

Documentar tu API con Swagger

Las API son las maneras más populares de comunicar dos aplicaciones distintas. La herramienta swagger es una herramienta de las más populares para documentar un API.

Es una herramienta dedicada para el manejo de las APIs's REST por lo cual nos ofrece lo siguiente:

- Documentación Manual
- documentación automatizada
- Generación de pruebas de caso de uso
- Interfaz en la nube
- Generación de Código

Ejemplo: Tengo una pagina web de una agencia de viajes

Tengo lo siguiente:

MI API DE AGENCIA DE VIAJES



URL Base: <https://api.moontravels/v1/clientes/>

Insertar un nuevo cliente en la BD

Obtener un cliente en la BD

- URL: <https://api.moontravels/v1/clientes/>
- Método: POST
- Parámetros:
 - token autenticación** (header)
 - datos personales** (body)
- Respuesta:
 - 200:** idCliente
 - 400 - 500:** mensaje error

- URL: <https://api.moontravels/v1/clientes/idCliente>
- Método: GET
- Parámetros:
 - token autenticación** (header)
 - idCliente** (url)
- Respuesta:
 - 200:** idCliente
 - 400 - 500:** mensaje error



La documentación en Swagger se basa en un sintaxis archivo YAML

SINTAXIS ARCHIVO YAML



propiedad valor



```
1 openapi: 3.0.1
2 info:
3   title: API de agencia de viajes MoonTravels.
4   description: Esta API se encarga de manejar la información de la empresa
5               MoonTravels.
6   version: 1.0.0
7 servers:
8 - url: https://api.moontravels/v1/clientes/
9 tags:
10 - name: clientes
    description: Maneja la información de los clientes de MoonTravels
```

string, number, integer, boolean, array



Sintaxis :

- Línea 1: Propiedad y Valor
- Cuando una propiedad esta dentro de otra tenemos un objeto(cuando un objeto esta dentro de otro y nos encontramos un guion medio a principio significa que pertenece a una lista.

```
7 - url: https://api.moontravels/v1/clientes/
8 tags:
9 - name: clientes
10 description: Maneja la información de los cl
```

Tipos de valores que podemos tener

String: Cadena

Number: números enteros y decimales

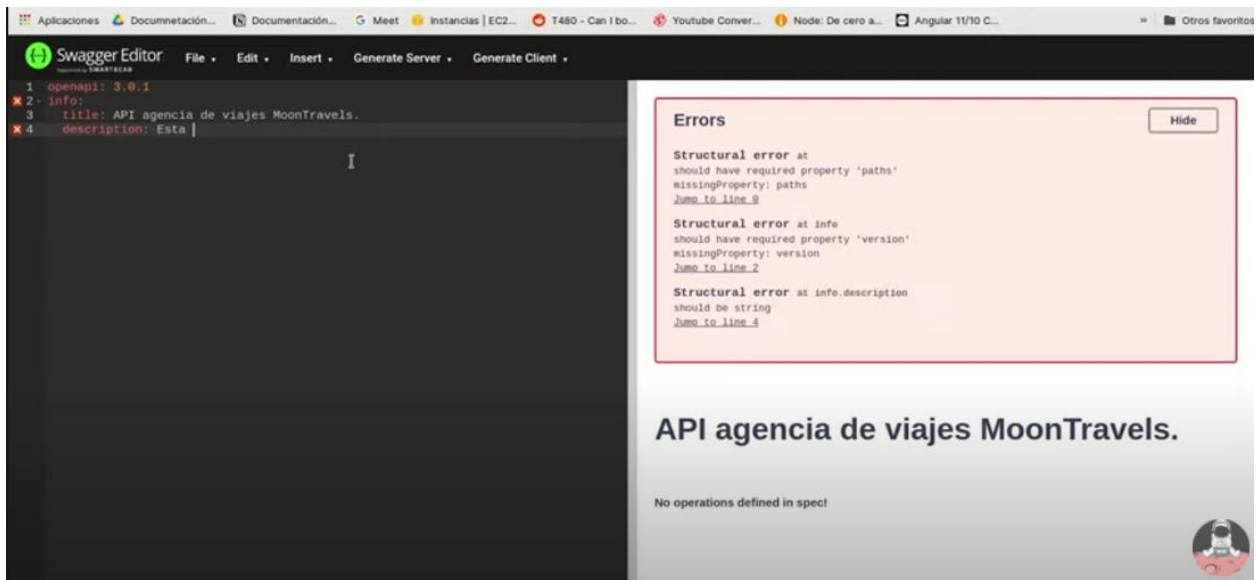
Integer: Solo enteros

Boolean: Verdadero o falso

Array: Listas

Url de la herramienta: <https://swagger.io/>

Inicio de la documentación



1. openapi: 3.0.1 // Estándar de documentación API
2. -info:
3. Schema:
4. Title :// (titulo de la AP)
5. Description : // Describir de que se encarga la API
6. Version:// Dale versión a la API
7. Servers: // URL bases
8. Tag://Agrupaciones de los diferentes servicios de la API(Ejemplo clientes- incluir name para darle nombre y description.
9. Path:

/:

10. Summary: breve comentario de la acción que realiza la url
11. Parameters:
 - Token: Por donde va a viajar el parámetro
 - In: Header
12. Required: true o false si el parámetro es requerido
13. \$ref: // para referenciar parámetro

```
summary: Inserta la infomación de un nuevo
parameters:
- $ref: '#/components/parameters/token'
requestBody:
  content:
    application/json:
```

Ejemplo:



```
1 openapi: 3.0.1
2 info:
3   title: API agencia de viajes MoonTravels.
4   description: Esta API se encarga de manejar la información de la base de
5     datos de la empresa MoonTravels.
6   version: 1.0.0
7 servers:
8   - url: https://api.moontravels/v1/clientes/
9 tags:
10  - name: clientes
11    description: Este endpoint maneja toda la información relacionada con la
12      información de los clientes.
13 paths:
14   /:
15     post:
16       tags:
17       - clientes
18       summary: Inserta la infomación de un nuevo cliente a la BD.
19       parameters:
20       - token:
21         in: header
22         description: Token de autenticación.
23         required: true
```

Referencias propiedades necesarias para el código

```

responses:

parameters:
  token:
    in: header
    description: Token de autenticación.
    required: true

schemas:
  BodyClientesPost:
    type: object
    properties:
      primer_nombre:
        type: string
        description: Primer nombre del cliente
      segundo_nombre:
        type: string
        description: Segundo nombre del cliente
      apellido_materno:
        type: string
        description: Apellido materno del cliente
      apellido_paterno:
        type: string
        description: Apellido paterno del cliente
      correo_electronico:
        type: string
        description: Email del cliente

  ExitoClientesPost:
    type: object
    properties:
      respuesta:
        type: integer
        enum: [1]
        description: Bandera que nos indica si la llamada al servicio fue exitosa

```

Creamos Schemas para respuesta exitosas y no exitosas del lado del servidor

EVIDENCIA DE APUNTES TOMADOS EN EL CURSO DE DOCUMENTACION SWAGGER