

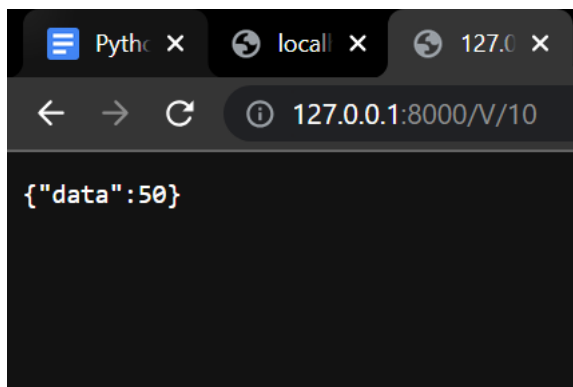
Task1

Step1 - Before moving further, we have to install some of the Python modules, such as pandas, fastapi, uvicorn, mysql.connector, and live server.

Step2 - To run main.py (the API file), write this in the terminal: Uvicorn main: app --reload

Step 3 - It will activate your local host. Then go to the link: <http://127.0.0.1:8000/>.

Step4 - To request the content from a.csv file using the API. Request the API link as http://127.0.0.1:8000/column_name/row_number. as local is will encrypted end point to request from the server.

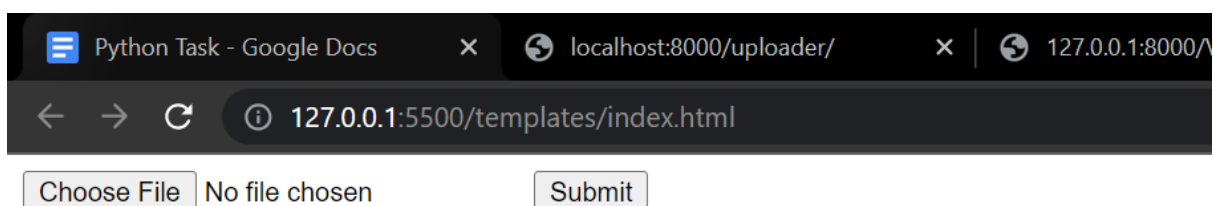


Then you can fetch the data from the CSV file from the local host.

Data has been extracted from the.csv file using the Pandas module and stored in the MySQL database. and further, that data can be exported or manipulated on the server.

Uploading the file

To upload the data to the server, go to index.html and go live using the live server, and then you will get the form to upload the data. upload the.csv file



Integration of python and API

```
import paho.mqtt.client as mqtt

# MQTT broker information
broker_address = 'mqtt.example.com'
broker_port = 1883

# Callback function for connection status
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print('Connected to MQTT broker')
    else:
        print('Connection failed')

# Callback function for receiving messages
def on_message(client, userdata, msg):
    print('Received message: ' + msg.payload.decode())

# Create MQTT client
client = mqtt.Client()

# Set connection and message callbacks
client.on_connect = on_connect
client.on_message = on_message

# Connect to MQTT broker
client.connect(broker_address, broker_port, 60)

# Subscribe to a topic
client.subscribe('sensors/temperature')

# Publish a message
client.publish('sensors/humidity', '50')

# Keep the client running to receive messages
client.loop_forever()
```

In this example, Python utilizes the `paho.mqtt.client` library to connect to an MQTT broker running on an IoT gateway device. It sets up callback functions for connection status and message reception. The client connects to the broker, subscribes to a topic for receiving messages, and publishes a message to another topic. Finally, the client enters a loop to maintain the connection and continuously receive messages.

