Data Science Team API v2.1 Tutorial

This document explains how to use the API of City of Melbourne Open Data. At the same time, to make it more convenient to replace API data and API keys. The team decided to provide reference code and instructions.

Please note that this code and instructions only apply to API v2.1 and the City of Melbourne Open Data

Note

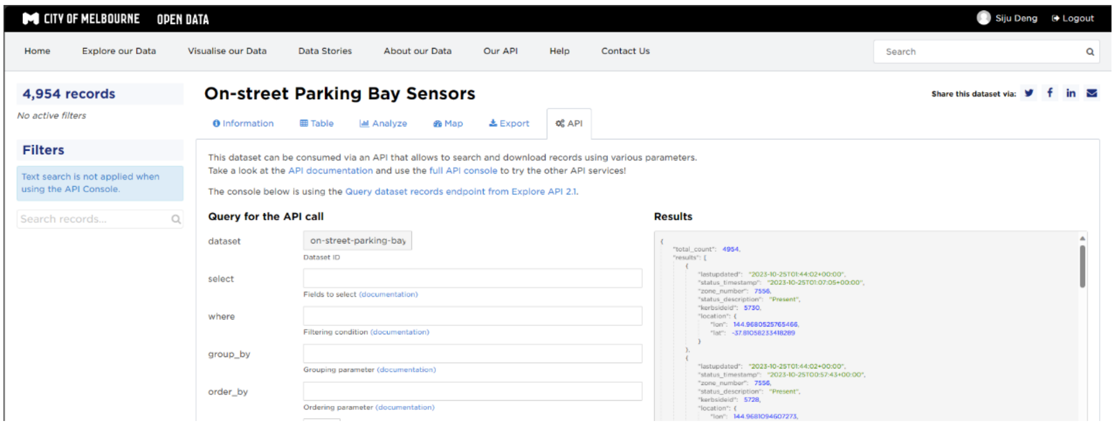
Each organization's API settings are different.

Some parts of the documentation, such as loop acquisition. The government may ban it in subsequent updates, and it does not apply to team' API keys with advanced permissions. (For commercial enterprises, loop acquisition is likely to be disabled and may be defaulted by the system as an attack or illegal request)

Get URL

Let us take the API of On-street Parking Bay Sensors as an example.

The data set API page.



4,954 records

No active filters

Filters

Text search is not applied when using the API Console.

On-street Parking Bay Sensors

This dataset can be consumed via an API that allows to search and download records using various parameters.
Take a look at the API documentation and use the full API console to try the other API services!

The console below is using the Query dataset records endpoint from Explore API 2.1.

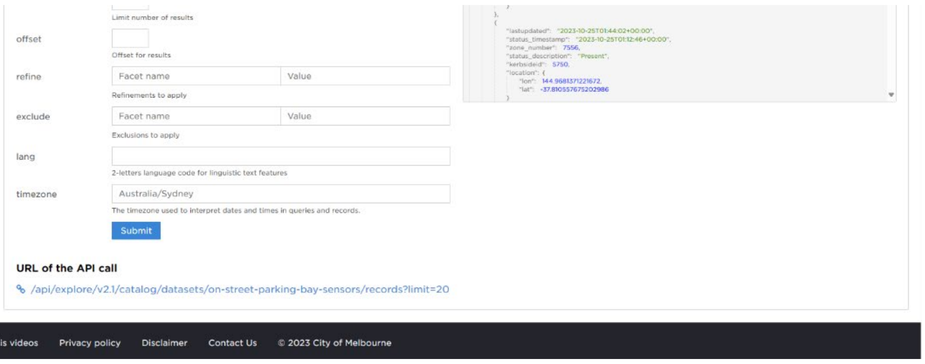
Query for the API call

dataset	on-street-parking-bay Dataset ID
select	Fields to select (documentation)
where	Filtering condition (documentation)
group_by	Grouping parameter (documentation)
order_by	Ordering parameter (documentation)

Results

```
{
    "total_count":  4954,
    "results": [
        {
            "lastupdated":  "2023-10-25T01:44:02+00:00",
            "status_timestamp":  "2023-10-25T01:07:05+00:00",
            "zone_number":  7556,
            "status_description":  "Present",
            "kerbsideid":  5730,
            "location": {
                "lon":  144.9680525765466,
                "lat":  -37.81058233418289
            }
        },
        {
            "lastupdated":  "2023-10-25T01:44:02+00:00",
            "status_timestamp":  "2023-10-25T00:57:43+00:00",
            "zone_number":  7556,
            "status_description":  "Present",
            "kerbsideid":  5728,
            "location": {
                "lon":  144.9681094607273,
```

Then, at the bottom, we can see an API connection.



	Limit number of results
offset	Offset for results
refine	Facet name / Value Refinements to apply
exclude	Facet name / Value Exclusions to apply
lang	2-letters language code for linguistic text features
timezone	Australia/Sydney The timezone used to interpret dates and times in queries and records.

```
            }
        },
        {
            "lastupdated":  "2023-10-25T01:44:02+00:00",
            "status_timestamp":  "2023-10-25T01:12:46+00:00",
            "zone_number":  7556,
            "status_description":  "Present",
            "kerbsideid":  5750,
            "location": {
                "lon":  144.9681371221672,
                "lat":  -37.810557675202986
            }
```

Submit

URL of the API call

/api/explore/v2.1/catalog/datasets/on-street-parking-bay-sensors/records?limit=20

Data Policy   Visualisation and analysis videos   Privacy policy   Disclaimer   Contact Us   © 2023 City of Melbourne

URL description

The URL is https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/on-street-parking-bay-sensors/records?limit=20

https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/ is basic.

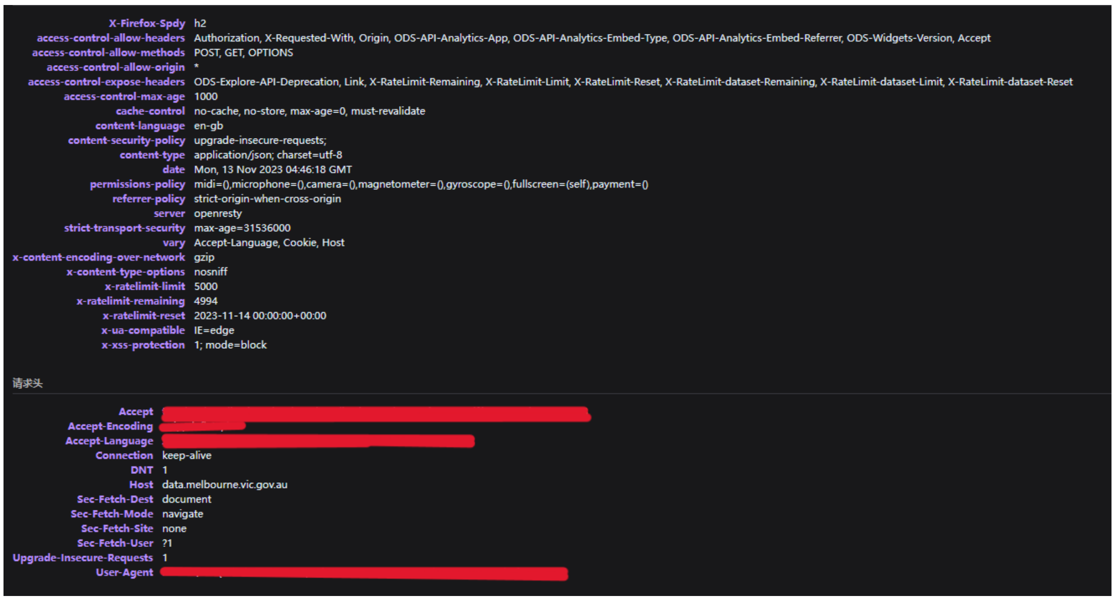
datasets means that we are using data sets. If using csv or others, changes here.

on-street-parking-bay-sensors represents the name of the data set. This name can be found in the API connection of each data set.

limit=20 is the amount of data requested.

URL Limit

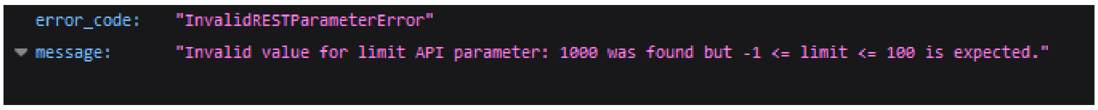
Let's open this URL with Firefox. Firefox will provide an overview of the data,



```
                  X-Firefox-Spdy  h2
     access-control-allow-headers  Authorization, X-Requested-With, Origin, ODS-API-Analytics-App, ODS-API-Analytics-Embed-Type, ODS-API-Analytics-Embed-Referrer, ODS-Widgets-Version, Accept
     access-control-allow-methods  POST, GET, OPTIONS
      access-control-allow-origin  *
    access-control-expose-headers  ODS-Explore-API-Deprecation, Link, X-RateLimit-Remaining, X-RateLimit-Limit, X-RateLimit-Reset, X-RateLimit-dataset-Remaining, X-RateLimit-dataset-Limit, X-RateLimit-dataset-Reset
          access-control-max-age  1000
                   cache-control  no-cache, no-store, max-age=0, must-revalidate
                content-language  en-gb
          content-security-policy  upgrade-insecure-requests;
                    content-type  application/json; charset=utf-8
                            date  Mon, 13 Nov 2023 04:46:18 GMT
              permissions-policy  midi=(),microphone=(),camera=(),magnetometer=(),gyroscope=(),fullscreen=(self),payment=()
                  referrer-policy  strict-origin-when-cross-origin
                          server  openresty
        strict-transport-security  max-age=31536000
                            vary  Accept-Language, Cookie, Host
x-content-encoding-over-network  gzip
           x-content-type-options  nosniff
               x-ratelimit-limit  5000
           x-ratelimit-remaining  4994
               x-ratelimit-reset  2023-11-14 00:00:00+00:00
              x-ua-compatible  IE=edge
              x-xss-protection  1; mode=block

请求头

                  Accept  [redacted]
         Accept-Encoding  [redacted]
         Accept-Language  [redacted]
              Connection  keep-alive
                     DNT  1
                    Host  data.melbourne.vic.gov.au
          Sec-Fetch-Dest  document
          Sec-Fetch-Mode  navigate
          Sec-Fetch-Site  none
          Sec-Fetch-User  ?1
Upgrade-Insecure-Requests  1
              User-Agent  [redacted]
```

Let's change the number of requests to 1000.

You can see that our maximum number of requests is 100.



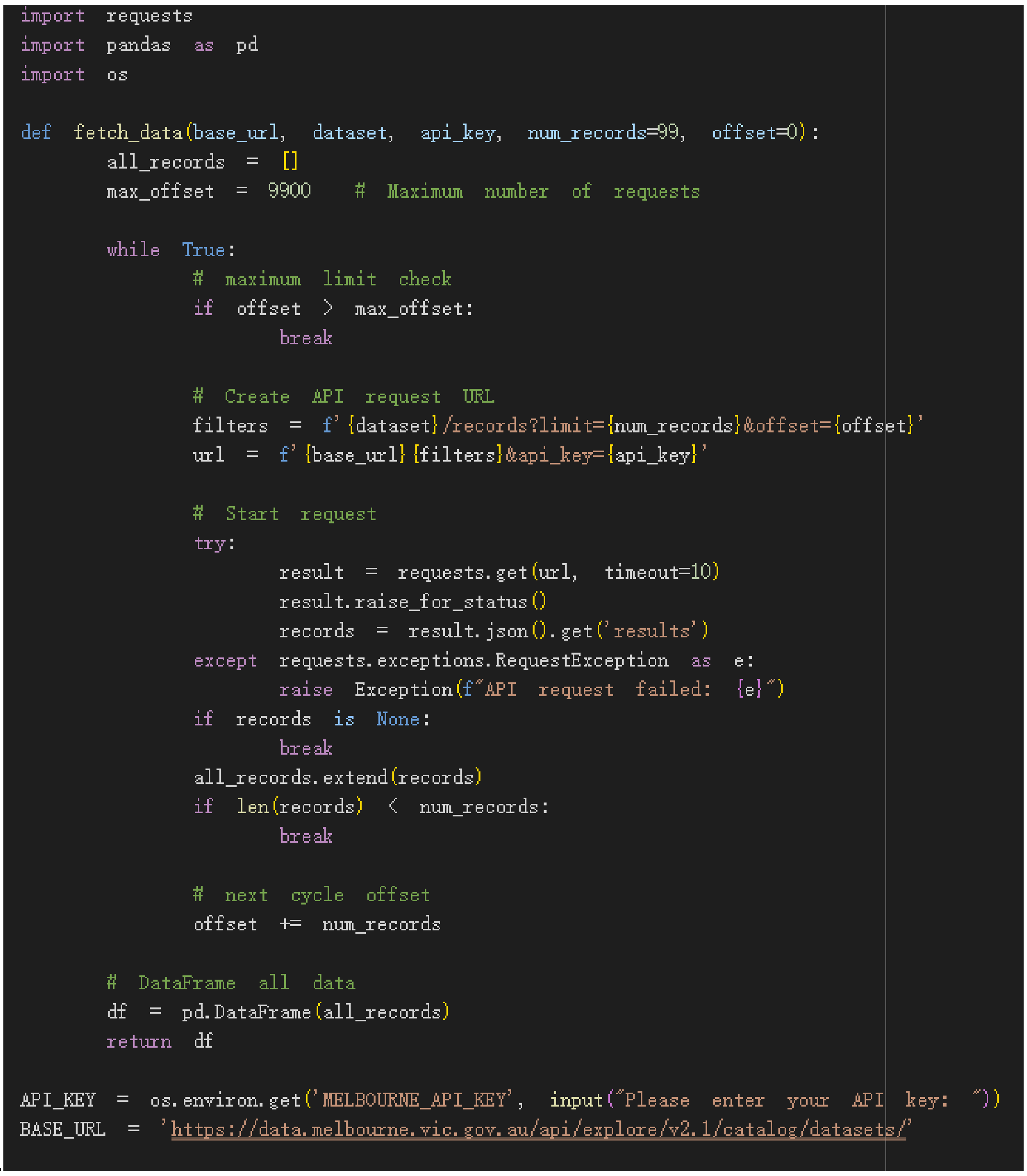
```
  error_code:    "InvalidRESTParameterError"
▼ message:       "Invalid value for limit API parameter: 1000 was found but -1 <= limit <= 100 is expected."
```

So, we need to loop the request until we obtain all the data collection data.

URL Code example

Use Datasets

We now know the base URL, data name and type we are getting. We set a function.

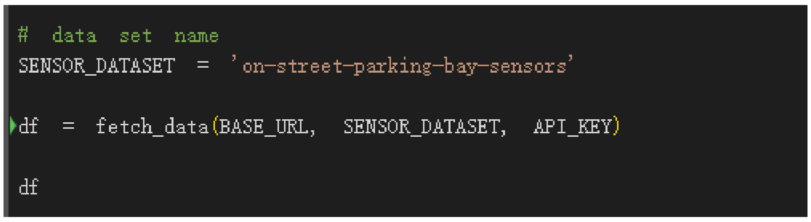


```python
import requests
import pandas as pd
import os

def fetch_data(base_url, dataset, api_key, num_records=99, offset=0):
        all_records = []
        max_offset = 9900   # Maximum number of requests

        while True:
                # maximum limit check
                if offset > max_offset:
                        break

                # Create API request URL
                filters = f'{dataset}/records?limit={num_records}&offset={offset}'
                url = f'{base_url}{filters}&api_key={api_key}'

                # Start request
                try:
                        result = requests.get(url, timeout=10)
                        result.raise_for_status()
                        records = result.json().get('results')
                except requests.exceptions.RequestException as e:
                        raise Exception(f"API request failed: {e}")
                if records is None:
                        break
                all_records.extend(records)
                if len(records) < num_records:
                        break

                # next cycle offset
                offset += num_records

        # DataFrame all data
        df = pd.DataFrame(all_records)
        return df

API_KEY = os.environ.get('MELBOURNE_API_KEY', input("Please enter your API key: "))
BASE_URL = 'https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/'
```

.

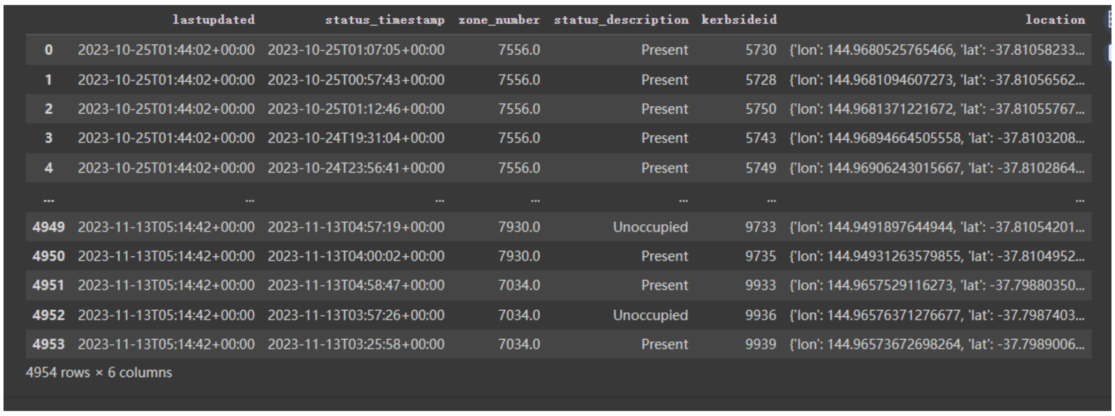
If you encounter request failure, you can try adding a waiting time to avoid non-interval requests.

Next, we start selecting the data set and obtaining the data.



```python
# data set name
SENSOR_DATASET = 'on-street-parking-bay-sensors'

df = fetch_data(BASE_URL, SENSOR_DATASET, API_KEY)

df
```

Success



	lastupdated	status_timestamp	zone_number	status_description	kerbsideid	location
0	2023-10-25T01:44:02+00:00	2023-10-25T01:07:05+00:00	7556.0	Present	5730	{'lon': 144.9680525765466, 'lat': -37.81058233...
1	2023-10-25T01:44:02+00:00	2023-10-25T00:57:43+00:00	7556.0	Present	5728	{'lon': 144.9681094607273, 'lat': -37.81056562...
2	2023-10-25T01:44:02+00:00	2023-10-25T01:12:46+00:00	7556.0	Present	5750	{'lon': 144.9681371221672, 'lat': -37.81055767...
3	2023-10-25T01:44:02+00:00	2023-10-24T19:31:04+00:00	7556.0	Present	5743	{'lon': 144.96894664505558, 'lat': -37.8103208...
4	2023-10-25T01:44:02+00:00	2023-10-24T23:56:41+00:00	7556.0	Present	5749	{'lon': 144.96906243015667, 'lat': -37.8102864...
...	...	...	...	...	...	...
4949	2023-11-13T05:14:42+00:00	2023-11-13T04:57:19+00:00	7930.0	Unoccupied	9733	{'lon': 144.9491897644944, 'lat': -37.81054201...
4950	2023-11-13T05:14:42+00:00	2023-11-13T04:00:02+00:00	7930.0	Present	9735	{'lon': 144.94931263579855, 'lat': -37.8104952...
4951	2023-11-13T05:14:42+00:00	2023-11-13T04:58:47+00:00	7034.0	Present	9933	{'lon': 144.9657529116273, 'lat': -37.79880350...
4952	2023-11-13T05:14:42+00:00	2023-11-13T03:57:26+00:00	7034.0	Unoccupied	9936	{'lon': 144.96576371276677, 'lat': -37.7987403...
4953	2023-11-13T05:14:42+00:00	2023-11-13T03:25:58+00:00	7034.0	Present	9939	{'lon': 144.96573672698264, 'lat': -37.7989006...

4954 rows × 6 columns

Use CSV

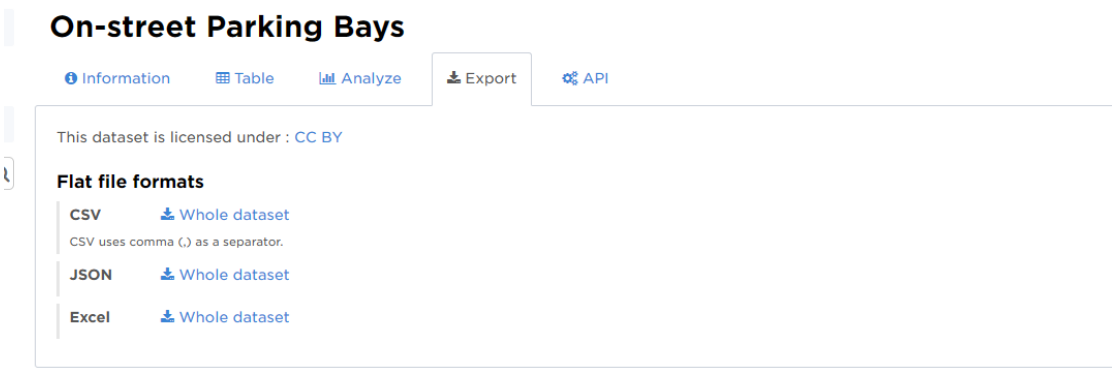
But please be aware. The number of APIs is limited. Only data before the restriction is allowed to be retrieved. If you have a lot of data. For example, on-street-parking-bays. Not all data will be retrieved.

```
9999 rows × 8 columns
```

You need to request the CSV file to obtain it now. We cannot use local files, but reading CSV files online is possible. Code is

There are two ways.

The first is to copy the download link directly from the download page.



On-street Parking Bays

This dataset is licensed under : CC BY

Flat file formats

CSV	Whole dataset
	CSV uses comma (,) as a separator.
JSON	Whole dataset
Excel	Whole dataset

Second, use the CSV request in the API description. The data type change to CSV, and a download limit is attached. Manually set quote_all and with_bom.

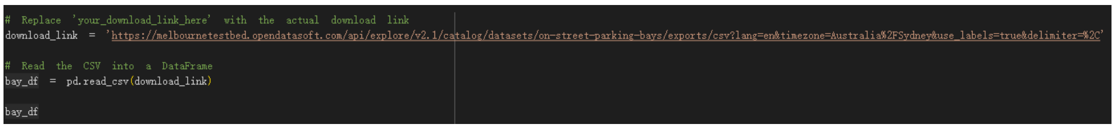
The suffix is:

csv?delimiter=%3B&list_separator=%2C&quote_all=false&with_bom=true

For on-street-parking-bays：

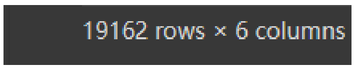
https://melbournetestbed.opendatasoft.com/api/explore/v2.1/catalog/datasets/on-street-parking-bays/exports/csv?delimiter=%3B&list_separator=%2C&quote_all=false&with_bom=true

The code on notebook to read csv link is:



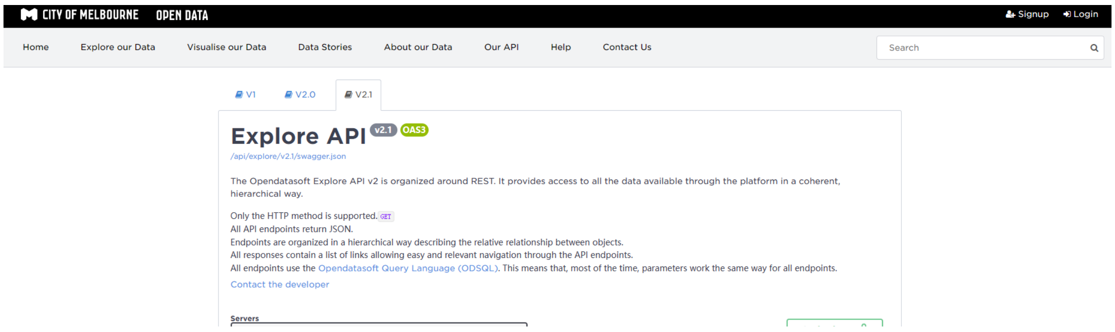
```
#  Replace  'your_download_link_here'  with  the  actual  download  link
download_link  =  'https://melbournetestbed.opendatasoft.com/api/explore/v2.1/catalog/datasets/on-street-parking-bays/exports/csv?lang=en&timezone=Australia%2FSydney&use_labels=true&delimiter=%2C'

#  Read  the  CSV  into  a  DataFrame
bay_df  =  pd.read_csv(download_link)

bay_df
```

Both methods can break through the limit.



```
19162 rows × 6 columns
```

For detailed API usage, please refer to the official API description. Please click on “our API”.



Explore API

/api/explore/v2.1/swagger.json

The Opendatasoft Explore API v2 is organized around REST. It provides access to all the data available through the platform in a coherent, hierarchical way.

Only the HTTP method is supported. GET
All API endpoints return JSON.
Endpoints are organized in a hierarchical way describing the relative relationship between objects.
All responses contain a list of links allowing easy and relevant navigation through the API endpoints.
All endpoints use the Opendatasoft Query Language (ODSQL). This means that, most of the time, parameters work the same way for all endpoints.
Contact the developer

Use GeoJSON

Use GeoJSON is same with CSV. You can still copy download link or write restrictions in URL by hand. All download files use same way.

Code in notebook change to:

```
# API URL of the GeoJSON file
ped_geojson_url = "https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/pedestrian-network/exports/geojson?lang=en&timezone=Australia%2FSydney"
# Read
ped_gdf = gpd.read_file(ped_geojson_url)
```

Author

Siju Deng