Universidad Tecnologica de Xicotepec de Juárez

Ingeniería en Desarrollo y Gestión de Software

Materia: Extracción de Conocimiento en Bases de datos

Como implementar swagger en python

Integrantes:

Elisama Arturo Calva Moreno M-200561

Eliel Díaz Galindo M-200327

Mariela González López M-181193

Fabricio Guzmán Vite M-180610

Docente: MTI. Marco Antonio Ramírez 9° “A”

Mayo-Agosto 2023

¿Qué es Swagger?

Swagger es un framework que resulta muy útil para documentar, visualizar y consumir servicios REST. El objetivo de Swagger es que la documentación del API RESTFul se vaya actualizando cada vez que se realicen cambios en el servidor.

Cuando hablamos de Swagger nos referimos a una serie de reglas, especificaciones y herramientas que nos ayudan a documentar nuestras APIs. De esta manera, podemos realizar documentación que sea realmente útil para las personas que la necesitan. Swagger nos ayuda a crear documentación que todo el mundo entienda.

El principal de ellos es que todo el mundo entiende Swagger, o casi todos, seas desarrollador o tengas pocos conocimientos sobre desarrollo podrás entenderla gracias al Swagger UI. Incluso las máquinas pueden leerlo, convirtiéndose en otra ventaja muy atractiva. Con Swagger la documentación puede utilizarse directamente para automatizar procesos dependientes de APIs. Otra de las ventajas, es que lo podemos encontrar integrado en herramientas populares y potententes como las de WSO2. Por último, tampoco nos podemos olvidar de las herramientas que ofrece.

¿Qué es Swagger UI?

Swagger UI es una de las herramientas atractivas de la plataforma. Para que una documentación sea útil necesitaremos que sea navegable y que esté perfectamente organizada para un fácil acceso. Por esta razón, realizar una buena documentación puede ser realmente tedioso y consumir mucho tiempo a los desarrolladores.

¿Para qué se usa el Swagger UI y por qué?

Utilizando el Swagger UI para exponer la documentación de nuestra API, podemos ahorrarnos mucho tiempo. Con el Swagger UI podremos organizar nuestros métodos e incluso poner los ejemplos que necesitemos.

Swagger UI utiliza un documento JSON o YAML existente y lo hace interactivo. Crea una plataforma que ordena cada uno de nuestros métodos (get, put, post, delete) y categoriza nuestras operaciones. Cada uno de los métodos es expandible, y en ellos podemos encontrar un listado completo de los parámetros con sus respectivos ejemplos. Incluso podemos probar valores de llamada.

Conclusiones

Sin duda, Swagger, Swagger UI y todas sus herramientas, son capaces de hacer el trabajo de nuestros desarrolladores mucho más fácil a la hora de documentar nuestras APIs. Unas APIs bien documentadas significa que son accesible por otros desarrolladores, y mejorando la accesibilidad de nuestras APIs podremos mejorarlas. Swagger, es un framework tan establecido que hasta está integrado en herramientas tan populares para la gestión de APIs como WS02 API Manager. Gracias a su popularidad y sus resultados, Swagger hace posible que cada API tenga el mejor diccionario para entenderla.

¿Qué es Python?

Python es un lenguaje de programación de código abierto, creado por Guido van Rossum en 1991. Se trata de un lenguaje orientado a objetos, fácil de interpretar y con una sintaxis que permite leerlo de manera semejante a como se lee el inglés. Es un lenguaje interpretado, esto significa que el código de programación se convierte en bytecode y luego se ejecuta por el intérprete, que, en este caso, es la máquina virtual de Python.

Para qué sirve Python

La respuesta es breve: para todo. Python está en todo, desde programación de instrumentos hasta software de computadoras, desarrollo web y aplicaciones móviles. Incluso, te permite hacer comentarios para que tengas recordatorios para funciones futuras o indicar problemas en una línea de código.

Python es genial para casi cualquier necesidad de desarrollo, ya sea programación de servidores, operación de sistemas, software, juegos y mucho más. A continuación, repasamos los usos más comunes.

Cómo crear una API en Python

Como científico o ingenieros de datos, muchas veces tenemos que compartir nuestro trabajo para que cualquier otra persona de la empresa pueda utilizar los procesos o modelos que hemos creado. Claramente, compartir un script no es una opción, ya que todo el mundo necesitaría tener esos mismos programas que tú. Aquí es cuando entran en juego las APIs y hoy, vamos a ver qué son y cómo crear una API en Python. ¿Suena interesante? ¡Pues vamos a ello!

Conceptos básicos sobre APIs

Una API (Application Programming Interface) permite que dos sistemas informáticos interactuen entre sí. Por ejemplo, si creamos una automatización que genere un informe y lo mande por email, el envío de ese email no se hace de forma manual, lo

hará el propio script. Para ello, Python (o el lenguaje que usemos), deberá pedirle a Gmail que mande ese correo, con ese informe adjunto a ciertas personas. La forma de hacerlo es mediante una API, en este caso la API de Gmail

Bien, ahora que sabes lo que es una API, veamos cuáles son las partes principales de una API:

Protocolo de transferencia HTTP: es la forma principal de comunicar información en la web. Existen diferentes métodos, cada uno de ellos utilizados para diferentes cuestiones:

GET: este método permite obtener información de la base de datos o de un proceso.

POST: permite mandar información, ya sea para añadir información a una base de datos o para pasar el input de un modelo de machine learning, por ejemplo.

PUT: permite actualizar información. Generalmente se usa para gestionar información en base de datos.

DELETE: este método se utiliza para eliminar información de la base de datos.

Url: es la dirección en la que podremos encontrar a nuestra API. Básicamente esta URL constará de tres partes:

Protocolo: como toda dirección, puede ser http:// o https://

Dominio: el host en el que está alojado, que va desde el protocolo hasta el final del .es o .com, o la terminación que sea. En mi web, por ejemplo, el dominio es anderfernandez.com.

Endpoint: al igual que una web tiene varias páginas (/blog), (/legal), una misma API puede incluir varios puntos y que cada uno haga cosas diferentes. Al crear nuestra API en Python nosotros indicaremos los endpoints, por lo que debemos asegurarnos de que cada enpoint sea representativo de lo que haga la API que está por detrás.

Bien, ahora que ya sabemos qué es una API y cuáles son sus partes principales, veamos cómo podemos crear una API en Python.

Cómo crear una API en Python

Existen diferentes formas de crear una API en Python, siendo las más utilizadas FastAPI y Flask. Así pues, explicaré cómo funcionan las dos, de tal forma que puedas usar la forma de crear APIs en Python que más te guste. Empecemos con FastAPI.

Crear una API REST a partir de un documento Swagger importado

Para crear una API REST en IBM Integration Bus, puede importar un documento Swagger utilizando IBM Integration Toolkit.

Antes de empezar

Cree un documento Swagger que describa los recursos y las operaciones que desea en la API REST. Asegúrese de que el documento Swagger se adhiere a las directrices de la especificación Swagger y que cumple los requisitos de IBM Integration Bus. Consulte Swagger RESTful API Documentation Specification Versión 2.0 y Restricciones a documentos Swagger para obtener información.

Acerca de esta tarea

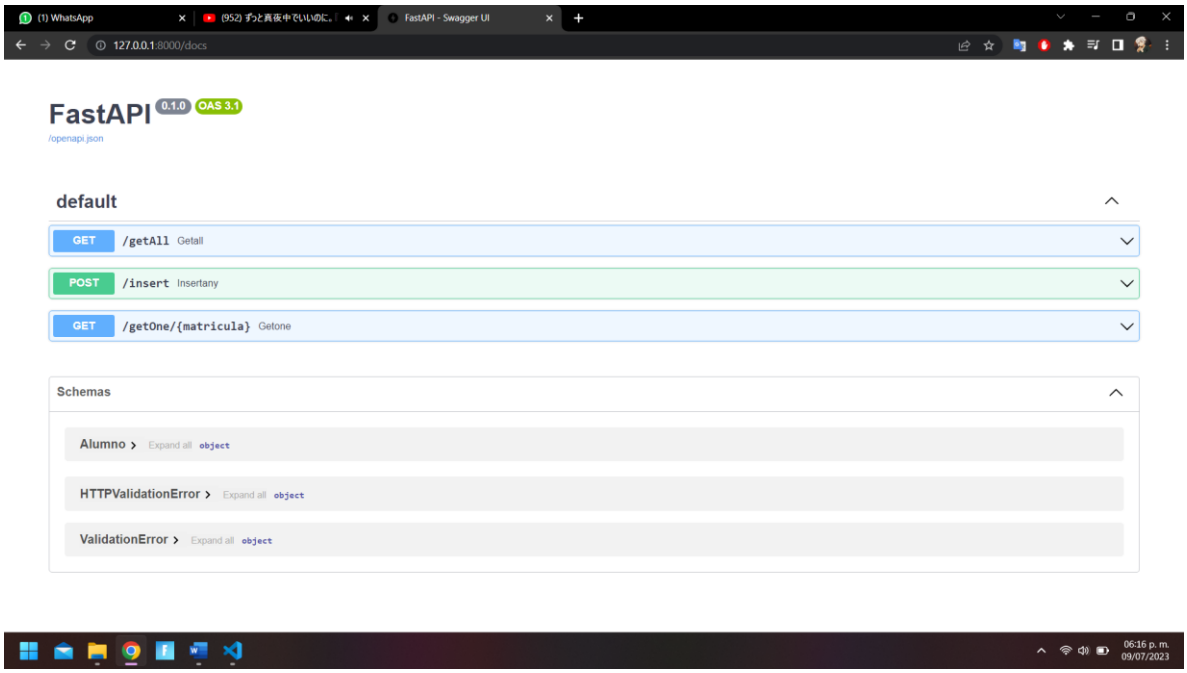
Para crear una API REST en IBM Integration Bus, puede utilizar el asistente de Crear una API REST en IBM Integration Toolkit para importar un documento de Swagger. A continuación, debe crear la API REST y otros artefactos de proyecto necesarios para implementar y desplegar la nueva API REST. De forma alternativa, puede crear una API REST definiendo los modelos, los recursos y las operaciones desde cero, tal como se describe en Creación de una API REST desde cero utilizando IBM Integration Toolkit.

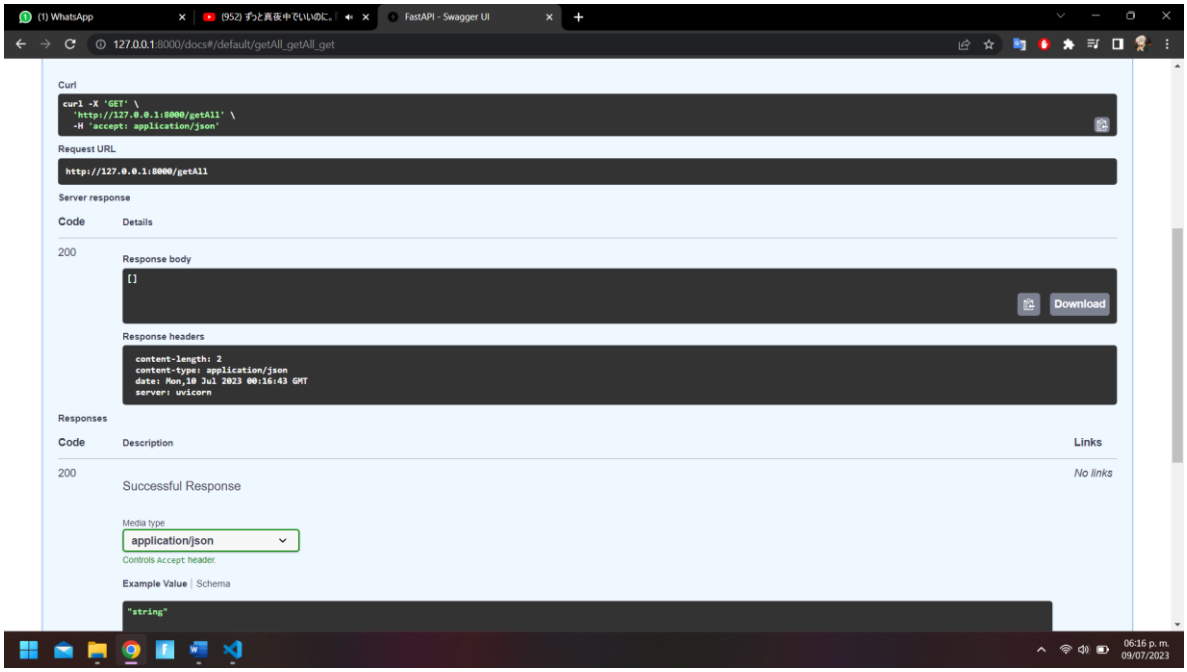
Procedimiento

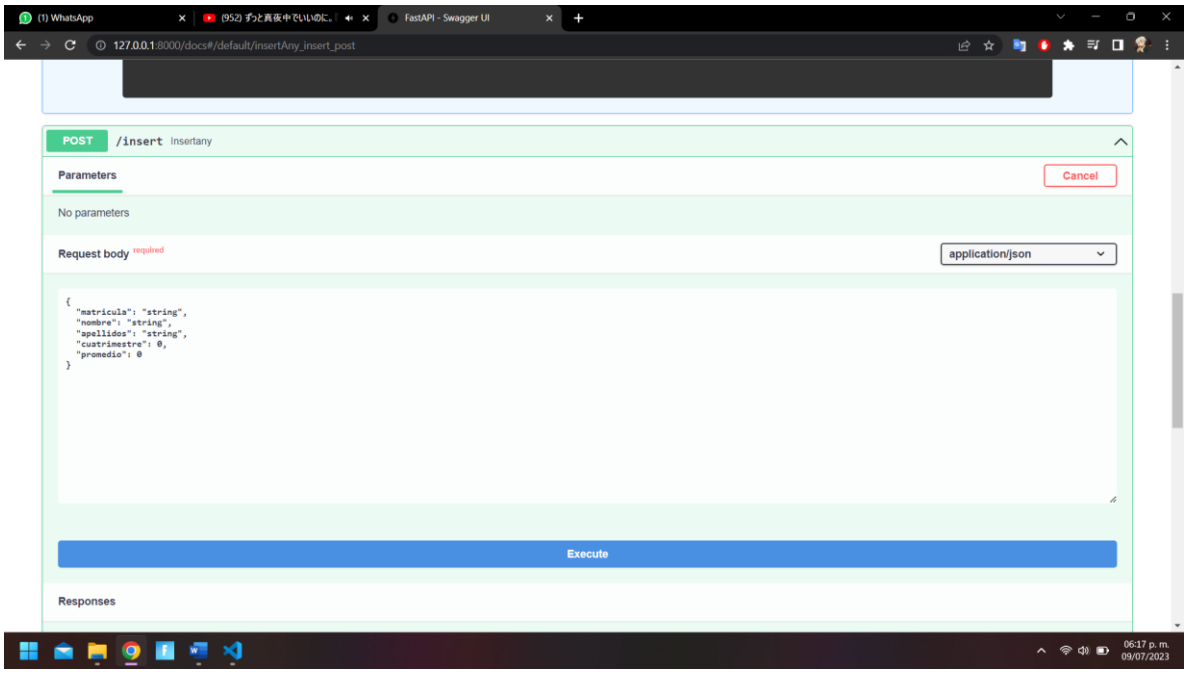
Para crear una API REST importando un documento Swagger , realice los pasos siguientes:

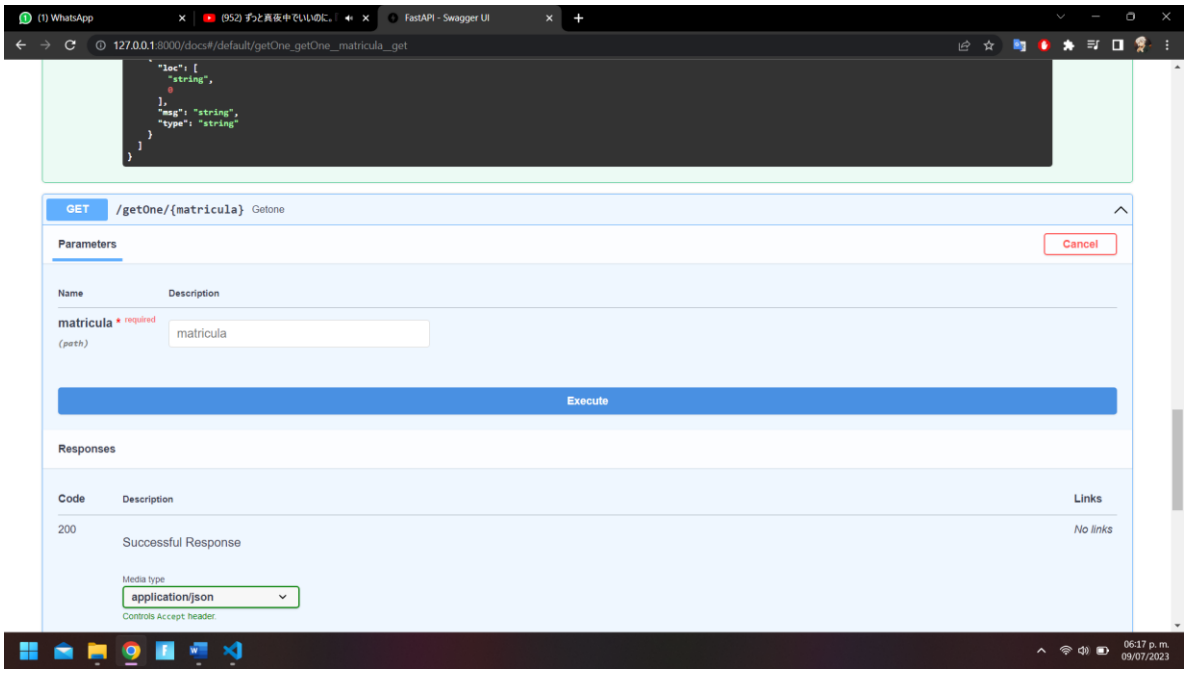
1. Abra el asistente de Crear una API REST pulsando Archivo > Nuevo > API REST.
2. Especifique un nombre para la API REST.
   El nombre que especifique se utiliza como el nombre del proyecto en IBM Integration Toolkit.
3. Seleccione Importar recursos y operaciones definidos en un documento Swagger y, a continuación, pulse Siguiente.
4. Seleccione la vía de acceso al documento Swagger que describe los recursos y las operaciones que desea en la API REST.
   Puede importar el documento Swagger desde el sistema de archivos o desde un proyecto existente en el espacio de trabajo.
   El archivo se válida para su uso en una API REST. Si se encuentran errores mientras se valida el documento Swagger seleccionado, estos errores de validación se muestran en la parte superior del asistente. No puede continuar si se encuentran errores de validación en el documento Swagger seleccionado.
5. Opcional: Para revisar la lista de recursos y operaciones contenidos en el documento Swagger seleccionado, pulse Siguiente.
6. Para finalizar la creación de la API REST, pulse Acabado.

El Descripción de la API REST para la nueva API REST se abre automáticamente, que puede utilizar para implementar operaciones como se describe en Implementación de una operación en una API REST.






FastAPI - Swagger UI
127.0.0.1:8000/docs#/default/insertAny_insert_post
POST /insert Insertany
Parameters
Cancel
No parameters
Request body required
application/json
{
"matricula": "string",
"nombre": "string",
"apellidos": "string",
"cuatrimestre": 0,
"promedio": 0
}
Execute
Responses


FastAPI - Swagger UI
127.0.0.1:8000/docs#/default/getOne_getOne__matricula__get
"loc": [
"string",
0
],
"msg": "string",
"type": "string"
}
]
}
GET /getOne/{matricula} Getone
Parameters
Cancel
Name
Description
matricula * required
(path)
matricula
Execute
Responses
Code
Description
Links
200
Successful Response
No links
Media type
application/json
Controls Accept header.