

Elaboro	Documento	Versión	Descripción	Fecha
Luis Sánchez Martínez	Diseño de APIs y OpenAPI	1.0	Creación de documento	10/01/2022
Luis Sánchez Martínez	Diseño de APIs y OpenAPI	1.1	Se actualiza documento	12/03/2024

Diseño de APIs y OpenAPI

¿Qué es OpenAPI?

OpenAPI es un estándar para la descripción de las interfaces de programación o Application Programming Interfaces (API). La especificación OpenAPI define un formato de descripción abierto e independiente de los fabricantes para los servicios de API. En particular OpenAPI puede utilizarse para describir, desarrollar, probar y documentar las API's compatibles con REST.

La actual especificación OpenAPI surgió del proyecto predecesor Swagger. La empresa de desarrollo SmartBear sometió la especificación existente de Swagger a una licencia abierta y dejó el mantenimiento y desarrollo posterior en manos de la iniciativa OpenAPI. Además de SmartBear, entre los miembros de la iniciativa OpenAPI se encuentran gigantes de la industria como Google, IBM y Microsoft. La fundación Linux también apoya este proceso.

Resumen OpenAPI

OpenAPI es una especificación, es decir, una descripción abstracta que no está ligada a una aplicación técnica concreta. Hasta la versión 2.0 esta especificación todavía se llamaba Swagger.

Con OpenAPI, una API puede describirse de manera uniforme. Esto se conoce como “definición API” y se genera en un formato legible por máquina. En particular, se utilizan los lenguajes YAML y JSON.

Técnicamente, YAML y JSON difieren solo ligeramente, por lo que es posible convertir automáticamente una definición API existente de un lenguaje a otro. YAML tiene una estructura más clara y es más fácil de leer para personas.

```
# YAML
servers:
  - url: http://development.project.com/v1
    description: Development server
  - url: http://staging.project.com/v1
    description: Staging server
  - url: http://api.project.com/v1
    description: Production server
```

```
// JSON
{
  "servers": [
    {
      "url": "http://development.project.com/v1",
      "description": "Development server"
    }
  ]
}
```

```

    },
    {
      "url": "http://staging.project.com/v1",
      "description": "Staging server"
    },
    {
      "url": "http://api.project.com/v1",
      "description": "Production server"
    }
  ]
}

```

La especificación OpenAPI define una serie de propiedades que pueden utilizarse para desarrollar una API propia. Estas propiedades se agrupan en los llamados objetos. En la actual versión 3.0.3, OpenAPI define la estructura de los siguientes objetos, entre otros:

- **Info Object:** versión, nombre, etc. de la API.
- **Contact Object:** datos de contacto del proveedor de la API.
- **License Object:** licencia bajo la cual la API proporciona sus datos.
- **Server Object:** nombre del host, estructura de la URL, puertos del servidor a través del cual se dirige la API.
- **Components Object:** componentes encapsulados que pueden utilizarse varias veces dentro de una definición de API.
- **Paths Object:** rutas relativas a los puntos finales de la API que se utilizan junto con el servidor del objeto.
- **Path Item Object:** operaciones permitidas para una ruta específica como GET, PUT, POST, DELETE.
- **Operation Object:** específica, entre otras cosas, los parámetros y las respuestas del servidor que se esperan de una operación.

Áreas de aplicación de OpenAPI

OpenAPI se utiliza para describir API REST de manera uniforme, es decir, la definición API, está disponible en un formato legible por máquina, estos incluyen:

- **Creación de documentación API:** La documentación basada en HTML se genera automáticamente a partir de la definición API legible por máquina. Esta sirve como material de consulta para los desarrolladores que acceden a los servicios API. Si la definición API cambia, la documentación se vuelve a generar para que ambas concuerden.
- **Creación de conexiones en diferentes lenguajes de programación:** Con las herramientas apropiadas se puede crear una biblioteca de software adecuada del cliente a partir de la definición API en un lenguaje de programación compatible. Esto permite a los programadores de todo tipo acceder a la API. La biblioteca de software se incorpora de manera convencional. Por lo tanto, el acceso a los servicios de API tiene lugar, por ejemplo, mediante el acceso a funciones dentro del mismo entorno de programación.
- **Elaboración de casos de pruebas:** Cada componente de un software debe someterse a diversas pruebas para asegurar su funcionalidad. En concreto, es preciso volver a probar un componente de

software cada vez que se cambie el código subyacente. A partir de la definición API, se pueden generar estos casos de prueba automáticamente para poder comprobar la funcionalidad de los componentes del software en todo momento.

Cabe señalar que no todas las API's pueden representarse utilizando OpenAPI. Sin embargo, las API REST son compatibles.

Ventajas de OpenAPI

La ventaja de OpenAPI es la que puesta en marcha, documentación y prueba de una API son coherentes y constantes durante el desarrollo y el mantenimiento. Además, el uso de la especificación OpenAPI permite una mejor coordinación del desarrollo de la API entre los equipos de backend y frontend. En ambos equipos, los componentes del código pueden generarse a partir de la definición API para que tanto en backend como en frontend puedan desarrollarlos y probarlos sin tener que esperar al otro.

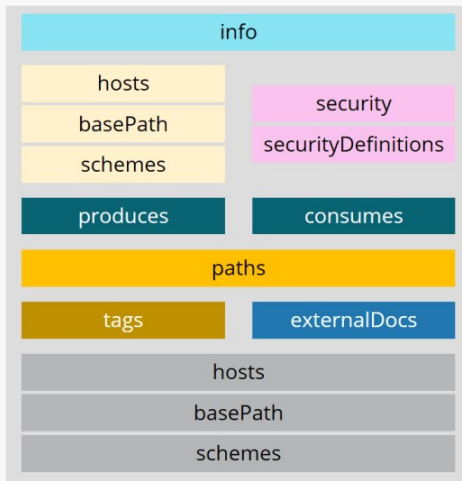
Además de las ventajas generales, se utiliza como estándar de base para el desarrollo de API REST. Esto es atractivo, porque desarrollar una API compatible con REST de forma manual no es una trivialidad. Sin embargo, las API REST ofrecen algunas ventajas. Por ejemplo, REST se ejecuta sobre HTTP o HTTPS y los puertos para ello están abiertos en cualquier cortafuegos.

Además, el uso de OpenAPI ofrece las siguientes ventajas:

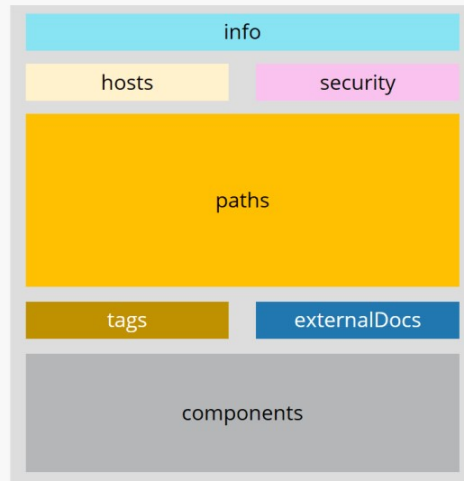
- Definir las API HTTP independientemente de un lenguaje de programación específico.
- Generar código de servidor para una API definida en OpenAPI.
- Generar bibliotecas de lado del cliente para una API compatible con OpenAPI en más de 40 lenguajes de programación.
- Programar una definición OpenAPI con las herramientas apropiadas.
- Crear documentación interactiva de API.
- Permitir que las personas y las máquinas descubran y entiendan las capacidades de un servicio sin tener que mirar el código fuente o la documentación adicional.
- Acceder a los servicios de API con un gasto mínimo de puesta en marcha.

OpenAPI structure

OpenAPI v2.0



OpenAPI v3.0



Referencias

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-openapi/>