



APIRESTfull

Flask

Introducción

- La API RESTful, creada por Roy Fielding, es una interfaz que permite el intercambio seguro de información entre dos sistemas informáticos a través de Internet.
- Funciona como una puerta de enlace entre los clientes y los recursos web.
- Es ampliamente utilizada en aplicaciones empresariales para comunicarse con otras aplicaciones internas o de terceros, ya que sigue estándares de comunicación de software seguros, confiables y eficientes.



- Flask es un framework de Python utilizado para el desarrollo web, incluyendo la creación de APIs.
- Es altamente valorado por su simplicidad y las facilidades que ofrece.
- Existen diversas formas de implementar una API REST en Flask, ya sea utilizando las funcionalidades base del framework o mediante extensiones con diferentes configuraciones.

Framework similares

- Bottle: microframework que es incluso más simple que Flask. Es una buena opción para proyectos pequeños o para desarrolladores que quieren tener mucho control sobre su código.
- Sinatra: microframework para Ruby que es similar a Flask. Es una buena opción para desarrolladores que están familiarizados con Ruby y quieren crear aplicaciones web rápidamente y fácilmente.
- Falcon: es un framework web de alto rendimiento para Python que está diseñado para la construcción de API RESTful. Es una buena opción para desarrolladores que necesitan construir aplicaciones web escalables y confiables.
- FastAPI: es un framework web de alto rendimiento para Python que se basa en la biblioteca de núcleo Python asíncrono (ASGI). Es una buena opción para desarrolladores que necesitan construir aplicaciones web que puedan manejar mucho tráfico.
- Sanic: es un framework web de alto rendimiento para Python que está escrito en C. Es una buena opción para desarrolladores que necesitan construir aplicaciones web que puedan manejar mucho tráfico y necesitan ser lo más rápidas posible.

Recursos

- Un recurso en el contexto de una API REST es un objeto con un tipo específico, datos asociados, relaciones con otros recursos y un conjunto de métodos estándar (HTTP GET, POST, PUT y DELETE) que operan en él.
- Es similar a una instancia de objeto en un lenguaje de programación orientado a objetos.
- Los recursos pueden agruparse en colecciones, que son homogéneas y desordenadas.
- En resumen, una API REST es simplemente una interfaz para interactuar con estos recursos.

Extensiones para API REST en Flask

- ✓ Flask Restful: Es una extensión que permite generar APIs REST fácilmente.
- ✓ Flask SQLAlchemy: Interactúa con la base de datos a través de su ORM (Mapeadores de Objetos Relacionales).
- ✓ Flask Migrate: Permite generar las tablas de la base de datos a partir de ficheros de migración.
- ✓ Flask Marshmallow: Facilita la serialización de los modelos de la base de datos a JSON y viceversa.
- ✓ Marshmallow SQLAlchemy: Integra Flask Marshmallow con SQLAlchemy.



- Visitemos la Web: <https://flask.palletsprojects.com/>
- Forma parte del proyecto: <https://palletsprojects.com/>
- ¿Qué versión es la actual en el Proyecto ?

Flask Instalación

- **Versión de Python**

- Dependencias

- Entornos virtuales

- macOS/Linux

 - \$ mkdir myproject

 - \$ cd myproject

 - \$ python3 -m venv venv

- Windows

 - > mkdir myproject

 - > cd myproject

 - > py -3 -m venv venv

Flask Instalación

- Activación del entorno virtual
- macOS/Linux
 - \$. venv/bin/activate
- Windows
 - venv\Scripts\activate
- Install Flask
 - \$ pip install Flask

venv

- Si usa Python 3.3 o posterior, el módulo venv es la forma preferida de crear y administrar entornos virtuales.
- venv está incluido en la biblioteca estándar de Python y no requiere instalación adicional.
- `sudo apt install python3.8-venv`

Instalación de Python 3.x

- ✓ **Revisar versión disponible Python**
- ✓ <https://phoenixnap.com/kb/how-to-install-python-3-ubuntu>
- ✓ `sudo apt update`
- ✓ `sudo apt install software-properties-common`
- ✓ `sudo add-apt-repository ppa:deadsnakes/ppa`
- ✓ `sudo apt update`
- ✓ `sudo apt install python3.10`

Flask Instalación

- ✓ source venv/bin/activate
- ✓ # to deactivate run below command
- ✓ deactivate
- ✓ <https://www.folkstalk.com/tech/ubuntu-activate-venv-with-code-examples/>
- ✓ pip3 install Flask

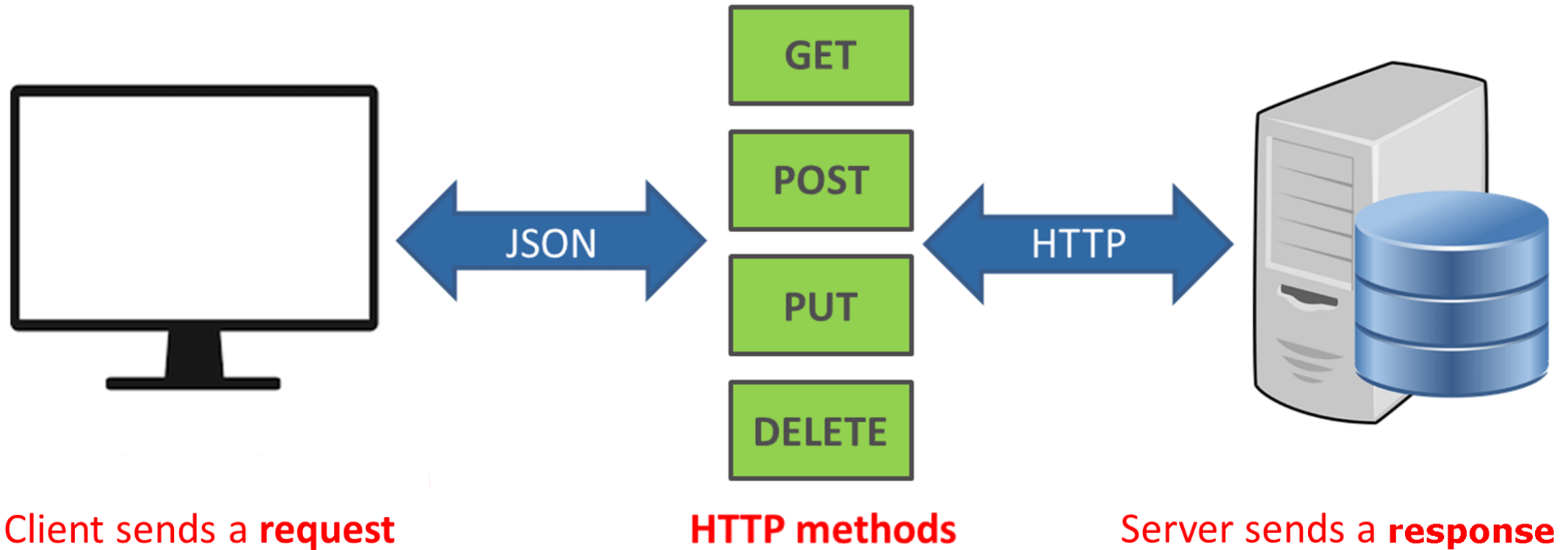
Apache, netstat y otros

- ✓ Para iniciar el servicio de servidor web Apache, usamos el comando:
- ✓ `sudo systemctl start apache2`
- ✓ `sudo systemctl stop apache2`
- ✓ `sudo apt install apache2`
- ✓ `netstat -atn`

Arrancar App en Flask

- ✓ flask --app **App** run --host 0.0.0.0 --port = 8080
- ✓ python **App.py**
- ✓ Ambas son similares

API RESTFULL



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

Invocación a la API Web

- Puede agregar secciones de variables a una URL utilizando `<variable_name>`. La función recibe la variable como argumento de palabra clave.
- Devolver salida serializable JSON
- El valor de retorno de una función en una aplicación Flask debe ser JSON serializable. Puede usar `jsonify` para hacer que su salida JSON sea serializable. Esta función envuelve `json.dumps()` para convertir la salida JSON en un objeto de respuesta con tipo mime `application/json`.

Ejemplos Básicos

flask --app *App_Hola_Mundo* run --host 0.0.0.0 --port = 8080

App_Hola_Mundo. <http://localhost:8080/hello/>

curl <http://localhost:8080/hello/>

Ejemplos Básicos

```
from flask import Flask
app = Flask(__name__)
@app.route('/hello/', methods=['GET', 'POST'])
def welcome():
    return "Hello World!"
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Ejemplos Básicos

- ✓ **App_2. /<int:number>/**
 - ✓ Retorna number++
- ✓ **App_3. /person/**
 - ✓ Retorna un JSON
- ✓ **App_4. /numbers/**
 - ✓ Devuelve una lista

Introducción a la Actividad Portal Web

- Se creará una pequeña aplicación web con ayuda de la biblioteca **psycopg2**, un conector de base de datos de PostgreSQL , que permite interactuar con su base de datos de PostgreSQL en Python.
- Apoyados en APIs Rest para su gestión.
- Se emplea Flask para realizar tareas básicas, como conectarse a un servidor de base de datos, crear tablas, insertar datos en una tabla y recuperar datos de una tabla.

Psycopg

- El conector de base de datos PostgreSQL más popular para el lenguaje de programación Python.
- Implementación completa de la especificación Python DB API 2.0 y la seguridad de subprocessos (varios subprocessos pueden compartir la misma conexión).
- Diseñado para aplicaciones con múltiples subprocessos que crean y destruyen muchos cursores y hacen una gran cantidad de "INSERT" o "UPDATE" simultáneos.
- Psycopg 2 es compatible tanto con Unicode como con Python 3.

Pasos

- ❑ Creación de la BD en PostgreSQL y el usuario.
- ❑ Instalación de psycopg2
 - ❑ `pip install psycopg2-binary` (que no requiere un compilador ni bibliotecas externas)
- ❑ Creación e inserción en la BD. ([python init_db.py](#))
- ❑ Despliegue y discusión de los Scripts

Variables de entorno

- Linux `export DB_USERNAME="XXX"`
- `export DB_PASSWORD="password"`
- Windows
- `Setx DB_USERNAME "XXX"`