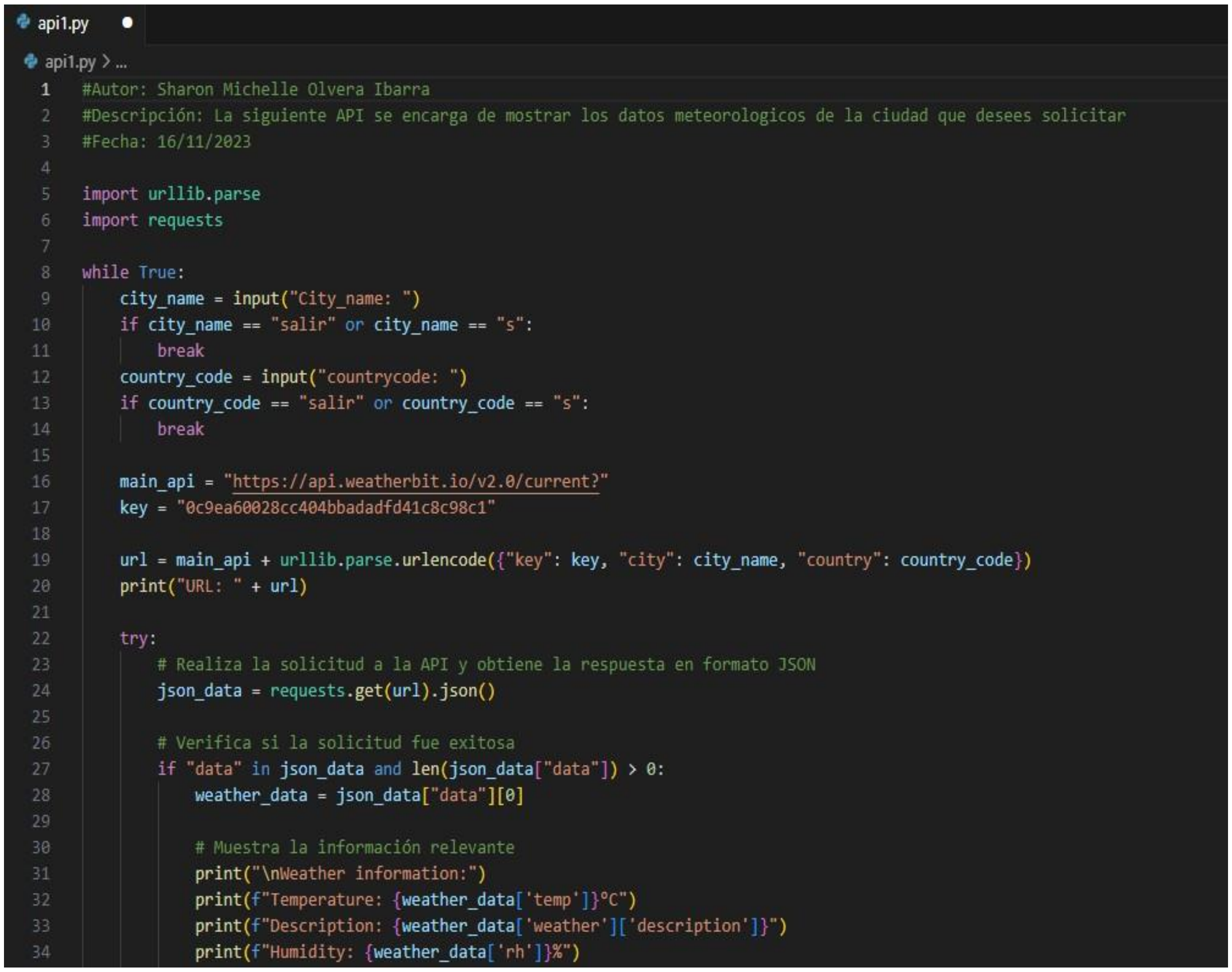
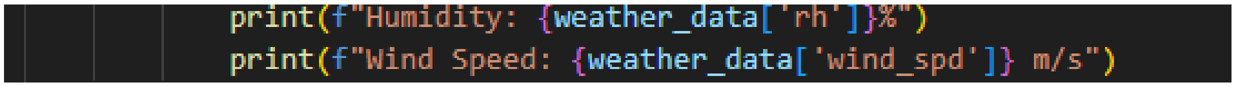
Olvera Ibarra Sharon Michelle

Actividad I API´s de terceros
Current Weather API

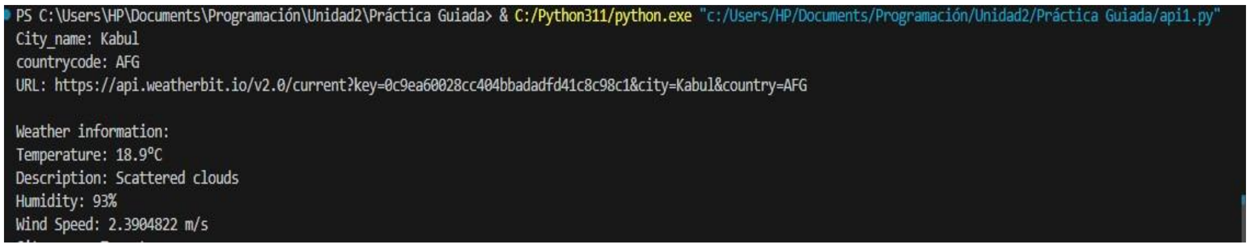


```python
#Autor: Sharon Michelle Olvera Ibarra
#Descripción: La siguiente API se encarga de mostrar los datos meteorologicos de la ciudad que desees solicitar
#Fecha: 16/11/2023

import urllib.parse
import requests

while True:
    city_name = input("City_name: ")
    if city_name == "salir" or city_name == "s":
        break
    country_code = input("countrycode: ")
    if country_code == "salir" or country_code == "s":
        break

    main_api = "https://api.weatherbit.io/v2.0/current?"
    key = "0c9ea60028cc404bbadadfd41c8c98c1"

    url = main_api + urllib.parse.urlencode({"key": key, "city": city_name, "country": country_code})
    print("URL: " + url)

    try:
        # Realiza la solicitud a la API y obtiene la respuesta en formato JSON
        json_data = requests.get(url).json()

        # Verifica si la solicitud fue exitosa
        if "data" in json_data and len(json_data["data"]) > 0:
            weather_data = json_data["data"][0]

            # Muestra la información relevante
            print("\nWeather information:")
            print(f"Temperature: {weather_data['temp']}°C")
            print(f"Description: {weather_data['weather']['description']}")
            print(f"Humidity: {weather_data['rh']}%")
```



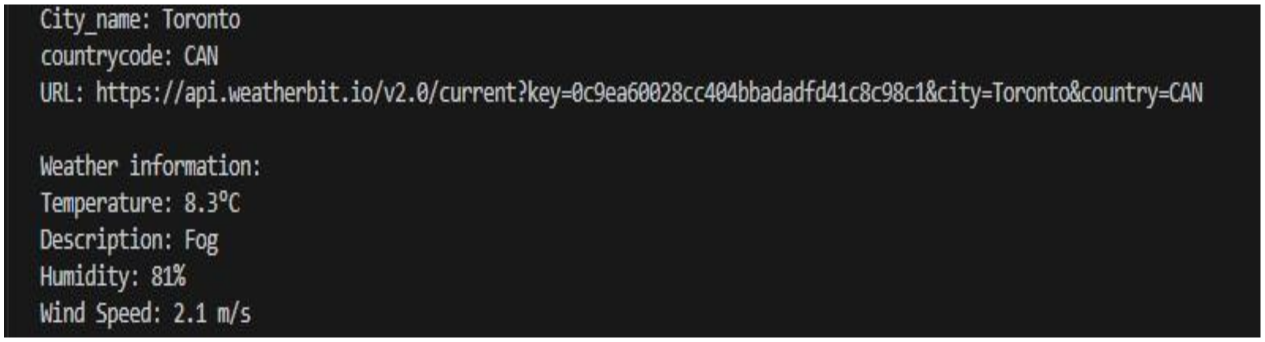
```python
            print(f"Humidity: {weather_data['rh']}%")
            print(f"Wind Speed: {weather_data['wind_spd']} m/s")
```

Pruebas de varias corridas:

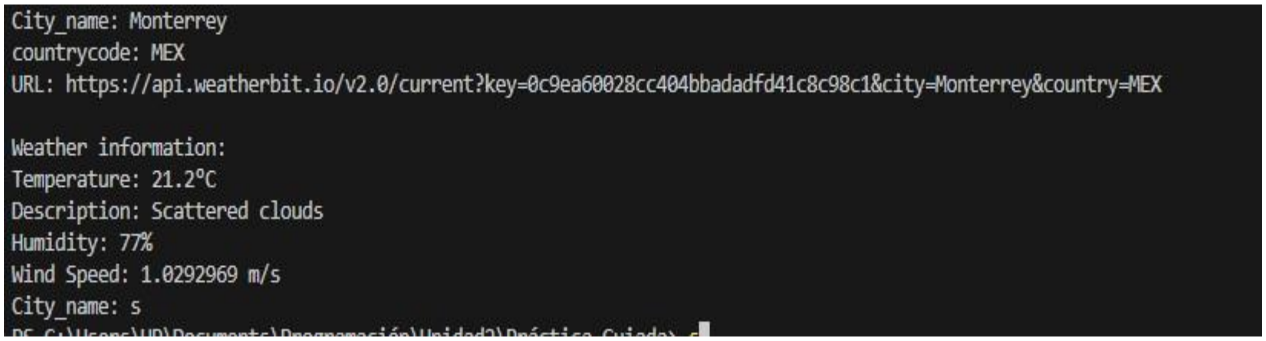


```
PS C:\Users\HP\Documents\Programación\Unidad2\Práctica Guiada> & C:/Python311/python.exe "c:/Users/HP/Documents/Programación/Unidad2/Práctica Guiada/api1.py"
City_name: Kabul
countrycode: AFG
URL: https://api.weatherbit.io/v2.0/current?key=0c9ea60028cc404bbadadfd41c8c98c1&city=Kabul&country=AFG

Weather information:
Temperature: 18.9°C
Description: Scattered clouds
Humidity: 93%
Wind Speed: 2.3904822 m/s
```

Olvera Ibarra Sharon Michelle

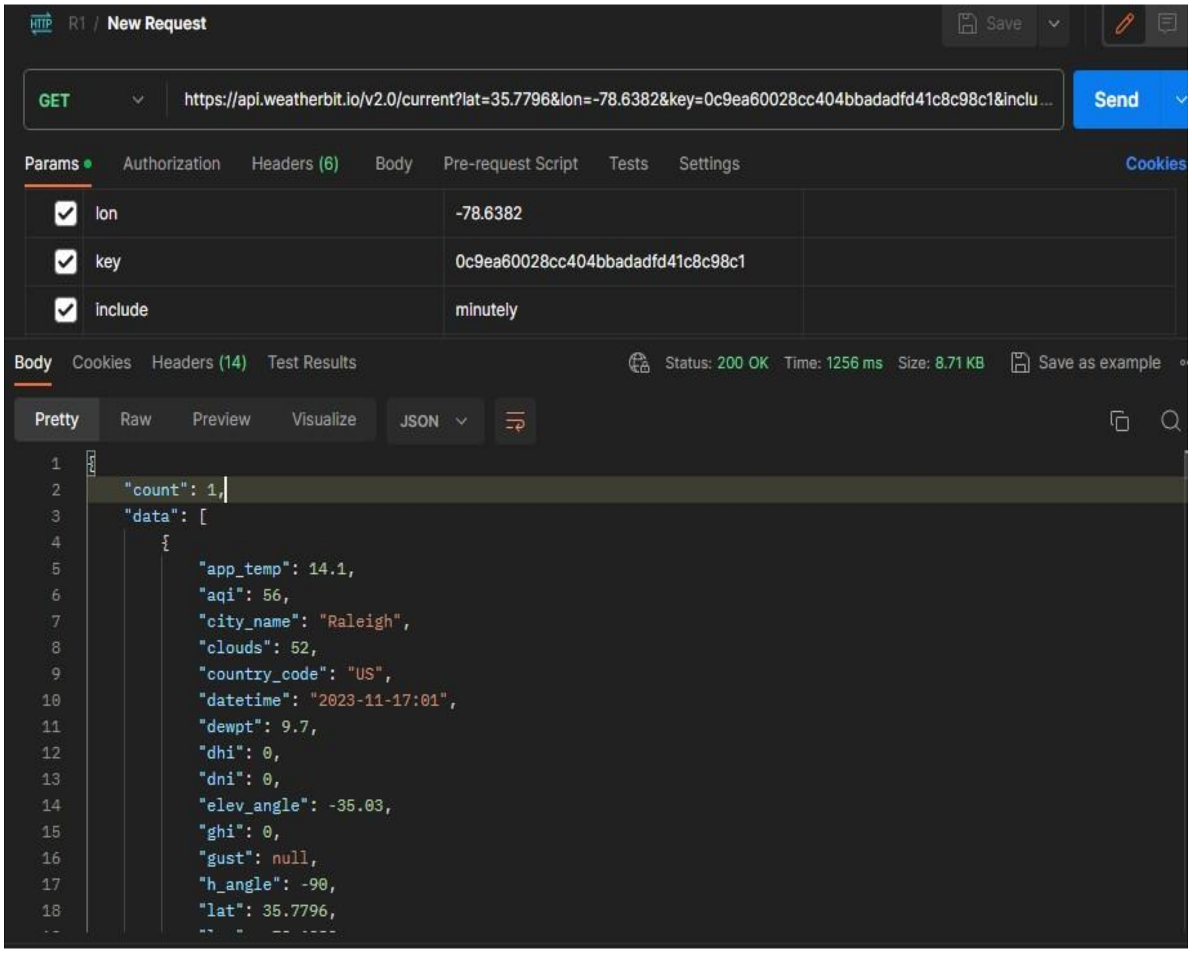


```
City_name: Toronto
countrycode: CAN
URL: https://api.weatherbit.io/v2.0/current?key=0c9ea60028cc404bbadadfd41c8c98c1&city=Toronto&country=CAN

Weather information:
Temperature: 8.3°C
Description: Fog
Humidity: 81%
Wind Speed: 2.1 m/s
```



```
City_name: Monterrey
countrycode: MEX
URL: https://api.weatherbit.io/v2.0/current?key=0c9ea60028cc404bbadadfd41c8c98c1&city=Monterrey&country=MEX

Weather information:
Temperature: 21.2°C
Description: Scattered clouds
Humidity: 77%
Wind Speed: 1.0292969 m/s
City_name: s
PS C:\Users\HP\Documents\Programación\Unidad2\Práctica Guiada> s
```

GET con postman:



R1 / **New Request**

GET https://api.weatherbit.io/v2.0/current?lat=35.7796&lon=-78.6382&key=0c9ea60028cc404bbadadfd41c8c98c1&inclu...

Params | Authorization | Headers (6) | Body | Pre-request Script | Tests | Settings | Cookies

✓ lon	-78.6382
✓ key	0c9ea60028cc404bbadadfd41c8c98c1
✓ include	minutely

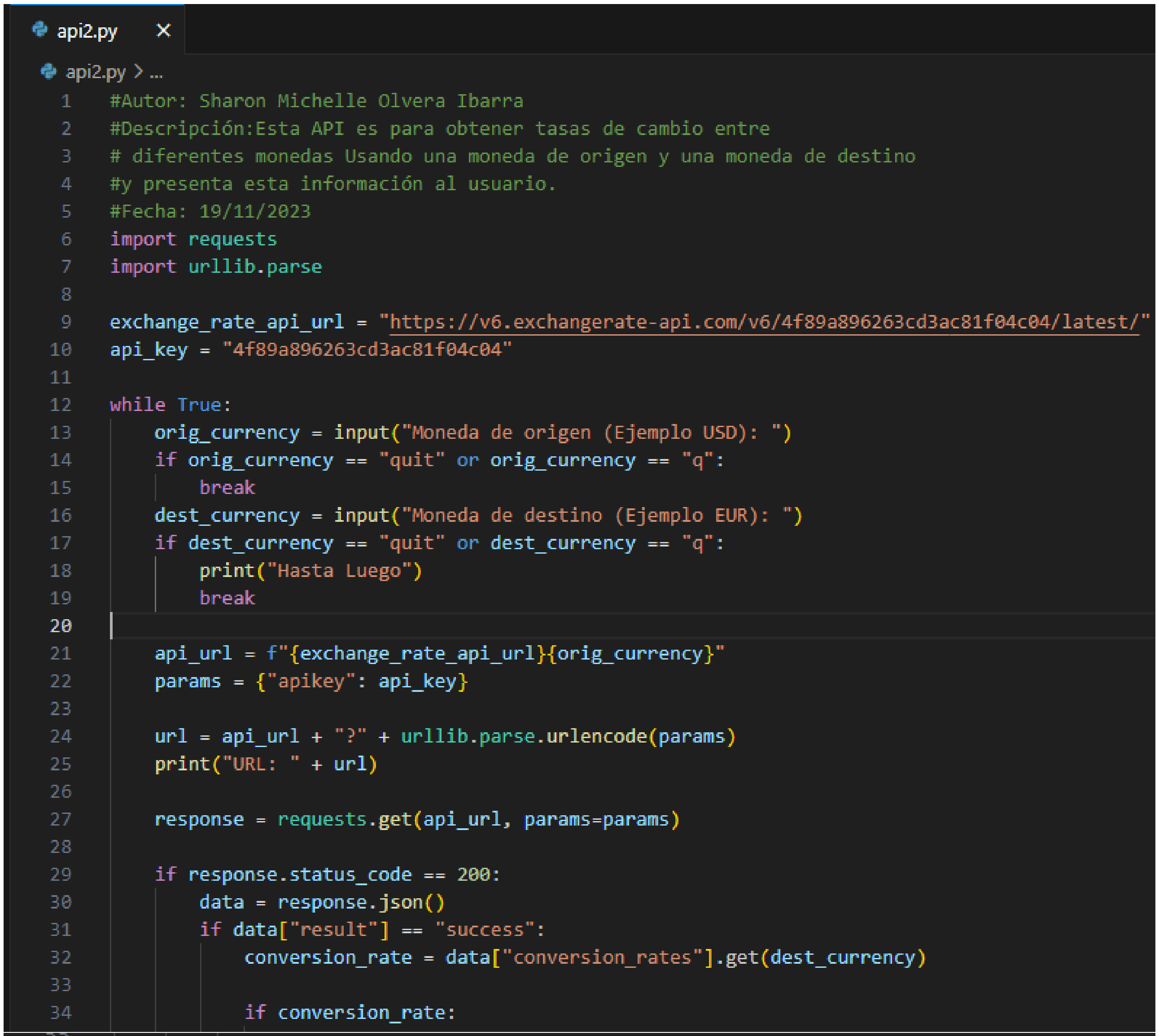
Body | Cookies | Headers (14) | Test Results — Status: 200 OK Time: 1256 ms Size: 8.71 KB Save as example

Pretty | Raw | Preview | Visualize | JSON

```
{
    "count": 1,
    "data": [
        {
            "app_temp": 14.1,
            "aqi": 56,
            "city_name": "Raleigh",
            "clouds": 52,
            "country_code": "US",
            "datetime": "2023-11-17:01",
            "dewpt": 9.7,
            "dhi": 0,
            "dni": 0,
            "elev_angle": -35.03,
            "ghi": 0,
            "gust": null,
            "h_angle": -90,
            "lat": 35.7796,
```

Olvera Ibarra Sharon Michelle

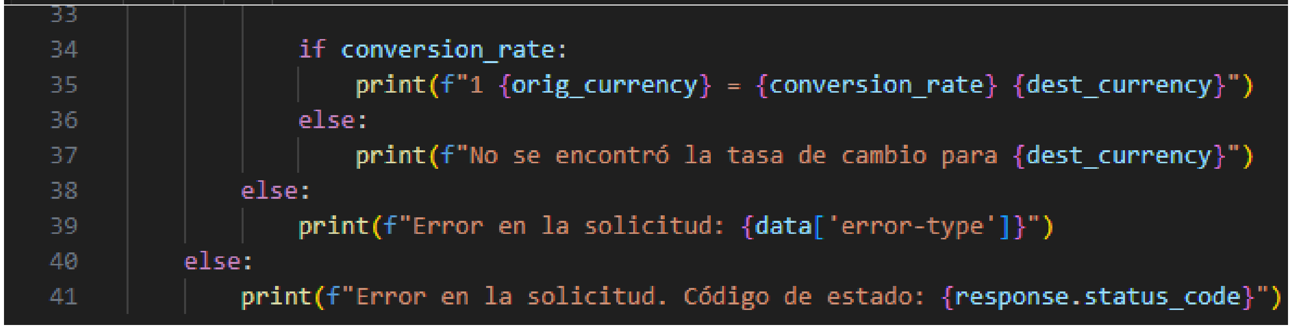
ExchangeRate-API



api2.py

api2.py > ...

```
#Autor: Sharon Michelle Olvera Ibarra
#Descripción:Esta API es para obtener tasas de cambio entre
# diferentes monedas Usando una moneda de origen y una moneda de destino
#y presenta esta información al usuario.
#Fecha: 19/11/2023
import requests
import urllib.parse

exchange_rate_api_url = "https://v6.exchangerate-api.com/v6/4f89a896263cd3ac81f04c04/latest/"
api_key = "4f89a896263cd3ac81f04c04"

while True:
    orig_currency = input("Moneda de origen (Ejemplo USD): ")
    if orig_currency == "quit" or orig_currency == "q":
        break
    dest_currency = input("Moneda de destino (Ejemplo EUR): ")
    if dest_currency == "quit" or dest_currency == "q":
        print("Hasta Luego")
        break

    api_url = f"{exchange_rate_api_url}{orig_currency}"
    params = {"apikey": api_key}

    url = api_url + "?" + urllib.parse.urlencode(params)
    print("URL: " + url)

    response = requests.get(api_url, params=params)

    if response.status_code == 200:
        data = response.json()
        if data["result"] == "success":
            conversion_rate = data["conversion_rates"].get(dest_currency)

            if conversion_rate:
```

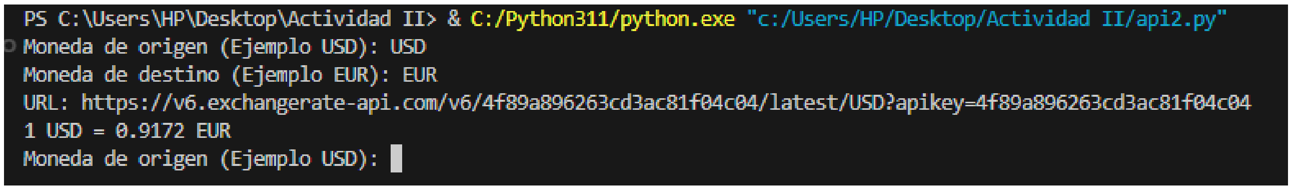


```
            if conversion_rate:
```

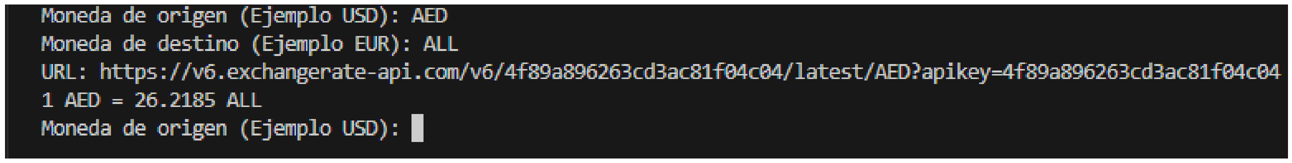
```
                print(f"1 {orig_currency} = {conversion_rate} {dest_currency}")
            else:
                print(f"No se encontró la tasa de cambio para {dest_currency}")
        else:
            print(f"Error en la solicitud: {data['error-type']}")
    else:
        print(f"Error en la solicitud. Código de estado: {response.status_code}")
```

Olvera Ibarra Sharon Michelle

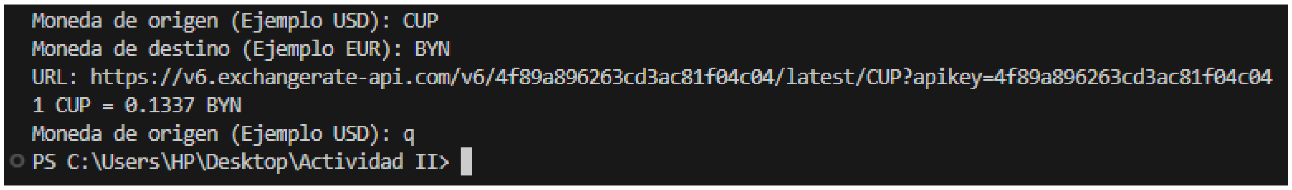
Pruebas de varias corridas:



```
PS C:\Users\HP\Desktop\Actividad II> & C:/Python311/python.exe "c:/Users/HP/Desktop/Actividad II/api2.py"
Moneda de origen (Ejemplo USD): USD
Moneda de destino (Ejemplo EUR): EUR
URL: https://v6.exchangerate-api.com/v6/4f89a896263cd3ac81f04c04/latest/USD?apikey=4f89a896263cd3ac81f04c04
1 USD = 0.9172 EUR
Moneda de origen (Ejemplo USD):
```

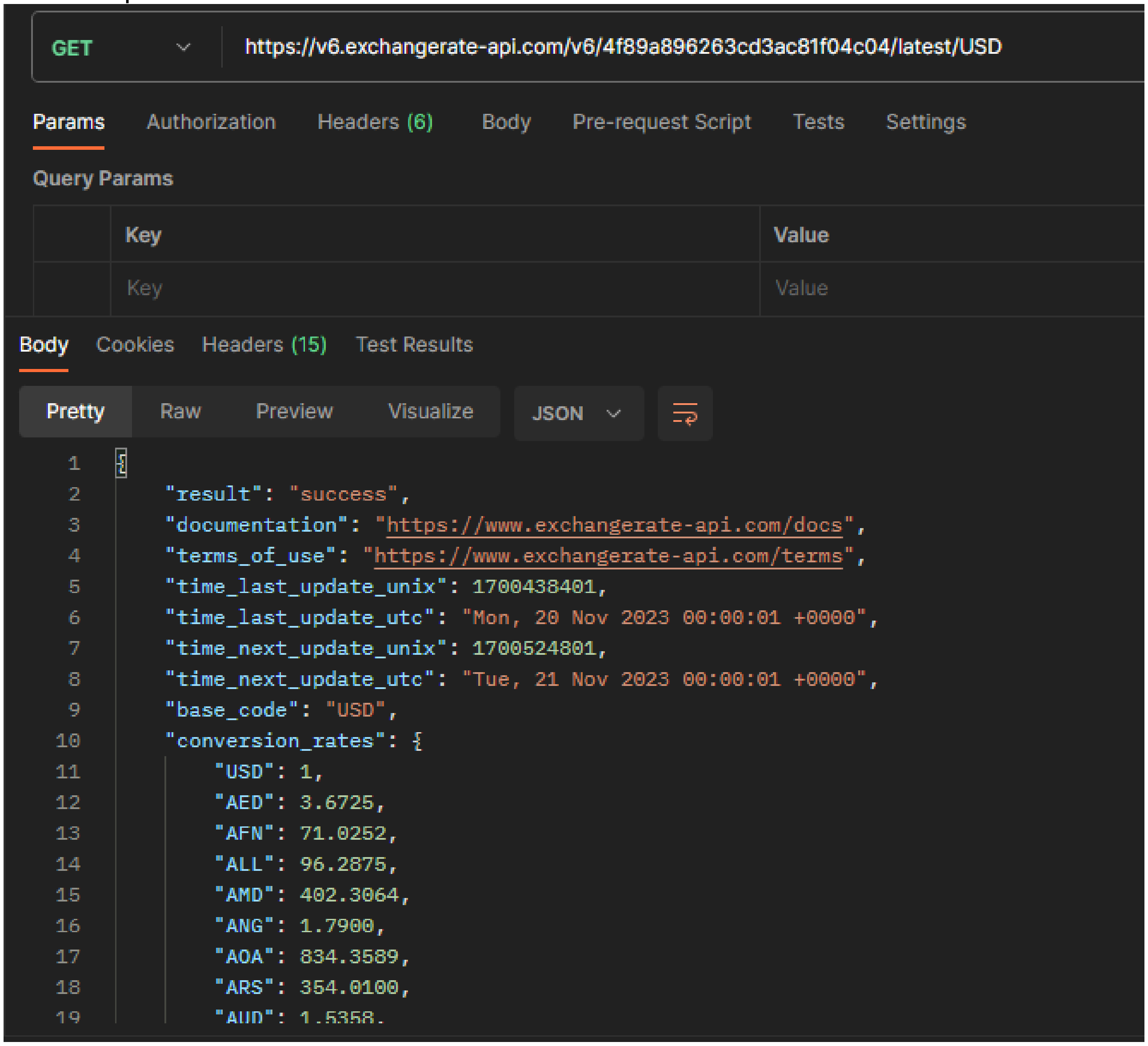


```
Moneda de origen (Ejemplo USD): AED
Moneda de destino (Ejemplo EUR): ALL
URL: https://v6.exchangerate-api.com/v6/4f89a896263cd3ac81f04c04/latest/AED?apikey=4f89a896263cd3ac81f04c04
1 AED = 26.2185 ALL
Moneda de origen (Ejemplo USD):
```



```
Moneda de origen (Ejemplo USD): CUP
Moneda de destino (Ejemplo EUR): BYN
URL: https://v6.exchangerate-api.com/v6/4f89a896263cd3ac81f04c04/latest/CUP?apikey=4f89a896263cd3ac81f04c04
1 CUP = 0.1337 BYN
Moneda de origen (Ejemplo USD): q
PS C:\Users\HP\Desktop\Actividad II>
```

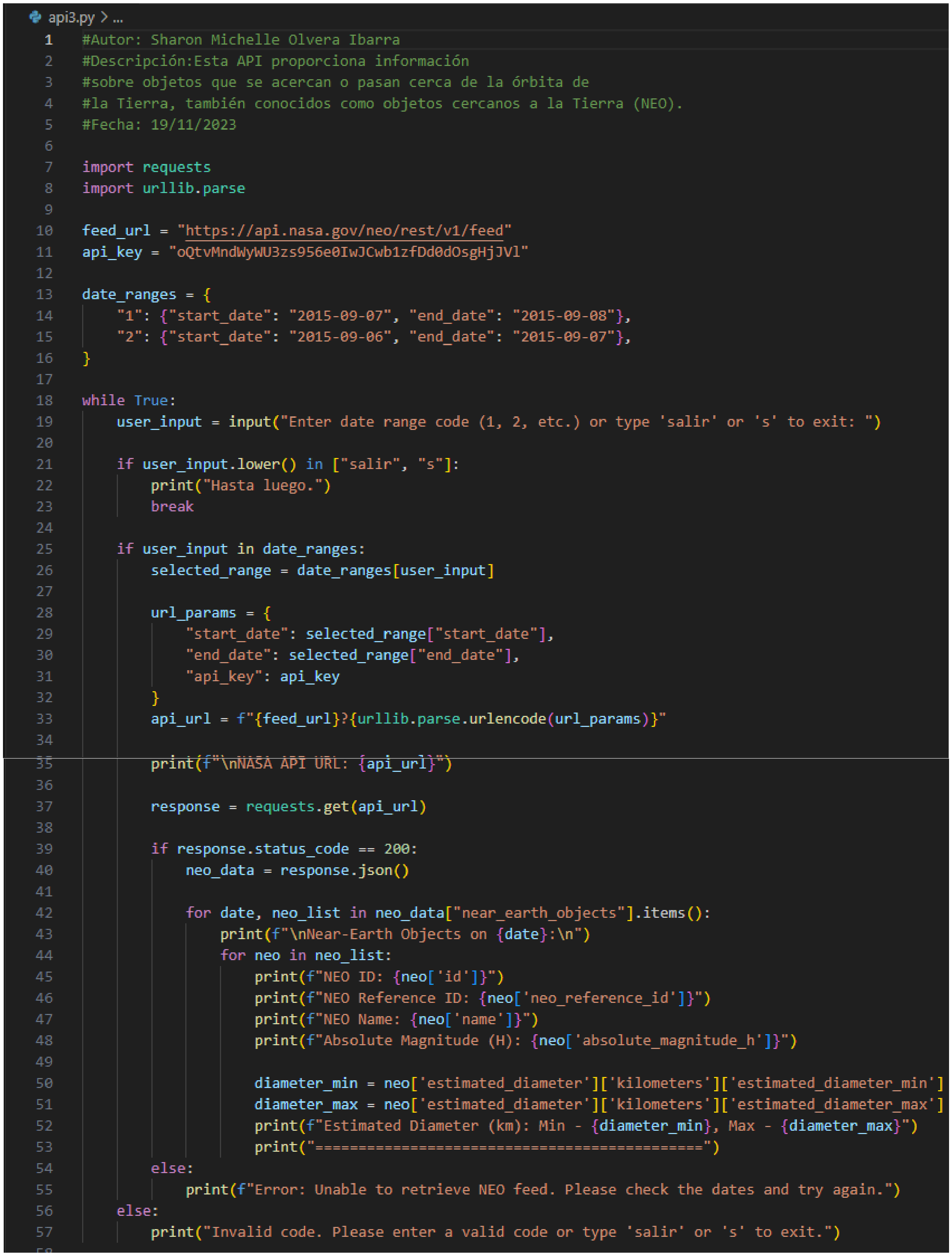
GET con postman:



```
{
    "result": "success",
    "documentation": "https://www.exchangerate-api.com/docs",
    "terms_of_use": "https://www.exchangerate-api.com/terms",
    "time_last_update_unix": 1700438401,
    "time_last_update_utc": "Mon, 20 Nov 2023 00:00:01 +0000",
    "time_next_update_unix": 1700524801,
    "time_next_update_utc": "Tue, 21 Nov 2023 00:00:01 +0000",
    "base_code": "USD",
    "conversion_rates": {
        "USD": 1,
        "AED": 3.6725,
        "AFN": 71.0252,
        "ALL": 96.2875,
        "AMD": 402.3064,
        "ANG": 1.7900,
        "AOA": 834.3589,
        "ARS": 354.0100,
        "AUD": 1.5358,
```

Olvera Ibarra Sharon Michelle

API de la NASA

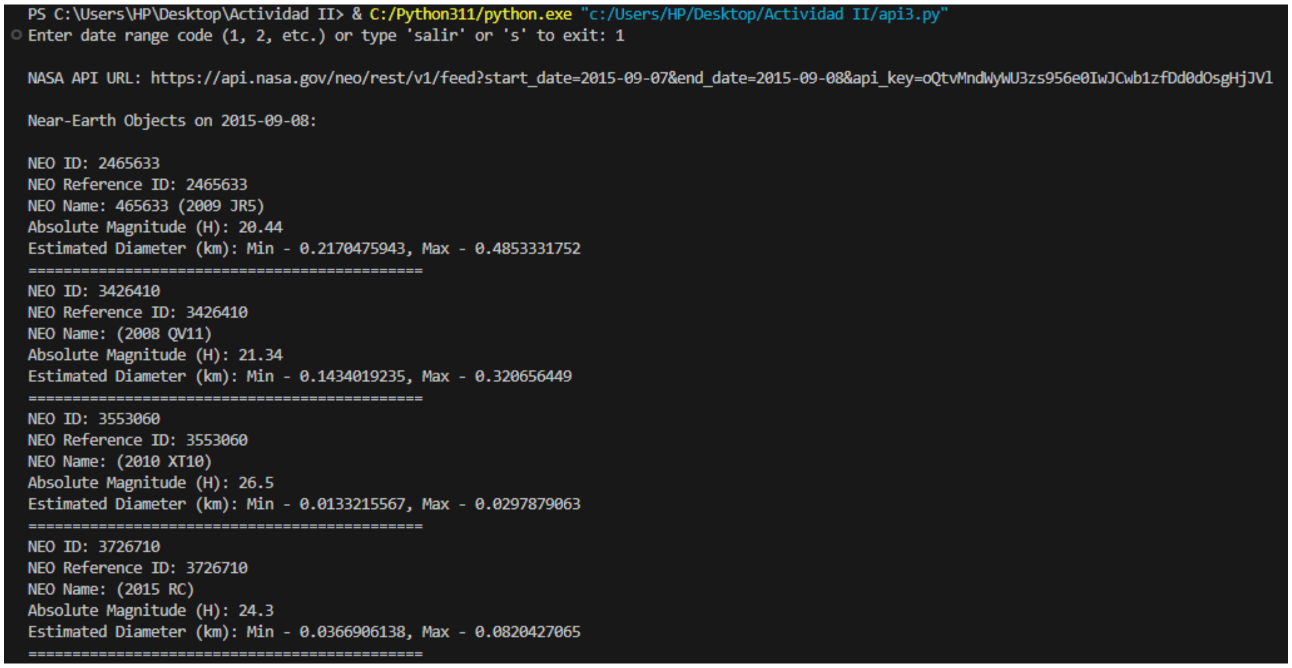


api3.py > ...

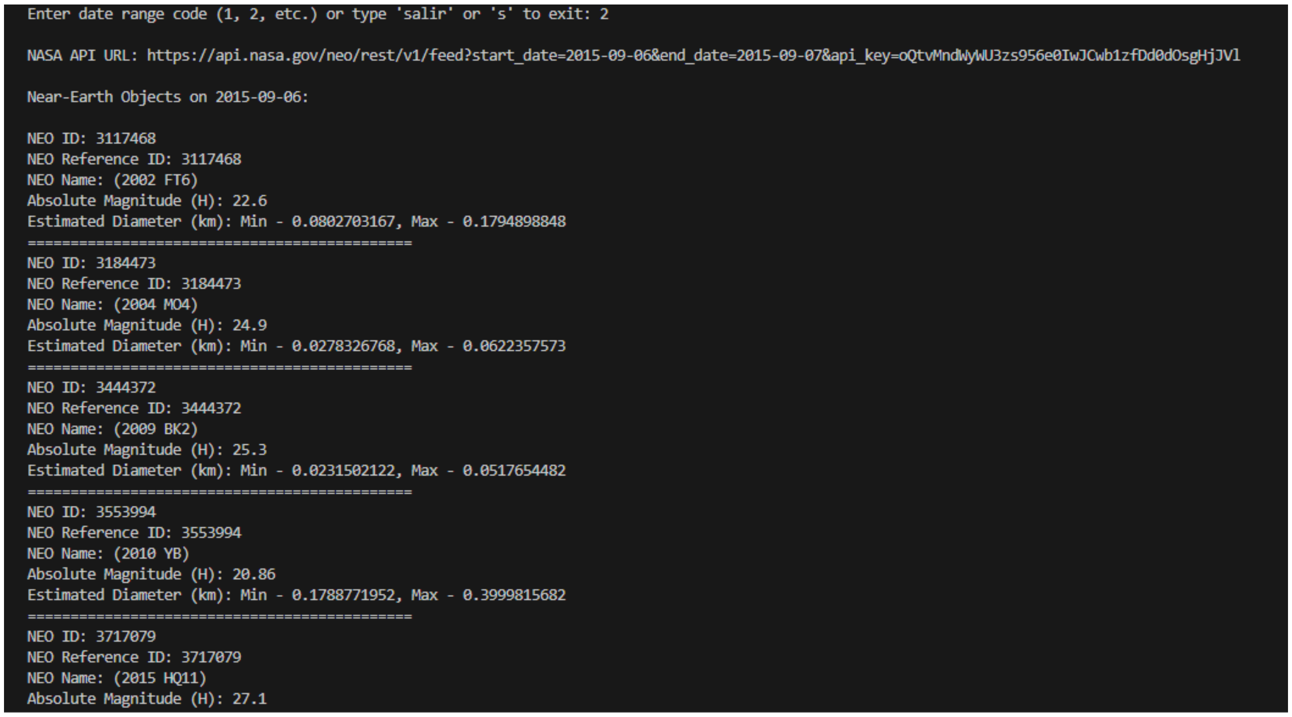
```python
#Autor: Sharon Michelle Olvera Ibarra
#Descripción:Esta API proporciona información
#sobre objetos que se acercan o pasan cerca de la órbita de
#la Tierra, también conocidos como objetos cercanos a la Tierra (NEO).
#Fecha: 19/11/2023

import requests
import urllib.parse

feed_url = "https://api.nasa.gov/neo/rest/v1/feed"
api_key = "oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl"

date_ranges = {
    "1": {"start_date": "2015-09-07", "end_date": "2015-09-08"},
    "2": {"start_date": "2015-09-06", "end_date": "2015-09-07"},
}

while True:
    user_input = input("Enter date range code (1, 2, etc.) or type 'salir' or 's' to exit: ")

    if user_input.lower() in ["salir", "s"]:
        print("Hasta luego.")
        break

    if user_input in date_ranges:
        selected_range = date_ranges[user_input]

        url_params = {
            "start_date": selected_range["start_date"],
            "end_date": selected_range["end_date"],
            "api_key": api_key
        }
        api_url = f"{feed_url}?{urllib.parse.urlencode(url_params)}"

        print(f"\nNASA API URL: {api_url}")

        response = requests.get(api_url)

        if response.status_code == 200:
            neo_data = response.json()

            for date, neo_list in neo_data["near_earth_objects"].items():
                print(f"\nNear-Earth Objects on {date}:\n")
                for neo in neo_list:
                    print(f"NEO ID: {neo['id']}")
                    print(f"NEO Reference ID: {neo['neo_reference_id']}")
                    print(f"NEO Name: {neo['name']}")
                    print(f"Absolute Magnitude (H): {neo['absolute_magnitude_h']}")

                    diameter_min = neo['estimated_diameter']['kilometers']['estimated_diameter_min']
                    diameter_max = neo['estimated_diameter']['kilometers']['estimated_diameter_max']
                    print(f"Estimated Diameter (km): Min - {diameter_min}, Max - {diameter_max}")
                    print("=============================================")
        else:
            print(f"Error: Unable to retrieve NEO feed. Please check the dates and try again.")
    else:
        print("Invalid code. Please enter a valid code or type 'salir' or 's' to exit.")
```

Olvera Ibarra Sharon Michelle

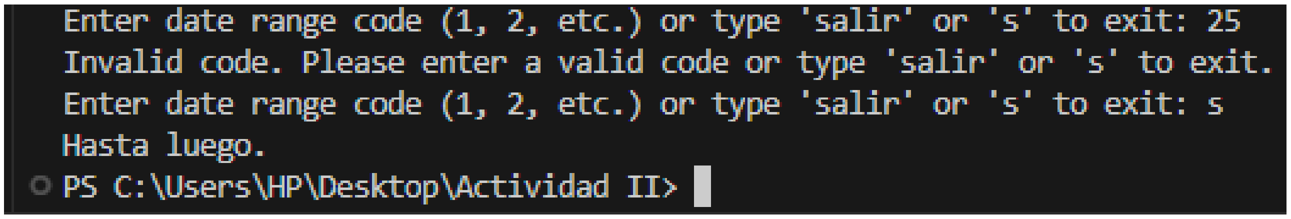
Pruebas de varias corridas:



```
PS C:\Users\HP\Desktop\Actividad II> & C:/Python311/python.exe "c:/Users/HP/Desktop/Actividad II/api3.py"
Enter date range code (1, 2, etc.) or type 'salir' or 's' to exit: 1

NASA API URL: https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&api_key=oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl

Near-Earth Objects on 2015-09-08:

NEO ID: 2465633
NEO Reference ID: 2465633
NEO Name: 465633 (2009 JR5)
Absolute Magnitude (H): 20.44
Estimated Diameter (km): Min - 0.2170475943, Max - 0.4853331752
=============================================
NEO ID: 3426410
NEO Reference ID: 3426410
NEO Name: (2008 QV11)
Absolute Magnitude (H): 21.34
Estimated Diameter (km): Min - 0.1434019235, Max - 0.320656449
=============================================
NEO ID: 3553060
NEO Reference ID: 3553060
NEO Name: (2010 XT10)
Absolute Magnitude (H): 26.5
Estimated Diameter (km): Min - 0.0133215567, Max - 0.0297879063
=============================================
NEO ID: 3726710
NEO Reference ID: 3726710
NEO Name: (2015 RC)
Absolute Magnitude (H): 24.3
Estimated Diameter (km): Min - 0.0366906138, Max - 0.0820427065
=============================================
```



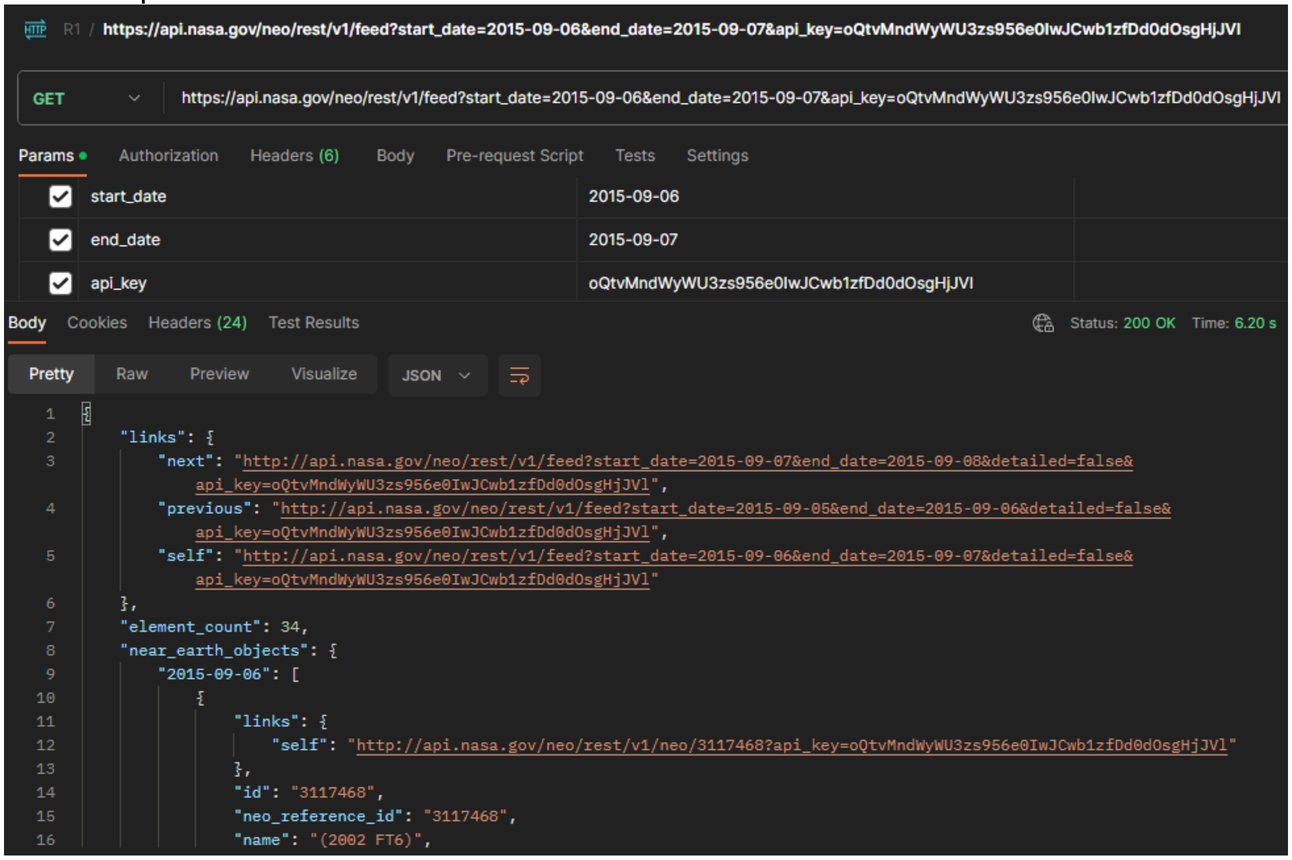
```
Enter date range code (1, 2, etc.) or type 'salir' or 's' to exit: 2

NASA API URL: https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-06&end_date=2015-09-07&api_key=oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl

Near-Earth Objects on 2015-09-06:

NEO ID: 3117468
NEO Reference ID: 3117468
NEO Name: (2002 FT6)
Absolute Magnitude (H): 22.6
Estimated Diameter (km): Min - 0.0802703167, Max - 0.1794898848
=============================================
NEO ID: 3184473
NEO Reference ID: 3184473
NEO Name: (2004 MO4)
Absolute Magnitude (H): 24.9
Estimated Diameter (km): Min - 0.0278326768, Max - 0.0622357573
=============================================
NEO ID: 3444372
NEO Reference ID: 3444372
NEO Name: (2009 BK2)
Absolute Magnitude (H): 25.3
Estimated Diameter (km): Min - 0.0231502122, Max - 0.0517654482
=============================================
NEO ID: 3553994
NEO Reference ID: 3553994
NEO Name: (2010 YB)
Absolute Magnitude (H): 20.86
Estimated Diameter (km): Min - 0.1788771952, Max - 0.3999815682
=============================================
NEO ID: 3717079
NEO Reference ID: 3717079
NEO Name: (2015 HQ11)
Absolute Magnitude (H): 27.1
```



```
Enter date range code (1, 2, etc.) or type 'salir' or 's' to exit: 25
Invalid code. Please enter a valid code or type 'salir' or 's' to exit.
Enter date range code (1, 2, etc.) or type 'salir' or 's' to exit: s
Hasta luego.
PS C:\Users\HP\Desktop\Actividad II>
```

Olvera Ibarra Sharon Michelle

GET con postman:



R1 / https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-06&end_date=2015-09-07&api_key=oQtvMndWyWU3zs956e0lwJCwb1zfDd0dOsgHjJVl

GET https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-06&end_date=2015-09-07&api_key=oQtvMndWyWU3zs956e0lwJCwb1zfDd0dOsgHjJVl

Params · Authorization · Headers (6) · Body · Pre-request Script · Tests · Settings

✓	start_date	2015-09-06
✓	end_date	2015-09-07
✓	api_key	oQtvMndWyWU3zs956e0lwJCwb1zfDd0dOsgHjJVl

Body · Cookies · Headers (24) · Test Results — Status: 200 OK Time: 6.20 s

Pretty · Raw · Preview · Visualize · JSON

```
{
    "links": {
        "next": "http://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&detailed=false&
            api_key=oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl",
        "previous": "http://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-05&end_date=2015-09-06&detailed=false&
            api_key=oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl",
        "self": "http://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-06&end_date=2015-09-07&detailed=false&
            api_key=oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl"
    },
    "element_count": 34,
    "near_earth_objects": {
        "2015-09-06": [
            {
                "links": {
                    "self": "http://api.nasa.gov/neo/rest/v1/neo/3117468?api_key=oQtvMndWyWU3zs956e0IwJCwb1zfDd0dOsgHjJVl"
                },
                "id": "3117468",
                "neo_reference_id": "3117468",
                "name": "(2002 FT6)",
```