Todo lo que pasa en internet, son llamadas a API’s

Cuando consultas a un sitio web, no consultas a la base de datos directamente, consultas a la API, el cual es una especie de intermediario que hace validaciones y decide que devolver de la base de datos.

Endpoints: Cada una de las llamadas que podemos ir haciendo. La API esta hospedada en el host (anfitrión). La url es un disfraz para la IP. La IP es la dirección Es un conjunto de endpoints. Cada endpoint es un chino que te trae un dato. Cada chino es una función.

Los endpoints hablan HTTP.

El protocolo es HTTP

HTTPS = Tiene seguridad adicional. Trabajaremos con HTTP original.

Al final es una función en Python.

Ejemplo de IP local: 127.0.0.1:5000

IP EJEMPLO

<http://host/endpoint>

EJEMPLO

<http://host/v1/Public/Characters?name=hulk>

Siempre todo se basa en una relación cliente – servidor.

La llamada es hacer una llamada de tipo get. Le pido al servidor que me traiga algo.

El navegador lo que hace es tomar la petición y reproducir el código html

Postman: Sirve para hacer pruebas al desarrollar una API. Ponemos la llamada y trae el html. Al dar al botón de code, da el código de Python que permite traer el html a Python.

Flask

Librería de Python que sirve para hacer API’s.

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Diferencia entre unauthorized y forbidden es que en el primero te reconoce pero no te permite, en el segundo no.

Patrón de fondo

Descripción generada automáticamente con confianza media

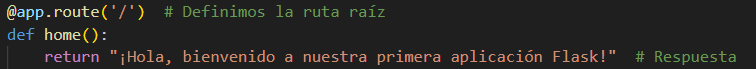
Put es modificar

Al buscar en Google, cuando vemos todo lo que agrega, añade códigos para personalizar nuestra consulta.

Lo que va después del arroba es un decorador. Lo de abajo es una función. El decorador toma lo que la funcion devuelve y hace cosas con el. Recordar ejemplo de oddity como decorador que aplicaba la función al resultado de la suma.

Nota: Postman muestra el código mientras que el navegador ejecuta el código.

A continuación vemos un endpoint:



Recomendaciones:

Error de mapping: 2 funciones se llaman igual

Poner nombre de las funciones de forma parecida o igual que la ruta.

Los errores suelen pasar cuando buscamos algo que no existe.

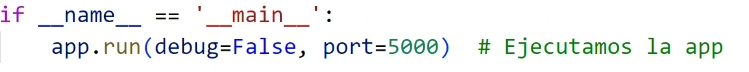
Nunca permitir que pase este error:

Imagen que contiene Texto

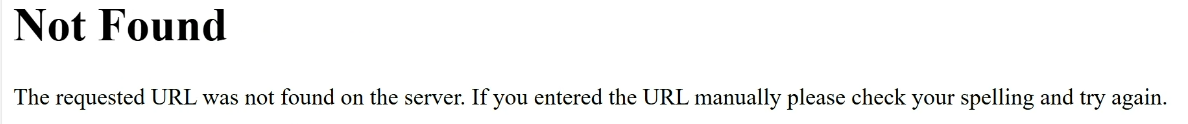
Descripción generada automáticamente

El modo debug permite que se rompa y arreglarlo. Al desactivarlo tenemos que reinciarla.

Cuando estemos en modo producción:



Para que salga esto:



HAY 2 OPCIONES PARA TRABAJAR CON API’s

Cambiando el nombre en la ruta

Texto

Descripción generada automáticamente

La otra es la del interrogante. Donde usamos request.args

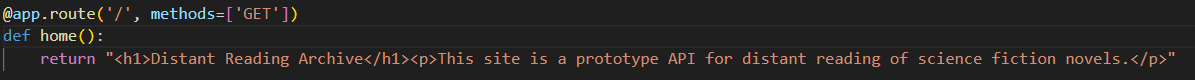
Gráfico

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Con esto especificamos que la llamada sea del tipo get:



QA (Quality Assurance): Auditores de códigos que andan por ahí revisando buenas practicas en el código.

Tarea

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Crear instancia

PostgreSQL 16 -> Elegir siempre la penúltima.

ID -> Para practicar ponemos postgres

Contraseña -> Ponemos también postgres

SIEMPRE postgres. Si es dato no estructurado: MongoDB

2 cpu’s

Núcleo compartido mas pequeño.

Agarrar siempre Bélgica

IMPORTANTE: IP PUBLICA

Agregar red: Restringe quien puede entrar.

0.0.0.0/0

Puerto comun: 5432

DESHABILITAR LA PROTECCION CONTRA LA ELIMINACION

Mantenimiento: No suele importar si no estamos mucho tiempo.

PARTE 2 (TAREA TIPS)

Se usa Libreria OI para guardar imágenes en la ram. En un html como leo un archivo de tipo entero base64. Con esto podemos guardar archivos sin necesidad de ponerlos en el disco duro.

Hay 2 ocpiones:

* Guardar la imagen creada en la ram y traerla luego.
* Gurardarla de forma temporal con oi

DESPLEGAR

Siempre crear un archivo de requirements. En este es recomendable poner lo utilizado (Revisando manualmente el proyecto) y poner la versión: pip freeze.

Texto

Descripción generada automáticamente

¿Cómo desplegar una APP?

Buscame un proveedor cloud que permita desplegar de forma gratutita.

Por el momento salen bien Render y Vercel.

Siempre seleccionar PostgreSQL.

TIPS

Siempre llamar al archivo y al fichero app.py.

Solucionar problema de la controseña de la url: Guardar la contraseña como una variable de entorno.

AL momento de desplegar:

Region: No importa

Root Directory: Aquí puedes decir si todo lo tienes en una carpeta api.

Build Commando: Se pone de forma predeterminada, muy importante.

Start Commando: gunicorn app:app

Poner variable de entorno; Al menos poner la contraseña de postgress.

En este caso es necesario siempre poner gunicorn en requirements ya que es necesario para que render funcione.

SECRETOS Y VARIABLES DE ENTORNO

Las variables de entonro pueden encontrarse así:

Para gestionarlas hay una librería:

pip install python-dotenv

luego ponemos

from dotenv import load\_dotenv

print(os.environ) => Manejra con cuidado, sirve para ver las variables de entorno. Podemos ver que es un diccionario.

Creamos un archive llamado .env con las contraseñas dentro:

Load\_dotenv va a ir al archivo de env, va a scar las variables como variables de entorno.

churro=tuprima

luego lo cargamos así

churro = os.eviron[“churro”]

el archivo .env debe meterse en un gitignore.

En un gitignore (Archivo de texto) se ponen los archivos que no queremos incluir:

.env

Train\_model.ipynb

NOTA: En los requirements hay que incluir python-dotenv

IMPLEMENTACIÓN DE AI (APIS de AI)

OPENAI

COHERE

GEMINI (GEMMA)

PERPLEXITY

MISTRAL: Startup Francesa.

Otros que no tienen API: QWEN (China), LLAMA (Facebook).

Otros:

Replicate: Tiene acceso a los otros proveedores. Uno paga el hierro. Pagas el hierro y tienes acceso al openssource. Usabamos flux .1 para generar imágenes.

GROQ: Es parecido a Replicate.

PASOS

Nos vamos a Google AI Studio. La versión 2.0 Flash Experimental es gratis actualmente.

<https://aistudio.google.com/prompts/new_chat?gad_source=1&gclid=Cj0KCQiAs5i8BhDmARIsAGE4xHy7tuef8iCtNjEslVjmrMhXcOk7YaNOl4D4hkl5K9o3hK4C8Fmy_G4aAjUkEALw_wcB>

Podemos poner max tokens, pero ojo que puede cortar abruptamente al modelo. Hay que decirle siempre al modelo que sea conciso y darle un número de palabras.

Prompt Engineering:

* Pedir que omita saludos, metadatos, que no de nada parte del texto solicitado.
* Omite todo tipo de formato enriquecido (markdown, html), dame solo texto plano.
* Pedirle que sea creativo, que se moje.

TIPS

* Conviene crear un archivo con las funciones que se llame utils.

SIEMPRE PREGUNTARME: DONDE ENCAJA LA AI AQUÍ?

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se importa así:

