OpenApi (Swagger): Documentación de una API

Gabriel Rodríguez Flores

December 15, 2021

ADA ITS
INSTITUTO TECNOLÓGICO SUPERIOR

- Aprender a realizar la documentación técnica de una API

Contents

1 Teoría

1.1 OpenAPI (Swagger)

Introducción

- ¿Swagger o OpenAPI?
- Documentación oficial
- Swagger Specification
  - OpenAPI Specification
    * GitHub

Documentación

- Página inicio documentación oficial
  - Documentación
  - Buenas prácticas

Tutoriales

- Manera tradicional (con comentarios)
- Tutorial con schemas y autorizacion
- Tutorial minimalista de integración
- Tutorial de integración con NodeJS

Herramientas

- Lista de herrramientas y frameworks
- Editor online

1.1.1 Conceptos

- Path
- Parameters (Query, path, body)
- Responses
- Formats
- oneOf, anyOf and allOf
- ReadOnly / WriteOnly
- Discrimitators
- Security

- Examples
- Tags

2 Ejemplos

- Ejemplo PetStore (UI)
- Ejemplo PetStore (JSON)

2.1 Ejemplos básicos

- Ejemplo TIC TAC TOE

2.2 Ejemplos complejos

2.2.1 Ejemplo herencia

```yaml
openapi: 3.0.1
info:
  title: Herency Example
  version: 1.0.0
paths:
  /list:
    get:
      responses:
        '200':
          description: Get a list of items
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/List'
components:
  schemas:
    ListBase:
      type: object
      properties:
        page:
          type: number
    ItemBase:
      type: object
      properties:
        name:
          type: string
    Item1:
      allOf:
```

```yaml
        - $ref: '#/components/schemas/ItemBase'
        - type: object
          properties:
            color:
              type: string
    Item2:
      allOf:
        - $ref: '#/components/schemas/ItemBase'
        - type: object
          properties:
            color:
              type: string
    List:
      allOf:
        - $ref: '#/components/schemas/ListBase'
        - type: object
          properties:
            results:
              type: array
              items:
                anyOf:
                  - $ref: '#/components/schemas/Item1'
                  - $ref: '#/components/schemas/Item2'
```

2.3 Division y estrucuturación en Ficheros

- Ejemplo con swagger-jsdoc
- Ejemplo con express-openpi

2.3.1 Base

> Nota: `parserYamlFolder` recoge todos los ficheros YML de una carpeta dada y los transforma en JSON

```js
const jsYaml = require('js-yaml');
const path = require('path');
const fs = require('fs');

function parseYamlFolder(folder) {
  const jsonFlat = fs.readdirSync(path.join(__dirname, folder)).reduce((acc, file) => {
    const json = jsYaml.safeLoad(fs.readFileSync(path.resolve(__dirname, folder, file), 'utf8'));
    return {
      ...acc,
      ...json,
    };
```

```
  }, {});
  return jsonFlat;
}

module.exports = {
  apiDoc: {
    openapi: '3.0.2',
    info: {
      title: 'My Awesome App',
      version: '0.0.1',
    },
    components: {
      schemas: parseYamlFolder('./schemas'),
      responses: parseYamlFolder('./responses'),
      securitySchemes: parseYamlFolder('./security'),
      examples: parseYamlFolder('./examples'),
    },
    security: [],
    paths: {},
  },
  getDoc: filename => jsYaml.safeLoad(fs.readFileSync(path.resolve(path
      .dirname(filename), `${path.basename(filename, '.js')}.yml`), '
      utf8')),
};
```

2.3.2 Paths

2.3.3 Schemas

2.3.4 Responses

2.3.5 Security

2.3.6 Examples

3 Ejercicios

4 Entregables

4.1 En clase

4.2 Tarea

4.3 Trabajo

Añadir documentación Swagger al proyecto de notas