🔑 APIs with Keys: Unlocking NASA Data 🚀

Hack for L.A. Data Collection Tutorial

API (With Key) Basics

- 🔑 Some APIs need a key
- 🤫 Keep it secret, keep it safe
- 📉 Limits on requests
- 💾 Cache results when possible

```
import os, requests

load_dotenv()  # comment this out if you're not using a .env file
API_KEY = os.getenv("NASA_API_KEY") # make sure so save your key in your env
url = f"https://api.nasa.gov/neo/rest/v1/neo/browse?api_key={API_KEY}"
response = requests.get(url)
```

Learning Goals

- ✅ Call a public API that requires a key
  - 🔗 https://api.nasa.gov/
    - A simple form — enter your name and email, and they'll send you an API key almost instantly. 🚀
- 🔒 Pass the key securely (header or query)
- ⚙️ Use environment variables for credentials
- 📊 Convert JSON responses into a DataFrame

Request → Response Mental Model

- 🌐 Endpoint: the URL you call
- 📡 Method: often GET for data
- 📋 Headers: where API key goes
- ❓ Query parameters: filters after ?
- 📬 Status code: success (200) or error
- 📦 Body: data returned, usually JSON

```
print(response.status_code)  # ✅ 200 means success

data = response.json()
print(data.keys())  # 📦 see what comes back
```

API Vocabulary (NASA Example)

- 🔑 API Key: secret token to access data
- 🛰️ Endpoint: `https://api.nasa.gov/neo/rest/v1/neo/browse`
- ⚙️ Query Parameters: `?api_key=YOUR_KEY&size=5`
- 📂 JSON: format of returned data
- ☄️ NEO: Near Earth Object (asteroid/comet)
- ✨ Absolute Magnitude (H): brightness → size estimate

```
neos = data["near_earth_objects"]
first = neos[0]

print(first["name"])          # ☄️ Asteroid name
print(first["absolute_magnitude_h"])  # ✨ Brightness
```

📝 Complete Example Code

```
import os, requests, pandas as pd

# 🔑 Load your API key
API_KEY = os.getenv("NASA_API_KEY")

# 🌐 Endpoint with query
url = f"https://api.nasa.gov/neo/rest/v1/neo/browse?api_key={API_KEY}"
response = requests.get(url)

# ✅ Check response and parse JSON
if response.status_code == 200:
    data = response.json()
    neos = data["near_earth_objects"]

    # 📊 Convert to DataFrame
    df = pd.json_normalize(neos)
    print(df.head())
else:
    print("Error:", response.status_code)
```