Programmatic Data Access

in the

Commonwealth of Virginia DataSAGE Environment

Prepared by

Qlarion

Last Updated 9/30/2020

Contents

Governed Data Access via the Socrata Data Sharing Platform

As approved and documented in the Commonwealth of Virginia Data Trust, the Socrata platform provides a single access point for data resources shared through the Office of the Chief Data Officer for the Commonwealth of Virginia and the DataSAGE environment. Users approved to access datasets on the Socrata platform may use the Socrata REST API to retrieve shared data. This document outlines the process for programmatic retrieval of data using Socrata's REST API.

Authentication

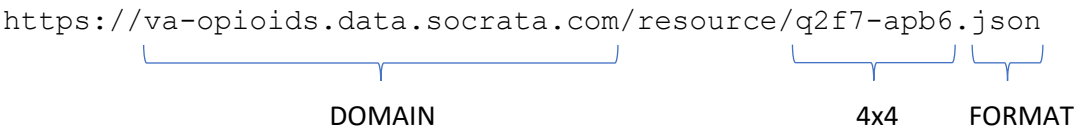
The Socrata platform requires users to making requests for data to use an API Key to authenticate their requests. The API Key includes a Key ID and a Key Secret, which function as proxies for login credentials to the Socrata platform. Example API requests shown in this document use `**API KEY ID**` and `**API KEY SECRET**` in place of the values of an actual API Key.

Users with login credentials to the Socrata platform can create an API Key in the developer settings found on the profile page for their user account on Socrata. For assistance with API Key creation, contact the Platform Administrator at va-faact.info@qlarion.com. Users without login credentials to the Socrata platform should contact the Platform Administrator to obtain an API Key.

API Keys provide access to approved users only, per the terms of the Commonwealth of Virginia Data Trust. As such, users should **NEVER** share API Key values with persons not explicitly authorized to make data requests under the approved data use case associated with a given API Key.

Socrata Dataset API Endpoint Convention

The Socrata platform automatically creates an API endpoint for every dataset on the platform. API endpoints provided by Socrata adhere to a standard convention as represented in the sample API endpoint provided below:



The DOMAIN refers to the Socrata platform site that hosts the shared data.

The 4x4 (pronounced “four-by-four”) uniquely identifies the dataset on the Socrata platform.

The FORMAT specifies the way in which the API will return the result set for the request. The API can return data as either JSON or CSV.

Users with login credentials to the Socrata platform can find the API endpoint for a dataset on the Primer page for that dataset. For assistance with locating the Primer page for a dataset, contact the Platform Administrator at va-faact.info@qlarion.com. Users without login credentials to the Socrata platform should contact the Platform Administrator to obtain the API endpoint for the dataset.

In addition, both users with login credentials to the Socrata platform and users without login credentials to the Socrata platform may optionally use the Socrata Discovery API to lookup API endpoints accessible to them. This document includes an explanation of the procedure for using the Socrata Discovery API in the **Request Library Method** section.

Making Requests from Socrata API Endpoint Using Python

This section demonstrates the procedure for using Python to make requests from a Socrata API endpoint. This demonstration uses a non-sensitive dataset containing Virginia labor force data; however, the same principles and processes apply to any dataset on Socrata. This includes filtered dataset views and SoQL derived views of datasets.

The dataset for this demonstration contains monthly employment and unemployment estimates reported from the Local Area Unemployment Statistics (LAUS) program between January 1990 and July 2020 for Virginia counties and cities. For the original dataset see Virginia Employment Commission. The dataset for this demonstration has the following API endpoint:

`https://va-opioids.data.socrata.com/resource/q2f7-apb6.json`

Python can request data via the API endpoint using either the request or sodapy libraries.

Request Library Method

To request the data using the `request` library, first import the necessary packages.

```
In [1]:
    import requests
    import json
    import pandas
```

Next, make the API request of the data using `request.get(...)` as shown below. The first argument specifies the API endpoint for the dataset. The second argument, `auth`, is a tuple containing the user's authentication, i.e., the API Key ID and Key Secret.

```
In [2]:
    response = requests.get(
        'https://va-opioids.data.socrata.com/resource/q2f7-apb6.json?$limit
    =50000',
        auth=('**API KEY ID**','**API KEY SECRET**'))
```

The API endpoint as shown in this example includes the optional `$limit` parameter which controls the total number of rows returned in the response. The number of records returned in a query response defaults to 1,000 records per request. For datasets with more records than the limit, the offset parameter allows successive requests of the same API endpoint to page through the set of records in the dataset. To learn more about using the `$limit` and `$offset` parameters to page through data in an API response review this Socrata Developer's documentation. The `$limit` and `$offset` parameters

are just a few of the query parameters that can control the contents of the response to the API request. To learn more about queries and parameters in Socrata's API, visit Socrata's API Developers Page. For more information on properties/methods associated with `response` object see w3school's python developer documentation.

Next, load the text of the response object as a JSON using the `loads` method from the `json` package.

```
In [3]:
      list_response = json.loads(response.text)
```

Observe that the first two elements in `list_response` contain a list of dictionaries containing column and value pairs for each record.

```
In [4]:
      print(list_response[0:2])
```

```
      [{'date': '1990-01-01T00:00:00.000', 'fips': '51001', 'statename': 'Vir
      ginia', 'areaname': 'Accomack County', 'laborforce': '14494', 'emplab':
      '13293', 'unemp': '1201', 'unemprate': '0.08286187387884641'}, {'date':
      '1990-02-01T00:00:00.000', 'fips': '51001', 'statename': 'Virginia', 'a
      reaname': 'Accomack County', 'laborforce': '14415', 'emplab': '13244',
      'unemp': '1171', 'unemprate': '0.08123482483524107'}]
```

The `pandas` package can read dictionaries of any representation as a two-dimensional dataframe using the `DataFrame` method. To accomplish this, pass the list of dictionaries into the `DataFrame` method.

```
In [5]:
      df = pandas.DataFrame(list_response)
      df
```

Out[5]:

	date	fips	statename	areaname	laborforce	emplab	unemp	unemprate
0	1990-01-01T00:00:00.000	51001	Virginia	Accomack County	14494	13293	1201	0.08286187387884641
1	1990-02-01T00:00:00.000	51001	Virginia	Accomack County	14415	13244	1171	0.08123482483524107
2	1990-03-01T00:00:00.000	51001	Virginia	Accomack County	14624	13638	986	0.0674234135667396
3	1990-04-01T00:00:00.000	51001	Virginia	Accomack County	14803	14006	797	0.05384043774910491
4	1990-05-01T00:00:00.000	51001	Virginia	Accomack County	15215	14542	673	0.04423266513309234
...	...	...	...	...	...	...	...	...
49106	2020-03-01T00:00:00.000	51840	Virginia	Winchester city	14923	14432	491	0.03290223145480131
49107	2020-04-01T00:00:00.000	51840	Virginia	Winchester city	14582	12944	1638	0.11233027019613222
49108	2020-05-01T00:00:00.000	51840	Virginia	Winchester city	14503	13182	1321	0.09108460318554781
49109	2020-06-01T00:00:00.000	51840	Virginia	Winchester city	14433	13270	1163	0.08057922815769418
49110	2020-07-01T00:00:00.000	51840	Virginia	Winchester city	14454	13364	1090	0.0754116507541165

49111 rows × 8 columns

Users can find the number of rows in a dataset by making the following request. This may prove useful when defining `limit` parameter. The first argument specifies the API endpoint for the dataset. The second argument, `auth`, is a tuple containing the user's authentication, i.e., the API Key ID and Key Secret.

```
In [6]:
    row_count = requests.get(
        'https://va-opioids.data.socrata.com/resource/gta7-tkpk.json?$selec
    t=row_count&dataset_id=q2f7-apb6',
        auth=('**API KEY ID**','**API KEY SECRET**'))

    print(row_count.text)


    [{"row_count":"49111"}]
```

Users with login credentials to the Socrata platform should verify data types by looking at the dataset metadata on the Primer Page. Users with or without login credentials to the Socrata platform can programmatically access the metadata by making the following request using the Socrata Discovery API. The first argument uses the API endpoint, `https://va-opioids.data.socrata.com/api/catalog/v1`, which is based on the DOMAIN for the Socrata platform site that hosts the shared data. The API endpoint includes the parameters `domains`, which limits the returned response to datasets on the DOMAIN for the Socrata platform site that hosts the shared data, and `ids`, which limits the returned response to provide metadata for only the dataset associated with the 4x4 provided. The second argument, `auth`, is a tuple containing the user's authentication, i.e., the API Key ID and Key Secret.

```
In [7]:
    meta_data = requests.get(
        'https://va-opioids.data.socrata.com/api/catalog/v1?domains=va-opio
    ids.data.socrata.com&ids=q2f7-apb6',
        auth=('**API KEY ID**','**API KEY SECRET**'))

    print(meta_data.text)
    {
      "results" :
        [
          {
            "resource" :
              {
                "name" : "Virginia Labor Force - Unemployment",
                "id" : "q2f7-apb6",
                "parent_fxf" : [],
                "description" : "This data set contains monthly employment
    and unemployment estimates reported from the Local Area Unemployment St
```

```
atistics (LAUS) program between January 1990 and July 2020 for 134 Virg
inia counties (or county equivalent).",
            "attribution" : "Local Area Unemployment Statistics Program
",
            "attribution_link" : "https://virginiaworks.com/download-ce
nter",
            "contact_email" : null,
            "type" : "dataset",
            "updatedAt" : "2020-09-18T19:58:06.000Z",
            "createdAt" : "2020-09-18T18:58:10.000Z",
            "metadata_updated_at" : "2020-09-18T19:58:06.000Z",
            "data_updated_at" : "2020-09-18T19:58:05.000Z",
            "page_views" :
              {
                "page_views_last_week" : 9,
                "page_views_last_month" : 22,
                "page_views_total" : 22,
                "page_views_last_week_log" : 3.3219280948873626,
                "page_views_last_month_log" : 4.523561956057013,
                "page_views_total_log" : 4.523561956057013
              },
            "columns_name" :
              [
                "date",
                "fips",
                "statename",
                "areaname",
                "laborforce",
                "emplab",
                "unemp",
                "unemprate"
              ],
            "columns_field_name" :
              [
                "date",
                "fips",
                "statename",
                "areaname",
                "laborforce",
                "emplab",
                "unemp",
                "unemprate"
              ],
            "columns_datatype" :
```

```
            [
              "calendar_date",
              "number",
              "text",
              "text",
              "number",
              "number",
              "number",
              "number"
            ],
          "columns_description" :
            [
              "NIEM_Type: nc:Date.  Month and year of record.",
              "NIEM_Type: nc:LocationCountyCode.  A five-digit Federa
l Information Processing Standards code which uniquely identifies count
ies and county equivalents in the United States.",
              "NIEM_Type: nc:LocationState.  Commonwealth of Virginia
.",
              "NIEM_Type: nc:LocationCounty.  Counties in the State o
f Virginia.",
              "NIEM_Type: nc:LocationAugmentationPoint.  Number of pe
ople active in the labor force.",
              "NIEM_Type: nc:LocationAugmentationPoint.  Number of pe
ople employed in the labor force.",
              "NIEM_Type: nc:LocationAugmentationPoint.  Number of pe
ople unemployed in the labor force.",
              "NIEM_Type: nc:LocationAugmentationPoint.  Percent of p
eople in the labor force who are unemployed."
            ],
          "columns_format" :
            [
              { "view" : "date_my" },
              { "noCommas" : "true" },
              {},
              {},
              {},
              {},
              {},
              {}
            ],
          "download_count" : 0,
          "provenance" : "official",
          "lens_view_type" : "tabular",
          "blob_mime_type" : null,
```

```
            "hide_from_data_json" : false,
            "publication_date" : "2020-09-18T19:58:06.000Z"
          },
        "classification" :
          {
            "categories" : [],
            "tags" : [],
            "domain_category" : "Reference",
            "domain_tags" : [ "unemployment", "county" ],
            "domain_metadata" : [],
            "domain_private_metadata" : []
          },
        "metadata" : { "domain" : "va-opioids.data.socrata.com" },
        "permalink" : "https://va-opioids.data.socrata.com/d/q2f7-apb6"
,
        "link" : "https://va-opioids.data.socrata.com/Reference/Virgini
a-Labor-Force-Unemployment/q2f7-apb6",
        "owner" : { "id" : "gbag-8mdk", "display_name" : "Jsaad" }
      }
    ],
  "resultSetSize" : 1,
  "timings" : { "serviceMillis" : 102, "searchMillis" : [ 4, 13 ] }
}
```


The returned response, shown above, shows that the datatypes and descriptions for columns in the dataset associated with the 4x4 provided in the API endpoint, which users can use to verify the metadata. If the metadata provided in the returned response does not

Inspection of the imported data `df` reveals the data types are all objects due to the JSON format.

```
In [8]:
      df.dtypes
```

```
Out[8]:
      date          object
      fips          object
      statename     object
      areaname      object
      laborforce    object
      emplab        object
      unemp         object
      unemprate     object
      dtype: object
```

Users may optionally choose to adjust the datatypes. From the returned response, shown above, users could opt give `date` the `datetime` data type, `laborforce`,`emplab`,`unemp` the `int` data type, and `unemprate` the `float` data type.

```
In [9]:
      # change date to datetime
      df['date'] = pandas.to_datetime(df['date'])

      # change laborforce to numeric int
      df['laborforce'] = pandas.to_numeric(df['laborforce'])

      # change emplab to numeric int
      df['emplab'] = pandas.to_numeric(df['emplab'])

      # change unemp to numeric int
      df['unemp'] = pandas.to_numeric(df['unemp'])

      # change unemprate to numeric float
      df['unemprate'] = pandas.to_numeric(df['unemprate'])


In [10]:
      df.dtypes


Out[10]:
      date          datetime64[ns]
      fips                  object
      statename             object
      areaname              object
      laborforce             int64
      emplab                 int64
      unemp                  int64
      unemprate            float64
      dtype: object
```

Then, then data displays using the newly assigned datatypes.

```
In [11]:
      df
```

Out[11]:

	date	fips	statename	areaname	laborforce	emplab	unemp	unemprate
0	1990-01-01	51001	Virginia	Accomack County	14494	13293	1201	0.082862
1	1990-02-01	51001	Virginia	Accomack County	14415	13244	1171	0.081235
2	1990-03-01	51001	Virginia	Accomack County	14624	13638	986	0.067423
3	1990-04-01	51001	Virginia	Accomack County	14803	14006	797	0.053840
4	1990-05-01	51001	Virginia	Accomack County	15215	14542	673	0.044233
...	...	...	...	...	...	...	...	...
49106	2020-03-01	51840	Virginia	Winchester city	14923	14432	491	0.032902
49107	2020-04-01	51840	Virginia	Winchester city	14582	12944	1638	0.112330
49108	2020-05-01	51840	Virginia	Winchester city	14503	13182	1321	0.091085
49109	2020-06-01	51840	Virginia	Winchester city	14433	13270	1163	0.080579
49110	2020-07-01	51840	Virginia	Winchester city	14454	13364	1090	0.075412

49111 rows × 8 columns

Users can use the properly formatted data for visualizations and analytics. For instance, to create a timeseries of Richmond City's unemployment rate over the past 30 years use the following method.

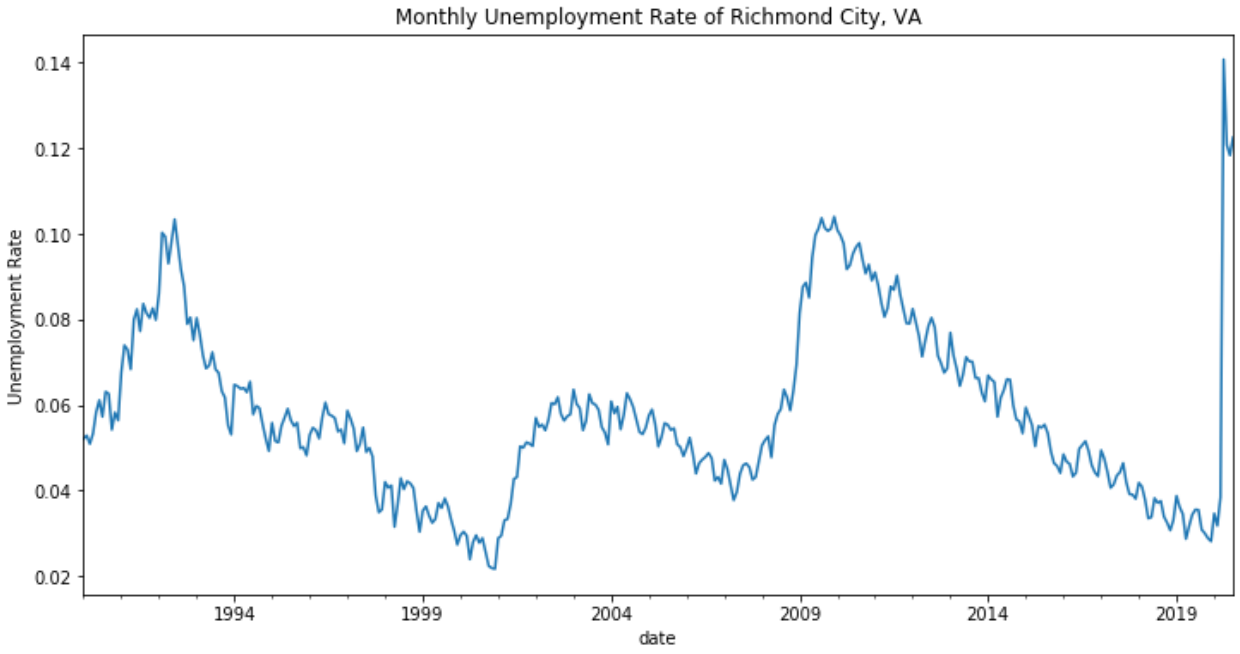
In [12]:

```
# import matplotlib.pyplots to for plotting functionality
import matplotlib.pyplot as plt

# define pandas series of richmond city unemployment rates
richmond_unemp = df['unemprate'][df['areaname']=='Richmond city']

# assign date as index of series
richmond_unemp.index = df['date'][df['areaname']=='Richmond city']

# plot the timeseries
richmond_unemp.plot(figsize=(12,6))
plt.title('Monthly Unemployment Rate of Richmond City, VA')
plt.ylabel('Unemployment Rate')
plt.show()
```



Sodapy Method

To request the data using the `sodapy` library, first import the necessary packages.

```
In [13]:
      import pandas
      from sodapy import Socrata
```

Next, connect to the Socrata domain using the `Socrata` method from `sodapy`. The first argument uses the domain name for the Socrata platform site that hosts the shared data. The second argument is the API token (not used in this example, set this value to "None", as shown). The third argument, `username`, uses the API Key ID. The fourth argument, `password`, is the API Key Secret.

```
In [14]:
client = Socrata("va-opioids.data.socrata.com", None,
                 username = '**API KEY ID**',
                 password = '**API KEY SECRET**')
```

```
WARNING:root:Requests made without an app_token will be subject to strict thr
ottling limits.
```

Most scenarios for requesting data will not experience throttling. Should users of the Socrata API to request data begin to experience throttling limits, contact the Platform Administrator to discuss provisioning an App Token.

Users can request the desired dataset by passing the 4x4 (unique dataset ID) and desired number of records using the `limit` parameter) into `client.get(...)`.

```
In [15]:
      results = client.get("q2f7-apb6", limit=50000)
```

This method returns results in JSON form.

Observe that the first two elements in `results` contain a list of dictionaries containing column and value pairs for each record.

```
In [16]:
      results[0:2]
```

```
Out[16]:
      [{'date': '1990-01-01T00:00:00.000',
        'fips': '51001',
        'statename': 'Virginia',
        'areaname': 'Accomack County',
        'laborforce': '14494',
        'emplab': '13293',
        'unemp': '1201',
        'unemprate': '0.08286187387884641'},
       {'date': '1990-02-01T00:00:00.000',
        'fips': '51001',
        'statename': 'Virginia',
        'areaname': 'Accomack County',
        'laborforce': '14415',
        'emplab': '13244',
        'unemp': '1171',
        'unemprate': '0.08123482483524107'}]
```

The `pandas` package can read dictionaries of any representation as a two-dimensional dataframe using the `DataFrame` method. To accomplish this, pass the list of dictionaries into the `DataFrame` method.

```
In [17]:
      results_df = pandas.DataFrame.from_records(results)
      results_df
```

Out[17]:

	date	fips	statename	areaname	laborforce	emplab	unemp	unemprate
0	1990-01-01T00:00:00.000	51001	Virginia	Accomack County	14494	13293	1201	0.08286187387884641
1	1990-02-01T00:00:00.000	51001	Virginia	Accomack County	14415	13244	1171	0.08123482483524107
2	1990-03-01T00:00:00.000	51001	Virginia	Accomack County	14624	13638	986	0.0674234135667396
3	1990-04-01T00:00:00.000	51001	Virginia	Accomack County	14803	14006	797	0.05384043774910491
4	1990-05-01T00:00:00.000	51001	Virginia	Accomack County	15215	14542	673	0.04423266513309234
...	...	...	...	...	...	...	...	...
49106	2020-03-01T00:00:00.000	51840	Virginia	Winchester city	14923	14432	491	0.03290223145480131
49107	2020-04-01T00:00:00.000	51840	Virginia	Winchester city	14582	12944	1638	0.11233027019613222
49108	2020-05-01T00:00:00.000	51840	Virginia	Winchester city	14503	13182	1321	0.09108460318554781
49109	2020-06-01T00:00:00.000	51840	Virginia	Winchester city	14433	13270	1163	0.08057922815769418
49110	2020-07-01T00:00:00.000	51840	Virginia	Winchester city	14454	13364	1090	0.0754116507541165

49111 rows × 8 columns

As shown in the Request Library Method, users should verify the metadata for the response and reconfigure datatypes as needed.