

Elaboro	Documento	Versión	Descripción	Fecha
Luis Sánchez Martínez	Práctica Crear proyecto Spring Boot con OpenAPI	1.0	Creación de documento	22/03/2024

Práctica Crear proyecto Spring Boot con documentación de OpenAPI.

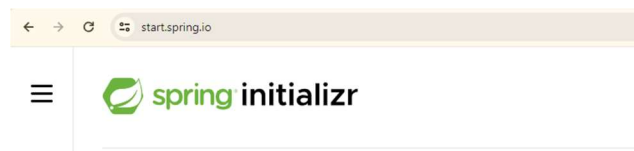
Tiempo estimado: 1 hora.

Uso de:

- Spring Initializr
- IntelliJ
- OpenAPI

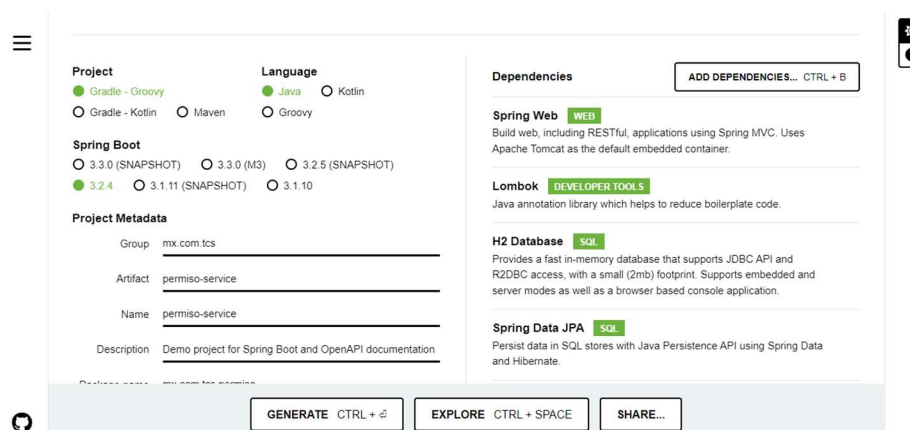
Generar proyecto Spring Boot en sitio spring initializr

1. Consultar la página: <https://start.spring.io/>

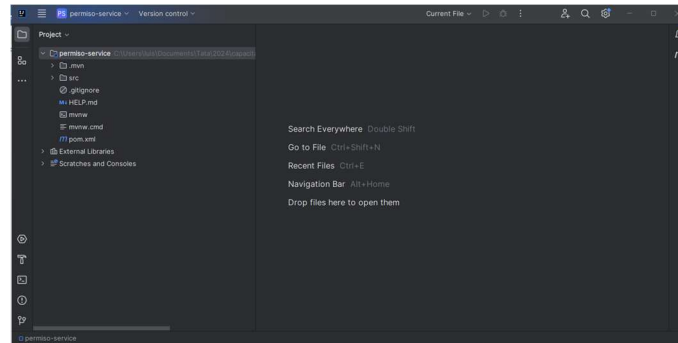
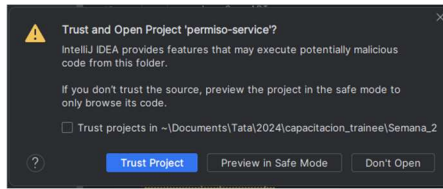


2. Configurar un proyecto Maven, seleccionando las librerías:
 - a. Spring Web
 - b. Lombok
 - c. H2 Database
 - d. Spring Data JPA

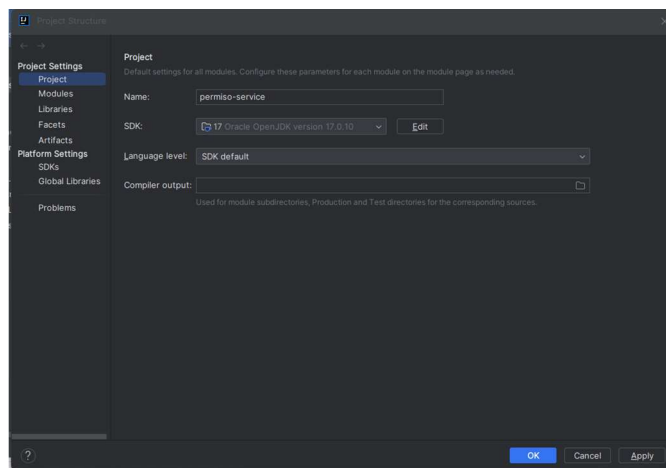
y dar clic en la opción *Generate*:



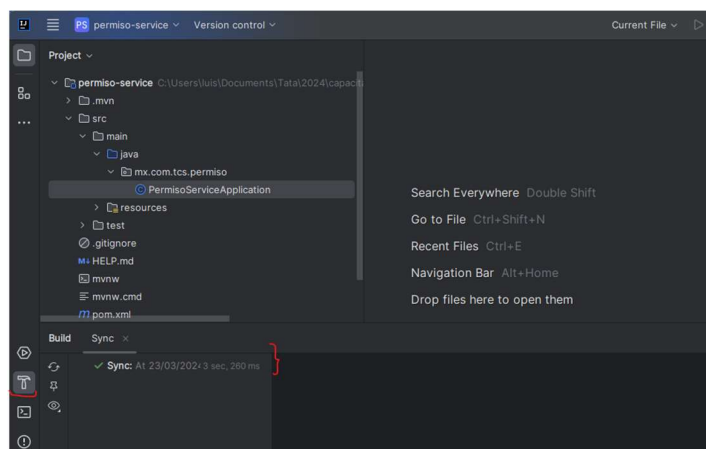
3. Abrir proyecto en IDE IntelliJ al dar clic en *Trust Project*:



4. Configurar SDK a Java 17 del proyecto al dar clic en el botón de settings y Project Structure:



5. Construir el proyecto dando clic en el botón martillo.



- Agregar la clase Controller y el agregar el primer endpoint GET /permiso de la API catpermiso-service.

```
PermisoController.java
32 public PermisoController(IPermisoService permisoService) {
33     this.permisoService = permisoService;
34 }
35
36 /**
37  * Endpoint used to get all records in Permiso entity.
38  * @return a ResponseEntity of the list of Permiso object.
39  */
40 no usages
41 @RequestMapping("/permiso")
42 public ResponseEntity<List<Permiso>> getAll() {
43     return permisoService.listAll();
44 }
45 }
```

- Construir el proyecto y ejecutarlo desde el IDE:

```
PermisoController.java
32 public PermisoController(IPermisoService permisoService) {
33     this.permisoService = permisoService;
34 }
35
36 /**
37  * Endpoint used to get all records in Permiso entity.
38  * @return a ResponseEntity of the list of Permiso object.
39  */
40 no usages
41 @RequestMapping("/permiso")
42 public ResponseEntity<List<Permiso>> getAll() {
43     return permisoService.listAll();
44 }
45 }
```

- Validar el endpoint /permiso con la función GET ejecutado desde postman:

testMicroservicePermisos / getPermiso

GET <http://localhost:8080/permiso> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 571 ms Size: 303 B Save as example

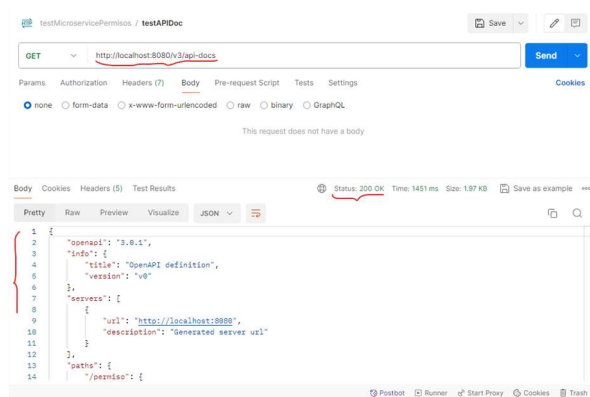
Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "nombre": "Academico",
5     "descripcion": "Presentacion de examen profesional",
6     "activo": 1,
7     "idPadre": 0,
8     "icono": "/images/icon_school.gif"
9   }
}
```

9. Integrar la librería springdoc en el proyecto:

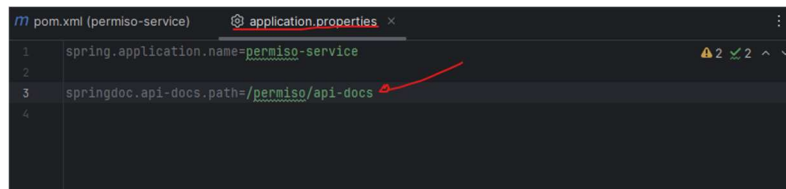
```
10 <groupId>com.h2database</groupId>
11 <artifactId>h2</artifactId>
12 <scope>runtime</scope>
13 </dependency>
14 <dependency>
15 <groupId>org.projectlombok</groupId>
16 <artifactId>lombok</artifactId>
17 <optional>true</optional>
18 </dependency>
19
20 <dependency>
21 <groupId>org.springdoc</groupId>
22 <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
23 <version>2.2.0</version>
24 </dependency>
```

10. Validar la información de OpenAPI consultando la URL: <http://localhost:8080/v3/api-docs> después de correr el proyecto:



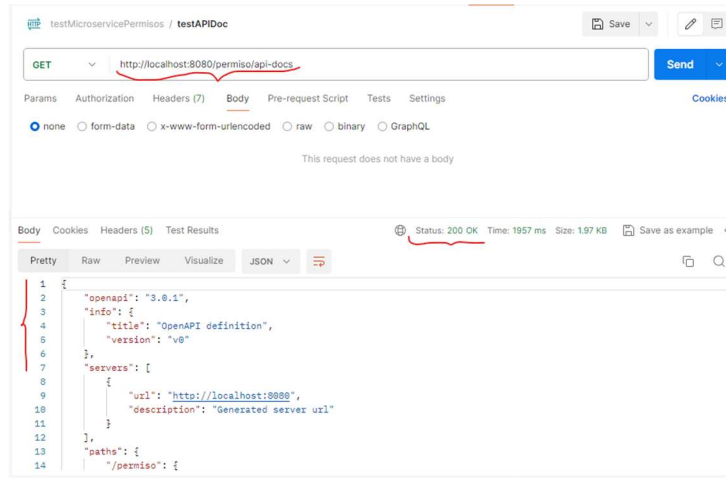
```
1 {
2   "openapi": "3.0.1",
3   "info": {
4     "title": "OpenAPI definition",
5     "version": "v0"
6   },
7   "servers": [
8     {
9       "url": "http://localhost:8080",
10      "description": "Generated server url"
11    },
12  ],
13  "paths": {
14    "/permiso": {
```

11. Actualizar el Path usado para acceder a la documentación de OpenAPI usando la propiedad `springdoc.api-docs.path=/api-docs`:

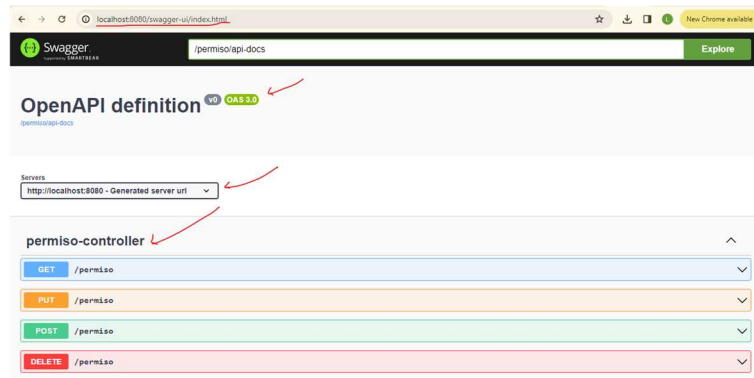


```
1 spring.application.name=permiso-service
2
3 springdoc.api-docs.path=/permiso/api-docs
4
```

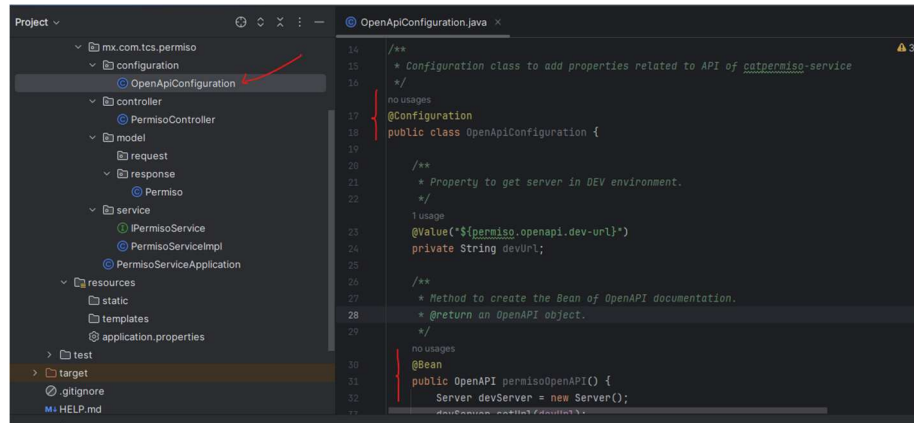
12. Validar la información de OpenAPI:



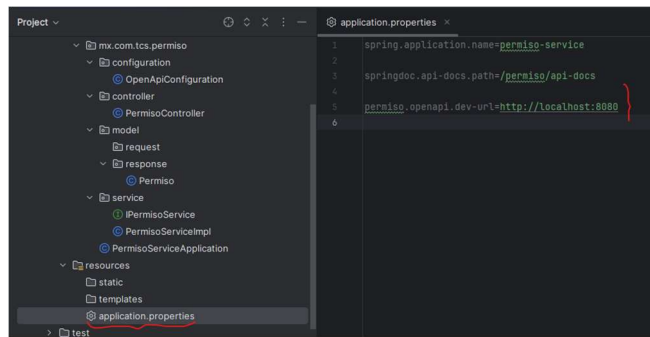
13. Para consultar el formato en html debe de ser desde un navegador web consultando la ruta:
<http://localhost:8080/swagger-ui/index.html>



14. Crear la clase Configuration para integrar la documentación de OpenAPI en el proyecto agregando información de las propiedades:
- Titulo
 - Versión
 - Contacto
 - Descripción
 - Servers

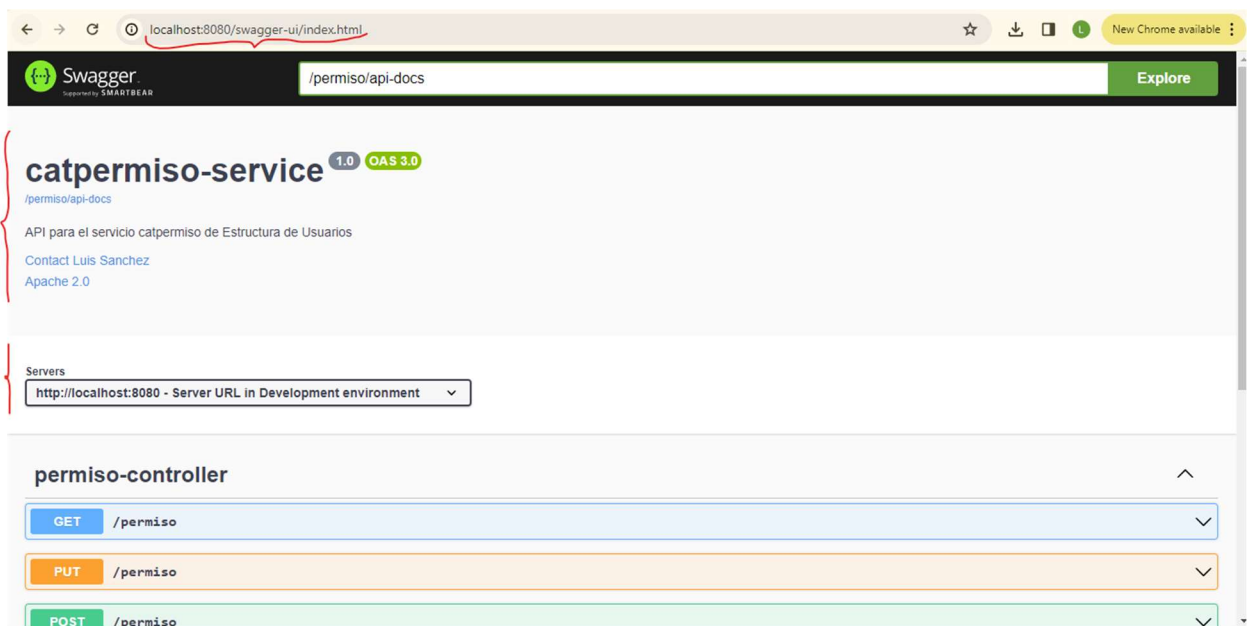


15. Validar la propiedad integrada con el valor del server en el ambiente DEV:



16. Consultar la documentación en formato html desde la ruta:

<http://localhost:8080/swagger-ui/index.html>



Referencias

<https://www.baeldung.com/spring-rest-openapi-documentation>

<https://www.baeldung.com/spring-boot-json>

<https://www.bezkoder.com/spring-boot-swagger-3/>