @PostMapping
28         @ApiOperation(value = "Yeni bir Pet oluşturma fonksiyonu", notes = "Oluşturma işlemi kritik bir işlemdir.")
29         public ResponseEntity<PetDto> create(@RequestBody @ApiParam(value = "animal") PetDto petDto) {
30             petDtoList.add(petDto);
31             return ResponseEntity.ok(petDto);
32         }
33
34
35         @GetMapping
36         @ApiOperation(value = "Pet listesi fonksiyonu", notes = "Tüm pet'leri getirir.")
37         public ResponseEntity<List<PetDto>> getAllPerson() { return ResponseEntity.ok(petDtoList); }
38     }
39
40 }
```

Şimdi uygulamamızı çalıştıralım.

<http://localhost:8081/pet>

GET http://localhost:8081/pet

+

...

▶ http://localhost:8081/pet

GET

▼

http://localhost:8081/pet

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

Body

Cookies

Headers (5)

Test Results

⌐ Status: 200 OK

Pretty

Raw

Preview

Visualize

JSON ▼

≡

1

[

2

{

3

"id": 1,

4

"name": "PetA",

5

"date": "2022-06-21T20:16:22.204+00:00"

6

}

7

]

<http://localhost:8081/v2/api-docs> adresine GET isteği attığımızda bize swagger dokümantasyonunu verecektir.

← → ↺ ⬆

localhost:8081/v2/api-docs

⌵

{ "swagger": "2.0", "info": { "description": "Pet API Dokümantasyonu", "version": "1.0.0", "title": "Spring Boot Swagger Örneği", "contact": { "name": "Emre", "license": { "url": "https://www.apache.org/licenses/LICENSE-2.0" } }, "host": "localhost:8081", "basePath": "/", "tags": [{ "name": "pet-controller", "description": "Pet Controller" }], "paths": { "/pet": { "get": { "tags": ["pet-controller"], "summary": "Pet listesi", "description": "pet'leri getirir.", "operationId": "getAllPersonUsingGET", "produces": ["*"], "responses": { "200": { "description": "OK", "schema": { "type": "array", "items": { "\$ref": "#/definitions/Pet API Objesi" } }, "400": { "description": "Unauthorized", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "401": { "description": "Unauthorized", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "403": { "description": "Forbidden", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "404": { "description": "Not Found", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" } } }, "post": { "tags": ["pet-controller"], "summary": "Yeni bir Pet oluşturma fonksiyonu", "description": "Yeni bir Pet oluşturma fonksiyonu", "operationId": "createUsingPOST", "consumes": ["application/json"], "produces": ["*"], "parameters": [{ "in": "body", "name": "petDto", "description": "animal", "required": true, "schema": { "type": "object", "properties": { "date": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "id": { "type": "integer", "format": "int32", "description": "Pet Objesi Id" }, "name": { "type": "string", "description": "Pet Objesi Name" } }, "title": "Pet API Objesi", "description": "API modelidir" } } }] }, "responses": { "200": { "description": "OK", "schema": { "\$ref": "#/definitions/Pet API Objesi" } }, "201": { "description": "Created", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "400": { "description": "Unauthorized", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "401": { "description": "Unauthorized", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "403": { "description": "Forbidden", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "404": { "description": "Not Found", "schema": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" } } } }, "definitions": { "Pet API Objesi": { "type": "object", "properties": { "date": { "type": "string", "format": "date-time", "description": "Pet Objesi Date" }, "id": { "type": "integer", "format": "int32", "description": "Pet Objesi Id" }, "name": { "type": "string", "description": "Pet Objesi Name" } }, "title": "Pet API Objesi", "description": "API modelidir" } } } }

Postman ile bakalım.

GET http://localhost:8081/pet

GET http://localhost:8081/v2/api-docs

+

...

Untitled Request

GET

http://localhost:8081/v2/api-docs

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"swagger": "2.0",

3

"info": {

4

"description": "Pet API Dokümantasyonu",

5

"version": "1.0.0",

6

"title": "Spring Boot Swagger Örneği",

7

"contact": {

8

"name": "Emre"

9

},

10

"license": {

11

"url": "https://www.apache.org/licenses/LICENSE-2.0"

12

}

13

},

14

"host": "localhost:8081",

15

"basePath": "/",

16

"tags": [

17

{

18

"name": "pet-controller",

Yukarıdaki Response:

```
{
  "swagger": "2.0",
  "info": {
    "description": "Pet API Dokümantasyonu",
    "version": "1.0.0",
    "title": "Spring Boot Swagger Örneği",
    "contact": {
      "name": "Emre"
    },
    "license": {
      "url": "https://www.apache.org/licenses/LICENSE-2.0"
    }
  },
  "host": "localhost:8081",
  "basePath": "/",
  "tags": [
    {
      "name": "pet-controller",
      "description": "Pet Controller"
    }
  ],
  "paths": {
    "/pet": {
      "get": {
        "tags": [
          "pet-controller"
        ],
        "summary": "Pet listesi fonksiyonu",
        "description": "Tüm pet'leri getirir.",
        "operationId": "getAllPersonUsingGET",
        "produces": [
          "*/*"
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Pet API Objesi"
              }
            }
          },
          "401": {
            "description": "Unauthorized"
          }
        }
      }
    }
  }
}
```

```

        },
        "403": {
            "description": "Forbidden"
        },
        "404": {
            "description": "Not Found"
        }
    }
},
"post": {
    "tags": [
        "pet-controller"
    ],
    "summary": "Yeni bir Pet oluşturma fonksiyonu",
    "description": "Oluşturma işlemi kritik bir işlemdir.",
    "operationId": "createUsingPOST",
    "consumes": [
        "application/json"
    ],
    "produces": [
        "*/*"
    ],
    "parameters": [
        {
            "in": "body",
            "name": "petDto",
            "description": "animal",
            "required": true,
            "schema": {
                "$ref": "#/definitions/Pet API Objesi"
            }
        }
    ],
    "responses": {
        "200": {
            "description": "OK",
            "schema": {
                "$ref": "#/definitions/Pet API Objesi"
            }
        },
        "201": {
            "description": "Created"
        },
        "401": {
            "description": "Unauthorized"
        },
        "403": {
            "description": "Forbidden"
        },
        "404": {
            "description": "Not Found"
        }
    }
}
},
}
},
"definitions": {
    "Pet API Objesi": {
        "type": "object",
        "properties": {
            "date": {
                "type": "string",
                "format": "date-time",
                "description": "Pet Objesi Date"
            },
            "id": {
                "type": "integer",
                "format": "int32",
                "description": "Pet Objesi Id"
            },
            "name": {
                "type": "string",
                "description": "Pet Objesi Name"
            }
        },
        "title": "Pet API Objesi",
        "description": "API modelidir"
    }
}
}
}

```


Swagger UI ekleyelim.

```
m pom.xml (sb-04-spring-api-doc) x
26      <!-- lombok -->
27      <dependency>
28          <groupId>org.projectlombok</groupId>
29          <artifactId>lombok</artifactId>
30          <optional>true</optional>
31      </dependency>
32      <!-- springfox swagger2 -->
33      <dependency>
34          <groupId>io.springfox</groupId>
35          <artifactId>springfox-swagger2</artifactId>
36          <version>2.9.2</version>
37      </dependency>
38      <!-- springfox swagger-ui -->
39      <dependency>
40          <groupId>io.springfox</groupId>
41          <artifactId>springfox-swagger-ui</artifactId>
42          <version>2.9.2</version>
43      </dependency>
44  </dependencies>
45
```

Şimdi uygulamamızı çalıştıralım.

<http://localhost:8081/swagger-ui.html>

