What's next?

What's next?

- APIs and queries: access OpenData programmatically
- Big data
- Databases
- Web scraping
- Automation
- Data dashboards
- Causal inference

APIs and queries

APIs: interfaces for getting data from online sources

APIs and queries

requests

- Tool for running http requests to get data from (or send data to) online resources
- https://realpython.com/python-requests/

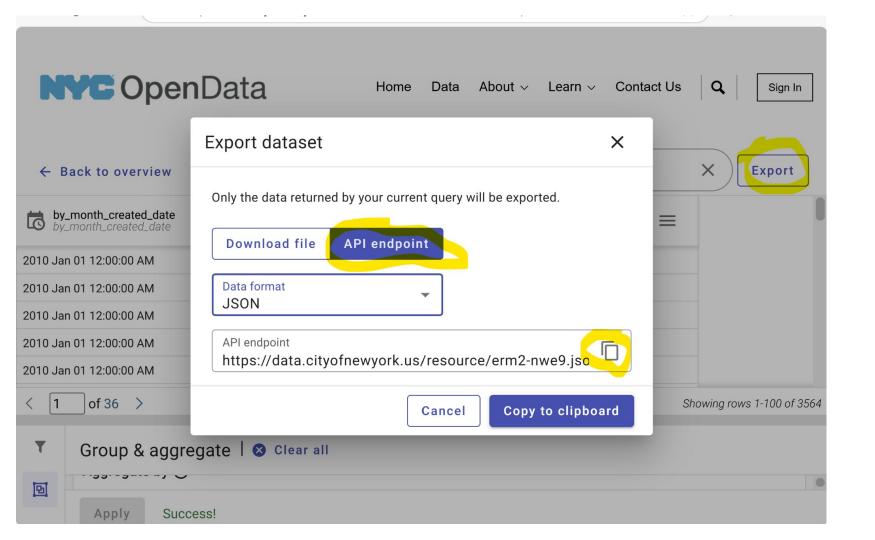
Databases

- SQL
 - Standard query language for getting data from databases
 - Intro tutorial: https://sqlbolt.com

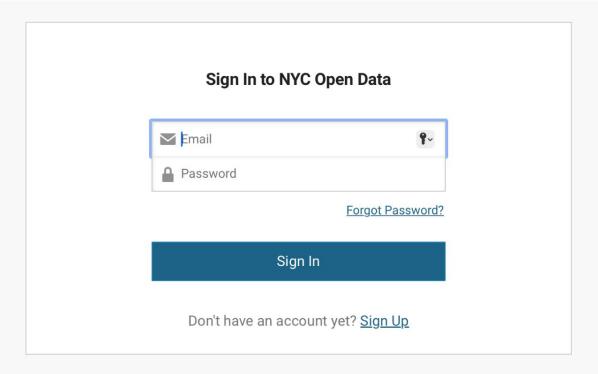
SQL

```
SELECT
    `station_complex_id`,
    `station_complex`,
   `payment_method`,
    sum(`ridership`) AS `sum_ridership`
WHERE
    `transit_timestamp`
    BETWEEN "2023-01-01T00:00:00"
    AND "2024-12-31T23:59:59"
GROUP BY
    `station_complex_id`,
    `station_complex`,
    `payment_method`
```

NYC OpenData requests











Home Data About v Learn v Contact Us Q Dan

Profile

Account Settings



Developer Settings

API Keys

► What are API Keys? ②

Search Q Name or ID Include Deleted Search

Api Key ID

Created At

↓ Actions

No Results

Api Key Name

Previous Next

App Tokens

► What are App Tokens? ②

Create New App Token

Create new API Key

Name

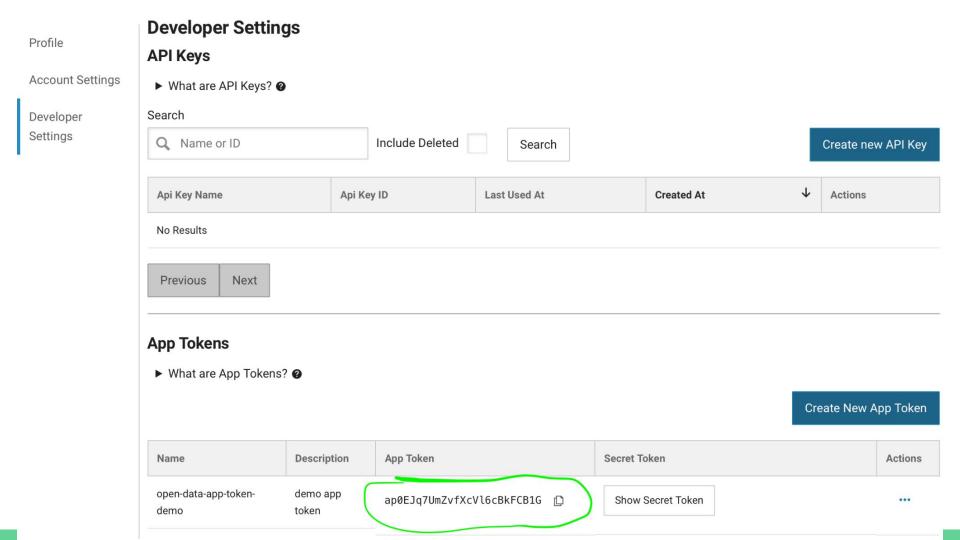
Description

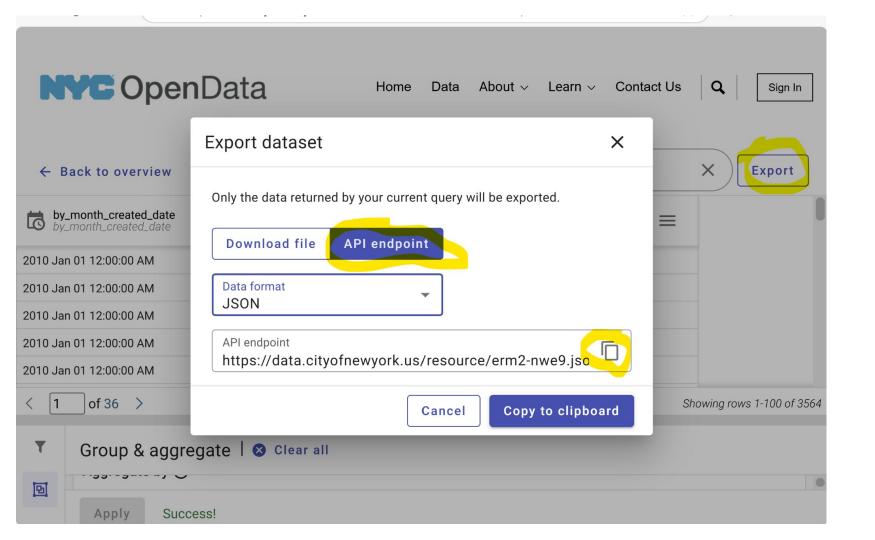
App Token

Last Used At

Secret Token

Actions





0A%20%20date_trunc_ym(%60created_date%60)%20AS%20%60by_month_created _date%60%2C%0A%20%20%20%60complaint_type%60%2C%0A%20%20count(DISTINCT%2 0%60unique_key%60)%20AS%20%60count_distinct_unique_key%60%0AWHERE%20 %60agency%60%20%3D%3D%20%22DSNY%22%0AGROUP%20BY%20date_trunc_ym(%60c reated_date%60)%2C%20%60complaint_type%60

https://data.cityofnewyork.us/resource/erm2-nwe9.json?\$query=SELECT%

```
https://data.cityofnewyork.us/resource/erm2-nwe9.json?$query=
SELECT
 date_trunc_ym(`created_date`) AS `by_month_created_date`,
```

`complaint_type`,

count(DISTINCT `unique_key`) AS `count_distinct_unique_key` WHERE `agency` == "DSNY"

GROUP BY date_trunc_ym(`created_date`), `complaint_type`

```
https://data.cityofnewyork.us/resource/erm2-nwe9.json?$query=
SELECT
  date_trunc_ym(`created_date`) AS `by_month_created_date`,
  `complaint_type`,
  count(DISTINCT `unique_key`) AS `count_distinct_unique_key`
WHERE `agency` == "DSNY"
GROUP BY date_trunc_ym(`created_date`), `complaint_type`
LIMIT 100000000
```

https://data.cityofnewyork.us/resource/erm2-nwe9.json?\$query=SELECT% 0A%20%20date_trunc_ym(%60created_date%60)%20AS%20%60by_month_created _date%60%2C%0A%20%20%060complaint_type%60%2C%0A%20%20count(DISTINCT%2 0%60unique_key%60)%20AS%20%60count_distinct_unique_key%60%0AWHERE%20%60agency%60%20%3D%3D%20%22DSNY%22%0AGROUP%20BY%20date_trunc_ym(%60created_date%60)%2C%20%60complaint_type%60%20LIMIT%201000000000%\$\$app token=xxxxxxxxxxxxxxx

```
OPEN_DATA_APP_TOKEN = 'xxx'
BASE_URL = 'https://data.cityofnewyork.us/'
table_id = 'erm2-nwe9'
query = """
    SELECT
        DATE_TRUNC_YMD(`created_date`) as date,
        COUNT(`unique_key`) as count
    WHERE
        (`complaint_type` LIKE "Homeless Person Assistance")
        AND (`created_date` >= "2019-01-01")
    GROUP BY
        DATE_TRUNC_YMD(`created_date`)
    LIMIT 100000
    1111111
data_request_url = (
    BASE_URL
    + 'resource/'
    + table_id
    +'.json?'
    + 'squery='
    + urllib.parse.quote(query)
    + '&$$app_token='
    + OPEN_DATA_APP_TOKEN
data = pd.read_json(data_request_url)
```

Big data

- dask
 - pandas syntax on big (including bigger-than-memory) datasets
 - https://www.dask.org
- DuckDB
 - Builds standalone database to run queries with SQL in Python
 - Handles bigger-than-memory data and loads and gueries fast
 - https://duckdb.org/docs/stable/clients/python/overview.html
 - https://realpython.com/python-duckdb/
- polars
 - Alternative dataframe syntax with fast execution on big data
 - https://docs.pola.rs/#key-features

Web scraping

- BeautifulSoup
 - For parsing web pages and extracting links, text, or data tables
 - https://beautiful-soup-4.readthedocs.io/en/latest/
 - https://realpython.com/beautiful-soup-web-scraper-python/

Automation

- Github Actions
 - For running tasks on a schedule
 - e.g. saving data from real-time Citi Bike stations feed:
 https://github.com/NYCComptroller/citi-bike-gbfs
 - https://palewi.re/docs/first-github-scraper/action.html
 - https://githubnext.com/projects/flat-data

Data dashboards

- Streamlit
 - https://streamlit.io/#install
- Dash
 - https://dash.plotly.com/tutorial
 - https://realpython.com/python-dash/
- Observable
 - https://observablehq.com/framework/

Causal inference

- Using statistics to determine if something caused something else
 - Causal Inference for the Brave and True
 https://matheusfacure.github.io/python-causality-handbook/

Where to learn

- Kaggle
 - https://www.kaggle.com/
- Real Python
 - https://realpython.com/
- Think Python
 - https://allendowney.github.io/ThinkPython
- BetaNYC
 - https://www.beta.nyc/featured-tools/nyc-opendata-classes/