FastAPI

El framework mas veloz para el desarrollo web con Python. Enfocado para realizar APIs, es el mas rápido en lo que respecta a la velocidad del servidor superando a Node.Js y a GO. Fue creado por Sebastian Ramirez, es de código abierto y se encuentra en Github, es usado por empresas como Uber, Windows, Netflix y Office.

Librerias y frameworks

- Uvicorn: es una librería de Python que funciona de servidor, es decir, permite que cualquier computadora se convierta en un servidor
- Starlette: es un framework de desarrollo web de bajo nivel, para desarrollar aplicaciones con este requieres un amplio conocimiento de Python, entonces FastAPI se encarga de añadirle funcionalidades por encima para que se pueda usar mas fácilmente
- Pydantic: Es un framework que permite trabajar con datos similar a pandas, pero este te permite usar modelos los cuales aprovechara FastAPI para crear la API

Instalación

```
pip install fastapi uvicorn
```

OpenAPI

FastAPI también está parado sobre los hombros de OpenAPI, el cual es un conjunto de reglas que permite definir cómo describir, crear y visualizar APIs. Es un conjunto de reglas que permiten decir que una API está bien definida. OpenAPI necesita de un software, el cual es Swagger, que es un conjunto de softwares que permiten trabajar con APIs. FastAPI funciona sobre un programa de Swagger el cual es Swagger UI, que permite mostrar la API documentada. Acceder a la documentación interactiva con Swagger UI: {localhost}/docs Acceder a la documentación interactiva con Redoc: {localhost}/redoc

Path operations

Path: es una routa (route o endpoint) la cual nosotros ingresamos seguido el dominio de nuestro aplicativo
Operation: Es un metodo http por el cual nos comunicamos, existen los siguientes:

- Get: Obtener
- Post: Crear
- Put: Modificar/Actualizar
- Delete: Eliminar

- Options: Devuelve un header adicional llamado allow que contiene los metodos http que pueden utilizarse en ese endpoint.
- Head: Devuelve info sobre el documento, mas no el documento en si.

- Patch: Hacer modificaciones parciales al documento a diferencia de put que permite cambiar el documento entero.
- Trace: Nos permite observar que esta pasando en la peticion y nos devuelve nuestro input con propositos de debugging.

> Path Operation Decorator: permite el ingreso de una path a una operation designada para la ejecucion de un codigo siguiente Path Operation Function: realiza la ejecucion de un codigo siguiendo la especificación del path operation decorator llamado anteriormente

Path parameters

Los parámetros de ruta son partes variables de una ruta URL . Por lo general, se utilizan para señalar un recurso específico dentro de una colección, como un usuario identificado por ID. Una URL puede tener varios parámetros de ruta.

Query parameters

Son un conjunto definido de parámetros adjuntos al final de una URL . Son extensiones de la URL que se utilizan para ayudar a definir contenido o acciones específicos en función de los datos que se transmiten.

Request Body y Response Body

Debes saber que bajo el protocolo HTTP existe una comunicación entre el usuario y el servidor. Esta comunicación está compuesta por cabeceras (headers) y un cuerpo (body). Por lo mismo, se tienen dos direcciones en la comunicación entre el cliente y el servidor y definen de la siguiente manera:

Request : Cuando el cliente solicita/pide datos al servidor. Response : Cuando el servidor responde al cliente.

Request Body

Con lo anterior mencionado, Request Body viene a ser el cuerpo (body) de una solicitud del cliente al servidor.

Response Body

Con lo anterior mencionado, Response Body viene a ser el cuerpo (body) de una respuesta del servidor al cliente.

Validaciones

Para especificar las validaciones, debemos pasarle como parámetros a la función Query lo que necesitemos validar.

Para tipos de datos str

max_length : Para especificar el tamaño máximo de la cadena. min_length : Para especificar el tamaño minimo de la cadena. regex : Para especificar expresiones regulares.

Para tipos de datos int

ge : (greater or equal than ≥) Para especificar que el valor debe ser mayor o igual. le : (less or equal than ≤) Para especificar que el valor debe ser menor o igual. gt : (greater than >) Para especificar que el valor debe ser mayor. lt : (less than <) Para especificar que el valor debe ser menor.

- Ejemplo:

```
# Validaciones de un nombre de usuario.
Query(None, min_length=1, max_length=50)):
```

Es posible dotar de mayor contexto a nuestra documentación. Se deben usar los parámetros title y description.

- title : Para definir un título al parámetro.
- description : Para especificar una descripción al parámetro.
- Ejemplo:

```
# Validaciones para un Identificador.
Query(
    None,
    title="ID del usuario",
    description="El ID se consigue entrando a las configuraciones del
perfil")
```

Diferencia Path, Query Parameters and Request Body

1. usamos Path Parameters cuando por ejemplo se trata de un id y esas cosas, como una variable etc.
2. Usamos los Query Parameters para solicitar infomación opcional del servidor.
3. Usamos los Requests Body para enviar información que tiene formato de un modelo.

Validacions Models

Para validar modelos tomamos uso de la clase de Pydantic Field, que funciona igual a las validaciones que ya hemos hecho con Path, Query y Body.

Pydantic

Todos estos tipos de datos corresponden a Pydantic, se pueden importar al igual que Field.

Tipos de datos clásicos

- str → Cadena de texto
- int → Número entero
- float → Número flotante (decimal)
- bool → Booleano

Tipos de datos exóticos

- Enum → Enumerar caracteres
- HttpUrl → Revisa si una URL es valida (***https://myapp.com***, ***<www.google.com>***) , response_model=PersonOutligatorio
- **Query parameter ->** opcional
- **Request body ->** El body de una Petición HTTP
- **Response body ->** El body de una Respuesta

Status codes

Los status code o codigos de estado son respuestas http los cuales indican el el estado de finalizacion de una solicitud especifica:

- Respuestas informativas (100-199)
- Respuestas Satisfactorias (200-299)
- Redirecciones (300-399)
- Errores de los clientes (400-499)
- Errores de los servidores (500-599)

> Más información Errores explicados con gatos

Formularios

```
# instalar libreria
pip install python-multipart
```

Cookies

Una pieza de código que un servidor mete en tu computadora cuando estas navegando en la web

Headers

Una parte de una petición o respuesta HTTP que contiene datos sobre la petición o la respuesta, como el formato, quien la hizo, el contenido, etc…

Archivos

Entrada de datos que se refiere a los archivos FastAPI, por ejemplo una imagen o un video, se utilizan dos clases File y UploadFile

UploadFile

Esta clase tiene una serie de parametros, se refiere a la clase donde se guardará el archivo

- filename: se refiere al nombre del archivo, con esto tenemos el control sobre el nombre del archivo que suba el cliente a la aplicación
- content_type: formato del archivo por ejemplo JPEG, MP4, GIF…
- file: se refiere al archivo en si mismo, los bytes del mismo

File

Hereda de Form y funciona similar a las clases Query, Path y Body, se encarga de guardar los bytes del archivo.
Ventajas de usar UploadFile en lugar de solo File o Bytes:

- El archivo se guardará en la memoria hasta que supere un tamaño máximo, al pasar ese límite se guardara en el disco, esto quiere decir que funciona mucho mejor con archivos grandes sin consumir toda la memoria RAM
- Puedes obtener metadata del archivo
- funciona como un file-like async interface.
- Usa metodo Asincronos como write, read, seek y close

> Más información sobre Request Files - Fas API

Deprecar

Deprecar una pieza de código sucede cuando:

1. Se encuentra un mejor método mas eficiente para resolver un problema que nosotros ya tenemos. Lo que hacemos no es eliminar dicho método si no la dejamos sin efecto. Para aprovechar el código posteriormente si lo requerimos nuevamente.
2. Una funcionalidad diferente de nuestro código a la que ya tenemos definidos.
3. Cuando se esta realizando una refactorización profunda del código, debido a que no tiene las mejores practicas, se define deprecar las path operation que se tienen por otras nuevas y se reemplazan. Nota: Siempre es mejor mantener el código que modificarlo desde cero.

Twitter API