

Instrucciones Ejecutar API

Diego Iván Perea Montealegre diego.perea@uao.edu.co

Ir a la carpeta donde están todos los archivos requeridos:

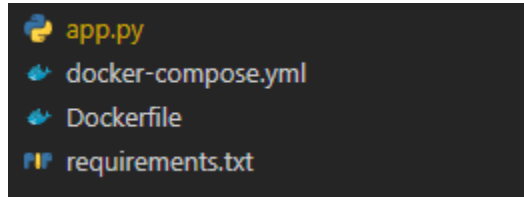


Figura 1 . Archivos necesarios para ejecutar API

En la terminal ejecutar el comando “docker-compose up” para crear contenedores e inicializarlos

```
PS C:\Users\User\Desktop\api-tecnica> docker-compose up
[+] Running 2/2
 - Container api-tecnica-flask-app-1 Created                                0.4s
 - Container mycontainer Created                                           0.4s
Attaching to api-tecnica-flask-app-1, mycontainer
```

Figura 2. Comando Docker compose up

Cuando este listo se visualizara el siguiente mensaje :

```
api-tecnica-flask-app-1 | warnings.warn(
api-tecnica-flask-app-1 | * Serving Flask app 'app.py'
api-tecnica-flask-app-1 | * Debug mode: off
api-tecnica-flask-app-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server i
nstead.
api-tecnica-flask-app-1 | * Running on all addresses (0.0.0.0)
api-tecnica-flask-app-1 | * Running on http://127.0.0.1:5000
```

Figura 3 . Mensaje de confirmación de creación de contenedores y inicializados.

También en Vscode se puede observar que están ejecutándose con el símbolo play :

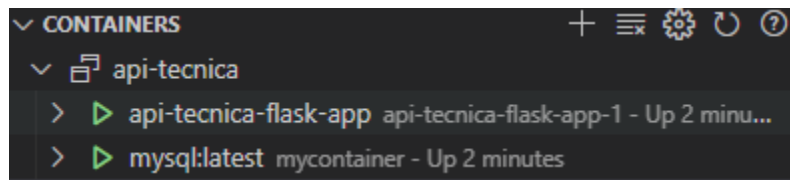


Figura 4. Contenedores inicializados

Usando Postman o herramientas para realizar request se da la ruta que seria : <http://127.0.0.1:5000/api/data> con el método GET, que esta ruta nos traerá los nombres de los países del mundo de una api externa :

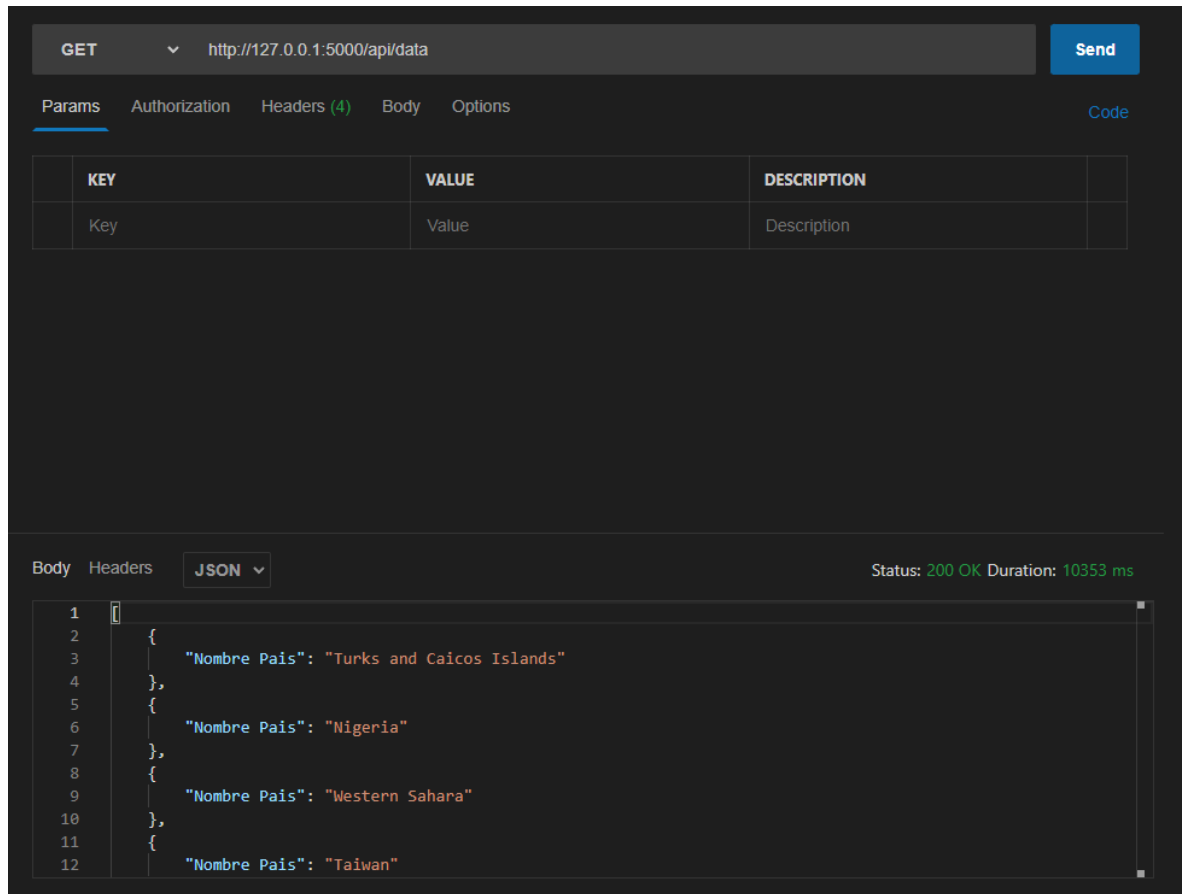


Figura 5. Método GET de obtención de datos API externa

Para almacenar los nombres de los países en la base de datos del contenedor de mysql se realiza el método POST con la misma ruta , dando como salida el data de los nombre de los países que se guardaron :

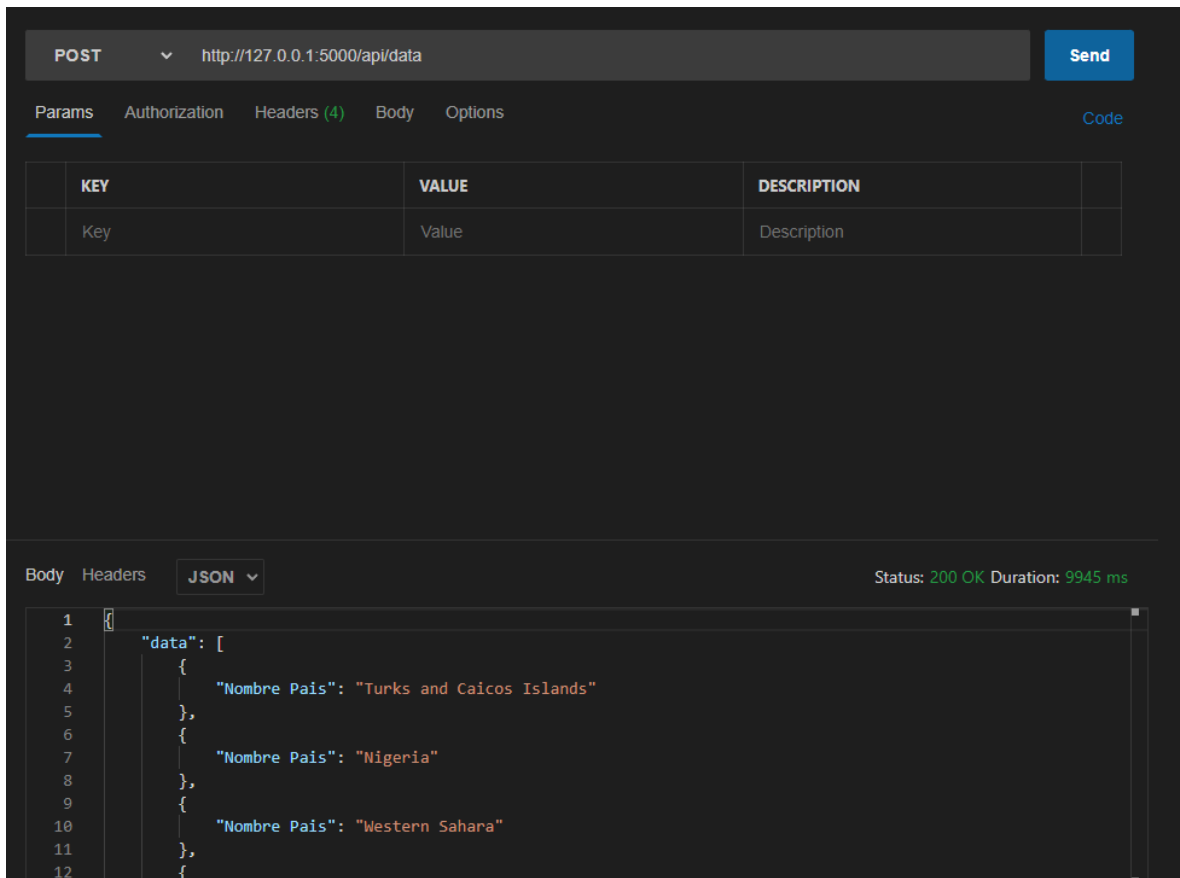


Figura 6. Método POST de almacenamiento de datos API externa

Para obtener usando GET , el nombre del país según el id , se da con la ruta <http://127.0.0.1:5000/api/data/{id}> ,que en este caso se seleccionó el id con 249:

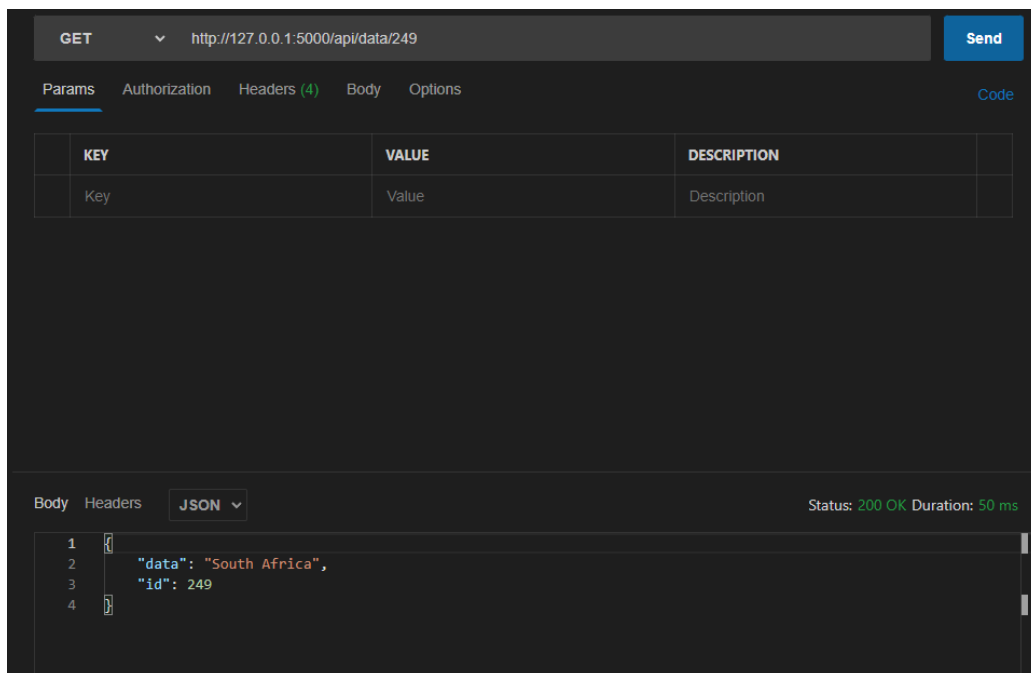


Figura 7. Método GET de obtención de nombre del país según id.

Para visualizar que el almacenamiento de los datos es correcto y su id es correcto , toca entrar al contenedor y comprobarlo, Se realiza con el siguiente comando : `docker exec -it {nombre contenedor mysql}bash` , en este caso el nombre del contenedor mysql es “mycontainer” entonces para este caso seria : `docker exec -it mycontainer bash`

```
PS C:\Users\User\Desktop\api-tecnica> docker exec -it mycontainer bash
bash-4.4#
```

Listo , para entrar a mysql es con el siguiente comando : `mysql -u root -p`

En donde se splicita la contraseña que en este caso es “mypassword”

```
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Con el comando `show databases;` se mira las abses de datos creadas , en donde se observa que ya esta creada nuestra base de datos flaskmysql .

```
mysql> show databases;
+-----+
| Database |
+-----+
| flaskmysql |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.07 sec)
```

Le damos el comando `use flaskmysql;` para usarla y mirar la tabla creada , que se ahce con el comando `show tables;`

```
Database changed
mysql> show tables;
+-----+
| Tables_in_flaskmysql |
+-----+
| data                  |
+-----+
1 row in set (0.01 sec)
```

Para mirar los datos de la tabla se ejecuta el comando “select * from data ;”

```
mysql> select * from data;
+----+-----+
| id | data                |
+----+-----+
| 1  | Turks and Caicos Islands |
| 2  | Nigeria              |
| 3  | Western Sahara        |
| 4  | Taiwan                |
| 5  | Costa Rica            |
| 6  | Benin                 |
+----+-----+
```

```

| 240 | Montenegro
| 241 | San Marino
| 242 | South Korea
| 243 | Uruguay
| 244 | Colombia
| 245 | Anguilla
| 246 | American Samoa
| 247 | Micronesia
| 248 | Sri Lanka
| 249 | South Africa
| 250 | Nicaragua
+----+-----+
250 rows in set (0.00 sec)
```

Se comprueba que tiene 250 filas en donde, el GET del id es correcto y que los datos fueron exitosamente almacenados.

Nota : Se usó SQLAlchemy por su facilidad de cambio de base de datos si se desea usar una diferente como postgresql , sqlite u otras compatibles.

Se usó Marshmallow debido a la facilidad de operar mediante un sistema poo sin usar directamente consultar sql , por lo que la escalabilidad de la app se mas fácil e útil al momento de leer el código e implementarlo.

Se usó flask debido a la gran variedad de bibliotecas para usar y facilidad de uso para realizar de cosas pequeñas a realizar cosas grandes por lo que si el proyecto de esta API se vuelve mas grande flask lo puede realizar sin ningún inconveniente. Y además de la cantidad de información y documentación que hay sobre la plataforma flask.