

U7A1 Documentación Rest API

```
OrganizationController.java 5 x
apiserver > src > main > java > com > xyz > openapi > server > controller > OrganizationController.java > Language Support for Java(TM) by Red Hat > {} com.xyz
12 import io.swagger.v3.oas.annotations.tags.Tag;
13 import org.springframework.beans.BeanUtils;
14 import org.springframework.web.bind.annotation.*;
15
16 import java.util.ArrayList;
17 import java.util.List;
18 import java.util.Optional;
19 import java.util.UUID;
20 import java.util.stream.Collectors;
21
22 @RestController
23 @RequestMapping("/api")
24 @Tag(name = "Organization Controls", description = "API Endpoints to operate on Employee and Departments")
25 public class OrganizationController {
26     private List<Employee> employees = new ArrayList<>();
27     private List<Department> departments = new ArrayList<>();
28
29     @GetMapping(value = "/employees")
30     @Operation(summary = "If departmentId is not passed, get all employees. Otherwise get employees from specific depa")
31     @ApiResponses({
32         @ApiResponse(responseCode = "200", content = {
33             @Content(schema = @Schema(implementation = EmployeeList.class))
34         }),
35         @ApiResponse(responseCode = "500", content = {
36             @Content(schema = @Schema(implementation = ErrorResponse.class))
37         })
38     })
39     public EmployeeList getAllEmployees(@RequestParam(required = false) String deptId) {
40         List<Employee> employees = this.employees.stream()
41             .filter(emp -> deptId == null ||
42                 (deptId != null && emp.getDepartment() != null && emp.getDepartment().getId().equals(deptId)))
43             .collect(Collectors.toList());
44         return EmployeeList.of(employees);
45     }
46
47     @GetMapping(value = "/employees/{id}")
```

```

@GetMapping(value = "/employees/{id}")
public Employee getEmployee(@PathVariable String id) {
    Optional<Employee> optional = employees.stream()
        .filter(emp -> emp.getId().equals(id))
        .findAny();
    if(optional.isEmpty()) {
        throw new IllegalArgumentException("Employee does not exist for id: "+id);
    }
    return optional.get();
}

@PostMapping(value = "/employees")
public String createEmployee(@RequestBody Employee emp){
    emp.setId(UUID.randomUUID().toString());
    employees.add(emp);
    return emp.getId();
}

@PutMapping(value = "/employees")
public String updateEmployee(Employee updatedEmp){
    employees.stream()
        .filter(e -> updatedEmp.getId().equals(e.getId()))
        .findFirst()
        .map(emp -> {
            BeanUtils.copyProperties(updatedEmp, emp);
            return emp;
        })
        .orElseThrow();
    return updatedEmp.getId();
}

```

```

@GetMapping(value = "/departments")
public List<Department> getAllDepartments() {
    return departments;
}

@GetMapping(value = "/departments/{id}")
public Department getDepartment(@PathVariable String id) {
    Optional<Department> optional = departments.stream()
        .filter(emp -> emp.getId().equals(id))
        .findAny();
    if(optional.isEmpty()) {
        throw new IllegalArgumentException("Department does not exist for id: "+id);
    }
    return optional.get();
}

@PostMapping(value = "/departments")
public String createDepartment(Department department){
    department.setId(UUID.randomUUID().toString());
    departments.add(department);
    return department.getId();
}

```

Añadimos dependencia de OpenAPI

```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-web:2.3.1.RELEASE'  
    implementation 'org.springframework:spring-context:5.2.7.RELEASE'  
    implementation 'org.reflections:reflections:0.9.12'  
    implementation 'org.springdoc:springdoc-openapi-ui:1.5.8'
```

```
build.gradle M X  
piserver > build.gradle > plugins  
1  plugins {  
2      id "java"  
3      targetCompatibility = "11"  
4      sourceCompatibility = "11"  
5      id "org.springframework.boot" version "2.3.1.RELEASE"  
6      id "io.spring.dependency-management" version "1.0.9.RELEASE"  
7      id "org.openapi.generator" version "5.1.0"  
8  }  
9  
10  
11  repositories {  
12      mavenCentral()  
13  }  
14  
15  dependencies {  
16      implementation 'org.springframework.boot:spring-boot-starter-web:2.3.1.RELEASE'  
17      implementation 'org.springframework:spring-context:5.2.7.RELEASE'  
18      implementation 'org.reflections:reflections:0.9.12'  
19      implementation 'org.springdoc:springdoc-openapi-ui:1.5.8'  
20  
21  
22      testImplementation 'org.junit.jupiter:junit-jupiter-api:5.7.0'  
23      testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine:5.7.0'  
24  }  
25  
26  test {  
27      useJUnitPlatform()  
28  }
```

```
build.gradle 2, M •
apiserver > build.gradle > openApiGenerate
15 dependencies {
23     testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine:5.7.0'
24 }
25
26 test {
27     useJUnitPlatform()
28 }
29
30 bootJar {
31     mainClassName = 'com.xyz.openapi.server.Application'
32 }
33
34 openApiGenerate {
35     inputSpec.set("${rootDir}/api-docs.json")
36     outputDir.set("${rootDir}/thinclient")
37
38     groupId.set("com.xyz.openapi")
39     id.set("openapi-thinclient")
40     version.set("0.0.1-SNAPSHOT")
41
42     apiPackage.set("com.xyz.openapi.thinclient.api")
43     invokerPackage.set("com.xyz.openapi.thinclient.invoker")
44     modelPackage.set("com.xyz.openapi.thinclient.model")
45
46     generatorName.set("java");
47     library.set("resttemplate")
48     configOptions.put("serializationLibrary", "jackson")
49 }
```

Configuración

```
package com.xyz.openapi.server.config;

import io.swagger.v3.oas.models.OpenAPI;
import io.swagger.v3.oas.models.info.Info;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class SwaggerConfiguration {

    @Bean
    public OpenAPI openAPI() {
        Contact contact = new Contact();
        contact.setEmail("help@xyz.com");
        contact.setName("XYZ Support");
        contact.setUrl("http://www.xyz.com");
        return new OpenAPI()
            .info(new Info().title(title:"Employee APIs").description(description:"Description here..")
                .version(version:"1.0").contact(contact)
                .license(new License().name("2015-2021 XYZ LLC All Rights Reserved")));
    }
}
```