

REST API endpoint

sábado, 9 de noviembre de 2024 10:17 p. m.

Configuración del ambiente virtual de python

```
# Para instalar el módulo de Flask
01: pip install flask
```

Importación de los módulos instalados

```
"""
Es importante que todo esto se realice en un documento
de python con el nombre 'app.py'
"""
01: from flask import Flask, request
```

Creación de una aplicación de Flask

```
"""
Para poder crear aplicación de Flask con la posibilidad de ejecutar la
aplicación.
"""
01: app = Flask(__name__)
```

Para levantar el servido y que nuestra API puedo retornar valores

```
# Se ejecuta en la terminal
01: flask run
```

Estructura JSON

```
"""
Para manipular datos, es necesario manejar una estructura de datos llamada
JSON. Esta estructura fue diseñada para resultar más cómoda al usuario y no
para optimizar los tiempos de procesamiento de las consultas.
"""
```

Es necesario verificar la compatibilidad de las palabras reservadas de Python con el formato que JSON acepta, como lo es el caso de los booleanos, que en Python son "True" y "False", pero en JSON son "true", "false".

```
"""
01: Stores = [
    {
        "name": "My Store",
        "items": [
            {
                "name": "Chair",
                "price": 15.99
            }
        ]
    }
]
```

Creación de un EndPoint (Método GET)

```
"""
Es la manera en la que se retornar el resultado de una consulta, dentro de
```

este apartado se especifican los datos que, a través de un JSON, serán enviados.

"""

```
01: @app.get("/store")
02: def get_stores():
    # Se retorna en la petición a la API lo siguiente
    return {"stores": stores}
```

Creación de registros a través de la API (Método POST)

```
01: @app.post("/store")
02: def create_store():
    # Convertimos el JSON recibido en un diccionario de Python
    request_data = request.get_json()
    # Creamos un diccionario auxiliar
    new_store = {"name": request_data["name"], "items": []}
    # Agregamos la información obtenida al diccionario existente
    stores.append(new_store)
    # Retornamos los datos ingresados y un código de estado
    # 200 = Todo está bien
    # 201 = Se han aceptado los datos y se creará el registro
    return new_store, 201
```