Migración SICOF ERP - Proceso: Guía de Documentación de Microservicios

La siguiente sección define las configuraciones que se deben tener presente en el desarrollo de los microservicios de la fábrica de desarrollo para la documentación de los microservicios.

Consideraciones Previas

- Los microservicios deben aplicar las configuraciones definidas en la mayoria de los casos a todas las clases que definan la lógica del negocio como Entidades, Controladores, Configuraciones etc.
- Si un microservicio requiere configuraciones especiales, estas deben ser validadas con los lideres de desarrollo (Pablo Quintana, Daberson Henao, Carlos Torres, Gersain Castañeda).

Información Importante

Se ha definido para el proces ode documentación de los microservcios el estandar OpenAPI Specification (OAS) 3.0.3. e[1] la cual define una descripción de interfaz estándar, independiente del lenguaje de programación para las API REST, que permite que tanto humanos como computadoras descubran y comprendan las capacidades de un servicio sin requerir acceso al código fuente, documentación adicional o inspección del tráfico de red. Cuando se define adecuadamente a través de OpenAPI, un consumidor puede comprender e interactuar con el servicio remoto con una cantidad mínima de lógica de implementación. Similar a lo que han hecho las descripciones de interfaz para la programación de nivel inferior, la especificación OpenAPI elimina las conjeturas al llamar a un servicio.

Documentación de Soporte

En la siguiente web se relacionan la forma de implementación de la especificación OAS[2] con la tecnología Sprintboot Documentación springdoc

Configuraciones del Microservicio

Todo microservicio que implemente documentación OAS[3] debe incluir en su archivo de configuración Maven **POM**[4] la dependencia del proyecto del Dominio de clases y entidades Comunes el cual se describe a continuación:

```
<dependency>
  <groupId>co.ada.models</groupId>
  <artifactId>ModelsClassesADA</artifactId>
  <version>0.0.1-SNAPSHOT</version>
```

```
</dependency>
```

El desarrollador debe utilizar la versión más reciente del proyecto la cual puede ser consultada en la sección Versionamiento de APIs

Como implementar documentación OAS con el Dominio de Clases de Entidades Comunes

Para usar el Dominio de Clases de Entidades Comunes en el proceso de documentación siga los siguientes pasos:

Paso 1

Registre la dependencia actualizada del Dominio de Clases de Entidades Comunes.

Paso 2

Cree un paquete llamado config

Paso 3

Cree la clase de configuración como se muestra continuación.

```
package co.ada.test.microservicio.pruebamodel.config;

import org.springframework.context.annotation.Configuration;

import co.ada.doc.config.SpringOpenApiConfigBase;

@Configuration
public class MySpringOpenApiConfig extends SpringOpenApiConfigBase {}
```

Paso 4

Configure en el bootstrap.yml del proyecto los parametros de documentación del microservicio. Apoyese del siguiente modelo de referencia extraido del proyecto modelo.

```
openapi-document:
   info:
      title: "Prueba Model ADA API"
      description: "Ejemplo de Microservicio utilizando el Dominio de clases
y entidades comunes"
```

```
      termsOfService: ""
      contact:
         name: "Carlos Torres"
         url: "http://ada.co"
         email: "carlos.torres@ada.co"
      license:
         name: "ADA 1.0"
         url: ""
      version: "0.0.1"
   servers:
      public:
         url: "http://zuul-mig.adacsc.co/api/prueba"
         description: "Development integration public server"
   externalDocs:
      description: ""
      url: ""
```

Paso 5

Documente las clases del microservicio con la especificación OAS[5)]. Apoyese del proyecto modelo el cual tiene un ejemplo de configuración.

```
//EJEMPLO CONTROLADOR
package co.ada.test.microservicio.pruebamodel.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import co.ada.core.controller.AdaController;
import co.ada.models.microservicio.usuario.Employee;
import co.ada.models.microservicio.usuario.Usuario;
import co.ada.test.microservicio.pruebamodel.service.EmployeeService;
import co.ada.test.microservicio.pruebamodel.service.UserService;
import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.tags.Tag;

@Tag(name = "Controlador Servicios", description = "Controlador de prueba de
Servicios de ejemplo")
@RestController
public class ServiciosController extends AdaController{

    @Autowired
    private EmployeeService employeeService;
```

```
    @Autowired
    private UserService usuarioServicie;

    @Operation(summary = "Consultar Empleados", description = "Ejemplo de
documentación de servicio de lista de empleados")
    @GetMapping(value = "employee")
    public ResponseEntity<List<Employee>> getEmployees() {
        return
ResponseEntity.status(HttpStatus.ACCEPTED).body(employeeService.getEmployees
());
    }

    @Operation(summary = "Consultar Usuarios", description = "Ejemplo de
documentación de servicio de lista de usuarios")
    @GetMapping(value = "usuarios")
    public ResponseEntity<List<Usuario>> getUsuarios() {
        return
ResponseEntity.status(HttpStatus.ACCEPTED).body(usuarioServicie.getUsuarios(
));
    }
}
```

```
//EJEMPLO ENTIDAD
package co.ada.models.microservicio.usuario;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;

import io.swagger.v3.oas.annotations.media.Schema;

@Schema(
         name = "Rol",
         description = "Clase entidad que contiene la definición de la tabla
Rol"
    )
@Entity
@Table(name="roles")
public class Role implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 6595145379859167872L;
```

```
    @NotNull
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotNull
    @Column(unique = true, length = 128)
    private String nombre;

    public Role() {}

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

Recomendaciones

- Toda documentación debe ser clara y precisa.
- Toda entidad debe estar documentada en la especificación OAS para ser incluida en el Dominio de clases y Entidades Comunes.
- Todo Controller debe ser documentado mínimo en todas las operaciones expuestas.
- Cada operación expuesta debe documentar los códigos de retorno generados en el consumo.

←Volver atrás

1) https://github.com/OAI/OpenAPI-Specification

2) 3) 5) https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.3.md

4) https://www.tutorialspoint.com/maven/maven_pom.htm