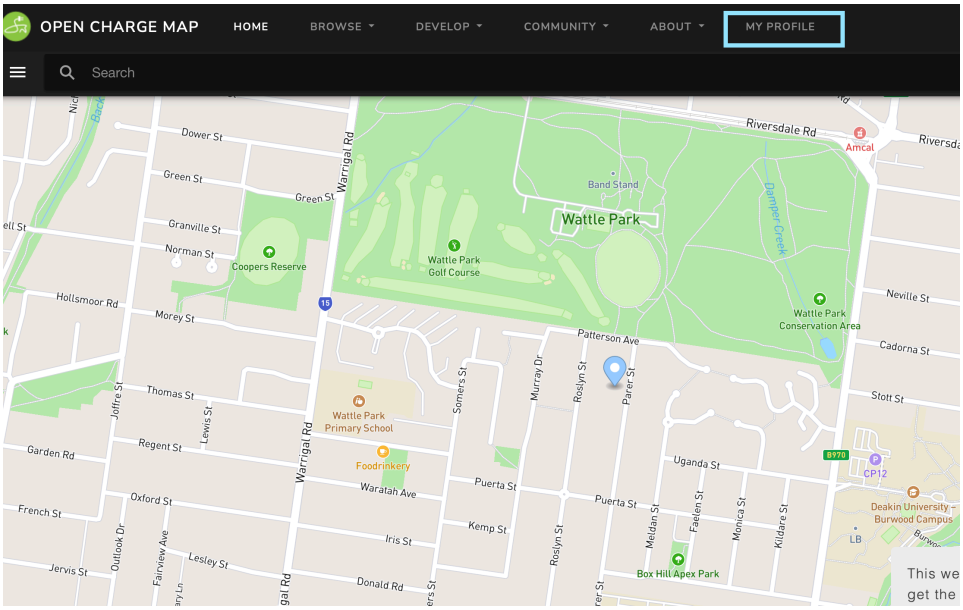
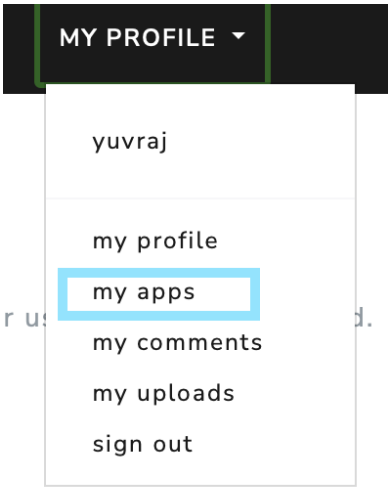
# EV Stations Data Collection Guide using Open Charge Map-

To access the Open Charge Map API and collect data, you'll need an API key. Follow these steps to obtain your API key:

- **Visit Open Charge Map:** Navigate to the Open Charge Map website at https://openchargemap.org/.
- **Create an Account:** If you don't have an account, sign up for free by clicking on the "Sign Up" button and filling out the required information.




- **Access Developer Dashboard:** Once logged in, access your account settings or navigate to the developer dashboard.




- **Register an application:** To get an API key you need to first register through this application first

ithout using your username and password.

here are no applications authorized to access your Open Charge Map account.

# MY API KEYS

you are a software developer you may need an API key to use the Open Charge Map API. Y

anage your API keys here.

REGISTER AN APPLICATION

| APP | WEBSITE | CREATED | ENABLED | PUBLIC | API KEY |
|---|---|---|---|---|---|

# CREATE NEW APPLICATION

Title

api key needed

Description

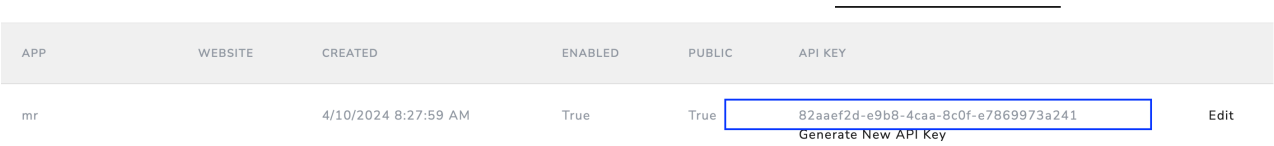
for my uni project

Website

List App in Public Showcase

SAVE

- **Generate API Key:** Look for the section related to API access or API keys. Generate a new API key if you don't have one already.




- **Copy API Key:** Once generated, copy the API key provided. This key will be used to authenticate your requests to the Open Charge Map API.

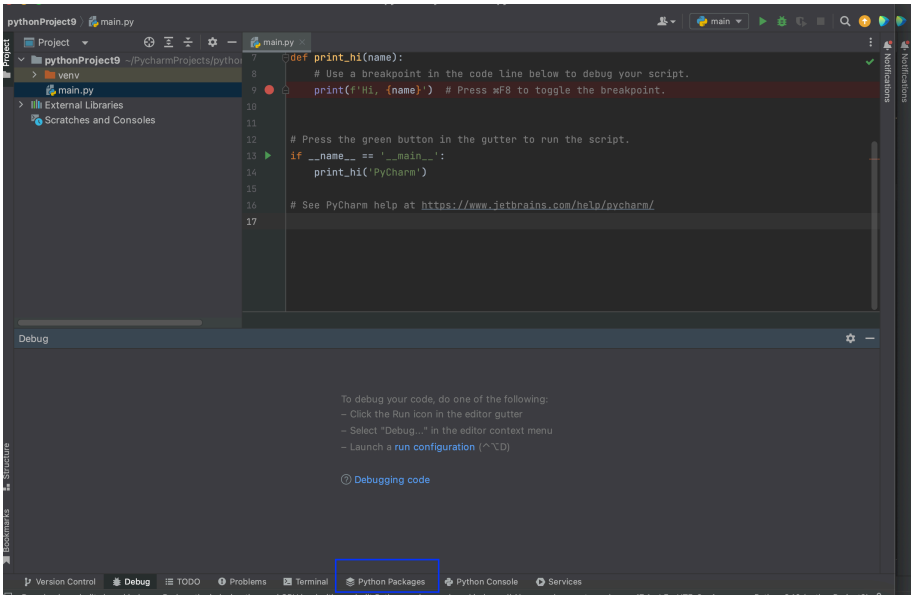
---

## Generalized Code to Get Data from the API:

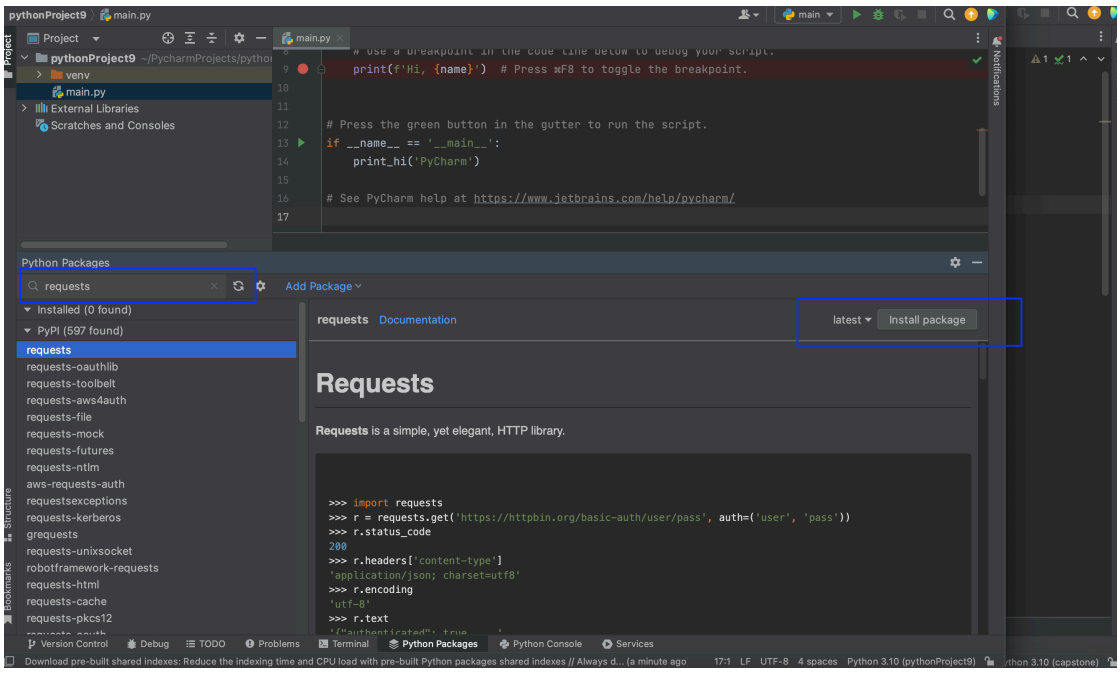
-First, we need to install the requests library using - pip install requests

```
ast login: Wed Apr 10 21:57:30 on ttys000
base) yuvrajsinghsekhon@yuvrajs-MacBook-Air ~ % pip install requests
equirement already satisfied: requests in /opt/anaconda3/lib/python3.11/site-packages (2.31.0)
equirement already satisfied: charset-normalizer<4,>=2 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2.0.4)
equirement already satisfied: idna<4,>=2.5 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (3.4)
equirement already satisfied: urllib3<3,>=1.21.1 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2.0.7)
equirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.11/site-packages (from requests) (2024.2.2)
```

- The next step is to add this library to the file we are writing code in
- I am using pycharm to write code so first make a new file and then go to the bottom and find python packages.



- Click on python packages search for the requests library and download it




- Now below is a generalised code, use the following Python code to retrieve data from the Open Charge Map API and process it:

```
import requests
```

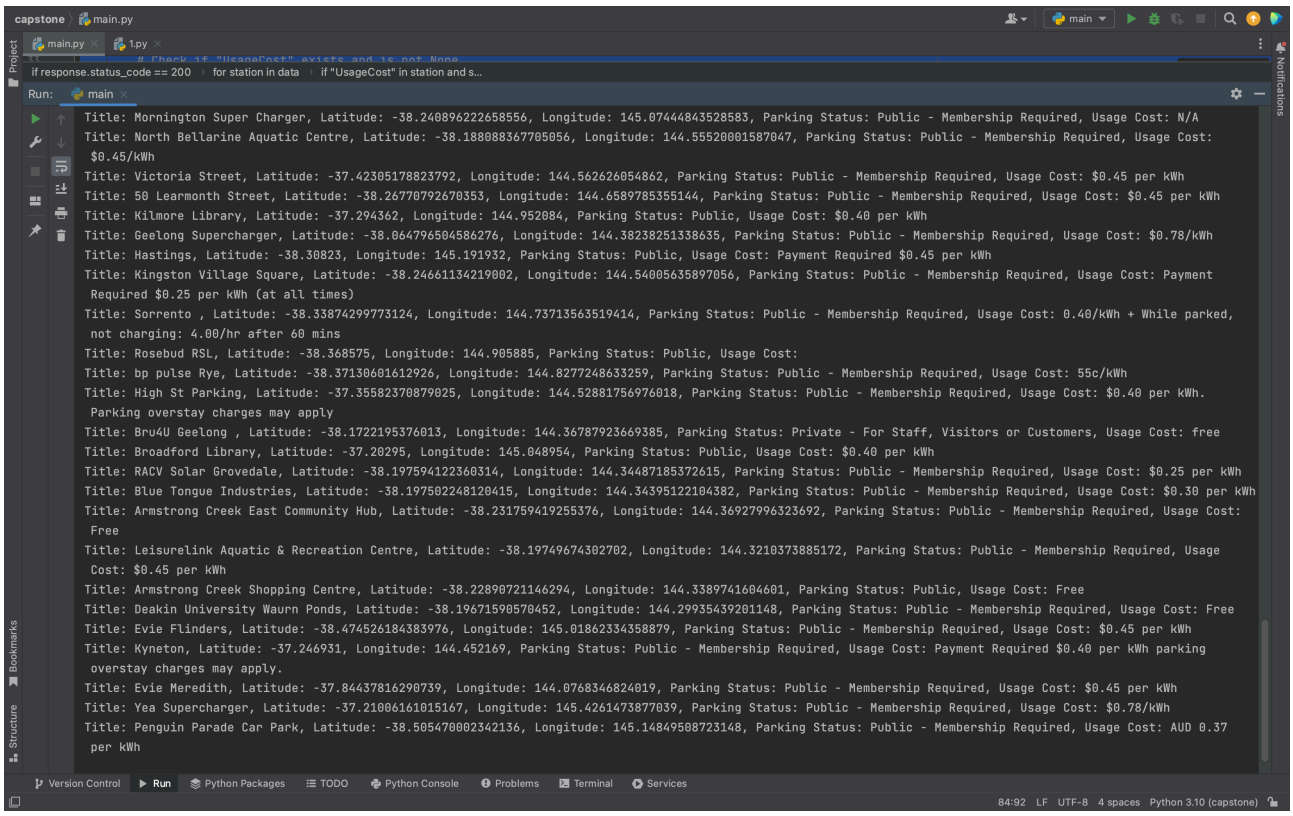
```
# Define the API endpoint and parameters
api_url = "https://api.openchargemap.io/v3/poi/"
params = {
    "output": "json",
    "countrycode": "AU",
    "maxresults": 500,  # Increase max results to ensure all stations are retrieved
    "latitude": -37.8136,  # Latitude of Melbourne
    "longitude": 144.9631,  # Longitude of Melbourne
```

```
    "distance": 50,  # Search radius in kilometers
    "key": "82aaef2d-e9b8-4caa-8c0f-e7869973a241"  # Get your
API key from Open Charge Map
}

# Make the API request
response = requests.get(api_url, params=params)

# Check if the request was successful
if response.status_code == 200:
    data = response.json()
    # Process the data as needed
    for station in data:
        title = station["AddressInfo"]["Title"]
        latitude = station["AddressInfo"]["Latitude"]
        longitude = station["AddressInfo"]["Longitude"]

        # Check if "UsageType" exists and is not None
        if "UsageType" in station and station["UsageType"] is
not None:
            parking_status = station["UsageType"]["Title"]
        else:
            parking_status = "N/A"

        # Check if "UsageCost" exists and is not None
        if "UsageCost" in station and station["UsageCost"] is
not None:
            usage_cost = station["UsageCost"]
```

```
        else:
            usage_cost = "N/A"
```

```
        print(
            f"Title: {title}, Latitude: {latitude},
Longitude: {longitude}, Parking Status: {parking_status},
Usage Cost: {usage_cost}")
else:
    print("Failed to fetch data. Status code:",
response.status_code)
```

The result I got is like this-




Consequently, in order to convert this file to a CSV file, I changed my code to convert the data I am receiving directly to a CSV file.

Below is the code used to turn the data into a CSV file.

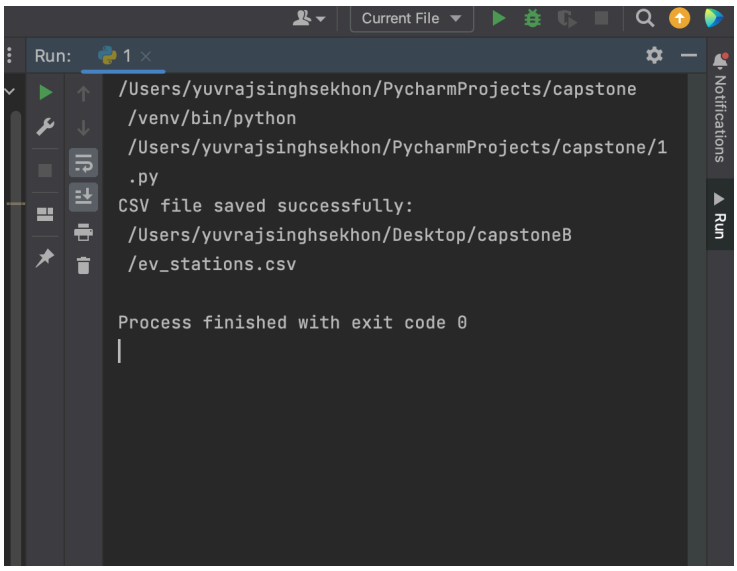
```python
import csv
import os
import requests



# Define the API endpoint and parameters
api_url = "https://api.openchargemap.io/v3/poi/"
params = {
    "output": "json",
    "countrycode": "AU",
    "maxresults": 500,  # Increase max results to ensure all stations are retrieved
    "latitude": -37.8136,  # Latitude of Melbourne
    "longitude": 144.9631,  # Longitude of Melbourne
    "distance": 75,  # Search radius in kilometers
    "key": "82aaef2d-e9b8-4caa-8c0f-e7869973a241"  # Get your API key from Open Charge Map
}

# Make the API request
response = requests.get(api_url, params=params)

# Check if the request was successful
if response.status_code == 200:
    data = response.json()
```

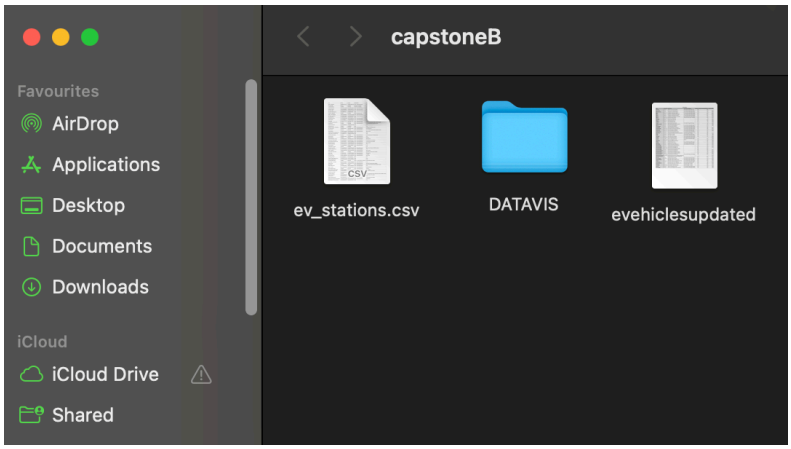
```
    # Specify CSV file path on the desktop
    desktop_path = os.path.join(os.path.expanduser("~/
Desktop"), "capstoneB")
    os.makedirs(desktop_path, exist_ok=True)  # Create the
folder if it doesn't exist
    csv_file_path = os.path.join(desktop_path,
"ev_stations.csv")

    try:
        # Open CSV file in write mode
        with open(csv_file_path, mode="w", newline="",
encoding="utf-8") as csv_file:
            # Create a CSV writer object
            csv_writer = csv.writer(csv_file)

            # Write header row
            csv_writer.writerow(["Title", "Latitude",
"Longitude", "Parking Status", "Usage Cost"])

            # Process the data and write rows to CSV
            for station in data:
                title = station.get("AddressInfo",
{}).get("Title", "N/A")
                latitude = station.get("AddressInfo",
{}).get("Latitude", "N/A")
                longitude = station.get("AddressInfo",
{}).get("Longitude", "N/A")
```

```
                # Check if "UsageType" exists and is not None
                if "UsageType" in station and
station["UsageType"] is not None:
                    parking_status = station["UsageType"]
["Title"]
                else:
                    parking_status = "Information not
available"

                # Check if "UsageCost" exists and is not None
                if "UsageCost" in station and
station["UsageCost"] is not None:
                    usage_cost = station["UsageCost"]
                else:
                    usage_cost = "Cost information not
available"

                # Write row to CSV
                csv_writer.writerow([title, latitude,
longitude, parking_status, usage_cost])

        print(f"CSV file saved successfully:
{csv_file_path}")

    except Exception as e:
        print("Error occurred while creating the CSV file:",
e)

else:
```

```
    print("Failed to fetch data. Status code:",
response.status_code)
```

I successfully converted all of the data to a CSV file in my desired directory, as evidenced by the output I saw on the console.




With the provided steps and customizable code, you can seamlessly extract comprehensive information about EV stations using the Open Charge Map API. Whether you need data on charging costs, parking availability, or any other details, this solution offers flexibility and efficiency. Feel empowered to tailor the code to your specific requirements, ensuring that you can gather the precise insights needed for your projects.

Happy data extraction!!!