

Automate This

[Introduction](#)

[Resources](#)

[Publisher API](#)

[Python - Windows](#)

[System Requirements](#)

[Set Up](#)

[Screenshots](#)

[Windows](#)

[Create New Environment Variable and Set Variable Value](#)

[Install socrata-py](#)

[Run Automate This script](#)

[Mac](#)

Introduction

The purpose of this document is to provide information and examples on automating data publishing on the Socrata platform.

Resources

Python Library for new Publishing API (supports Transforms)

<https://github.com/socrata/socrata-py> <- New!

Python - Windows

<https://www.python.org/downloads/release/python-370/>

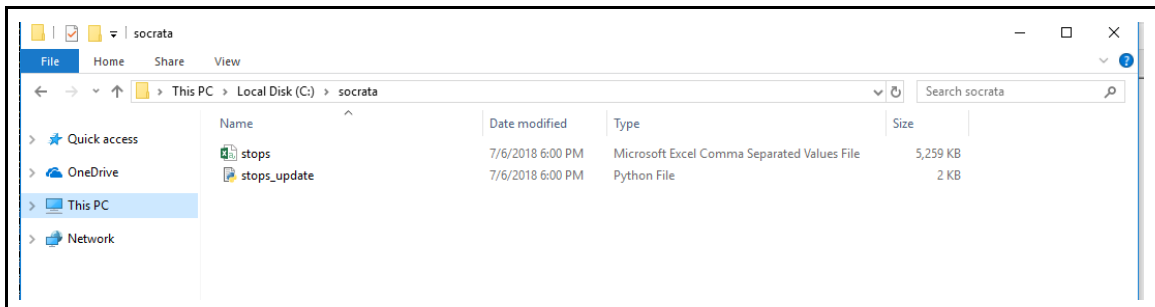
System Requirements

- A machine (app server, etc) **that can consistently run an automation job**
- Python 3** installed on that machine that will run the automation
- [socrata-py python library](#)** also installed on machine running the automation
- Socrata credentials** stored as **Environment Variables** on the OS
- A task scheduler (Windows Task Scheduler, Cron) on the same machine

Set Up & Run

Steps 1-4 are one-time setup tasks. Steps 5-10 will be repeated for each automation job.

1. Install Python (version 3+)
2. Install [socrata-py](#)
3. Set up Environment Variables permanently on [mac](#) or [windows](#)
 - a. MY_SOCRATA_USERNAME as the user variable name
 - i. Enter a Socrata username as the value
 - b. MY_SOCRATA_PASSWORD as the user variable name
 - i. Enter the corresponding Socrata password as the value
 - c. Note: you may need to restart your machine after this step.
4. Validate newly created Environment Variables
 - a. Launch new Terminal/Command Prompt after setting them
 - i. Mac: `echo $MY_SOCRATA_USERNAME`
 - ii. Windows: `echo %MY_SOCRATA_USERNAME%`
5. Copy *Automate This* script from Socrata menu when creating or updating a dataset.
6. Paste the script into a file (perhaps an IDE like IDLE, or a text editor)
7. Save *Automate This* script in an appropriate folder, perhaps named Socrata



8. Update file path in the script to file that will be published to Socrata

```

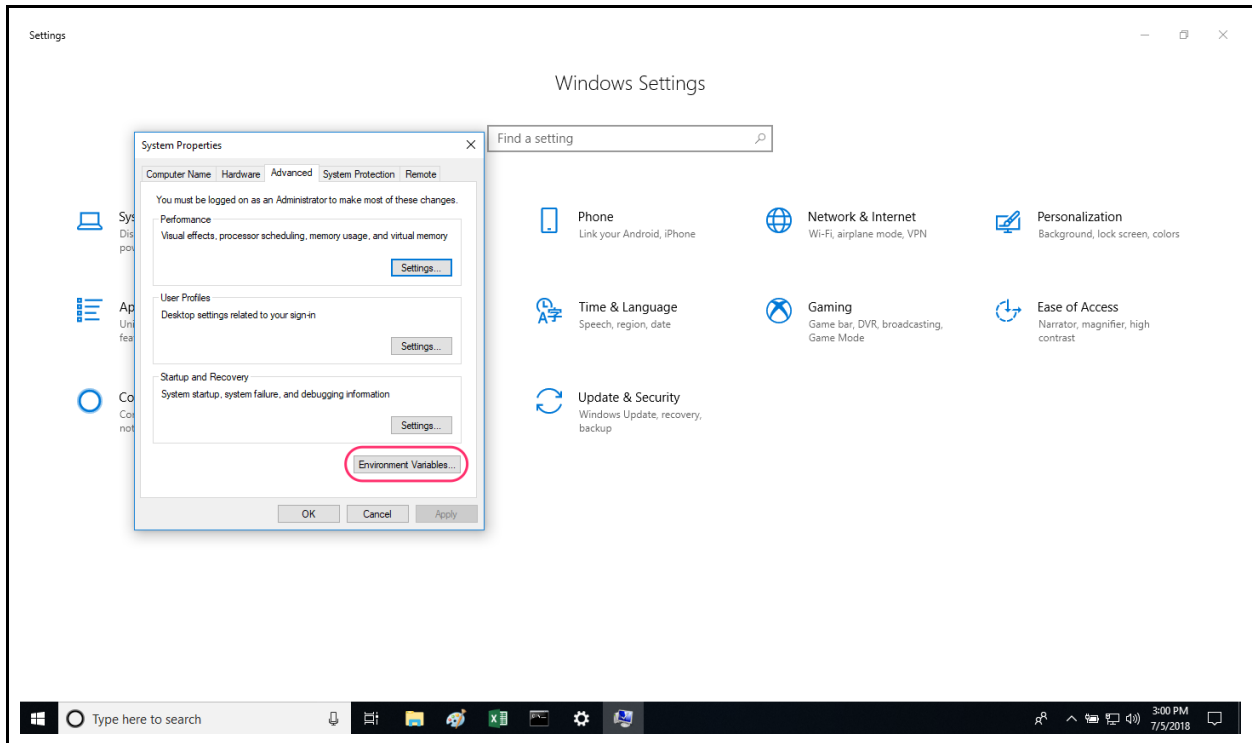
11 # set environmental vars for current user using >nano .bash_profile
12 auth = Authorization(
13     'alicia-cloud.demo.socrata.com',
14     os.environ['MY_SOCRATA_USERNAME'],
15     os.environ['MY_SOCRATA_PASSWORD']
16 )
17
18 socrata = Socrata(auth)
19
20 (ok, view) = socrata.views.lookup('4vym-79aq')
21 assert ok, view
22
23 with open('/Users/alicia.brown/data/stops.csv', 'rb') as my_file:
24     (ok, job) = socrata.using_config('stops_prior_2018_06-18-2018_99be', view).csv(my_file)
25     assert ok, job
26 # These next 3 lines are optional - once the job is started from the previous line, the
27 # script can exit; these next lines just block until the job completes
28     assert ok, job
29     (ok, job) = job.wait_for_finish(progress = lambda job: print('Job progress:', job.attributes['status']))
30     sys.exit(0 if job.attributes['status'] == 'successful' else 1)
31
  
```

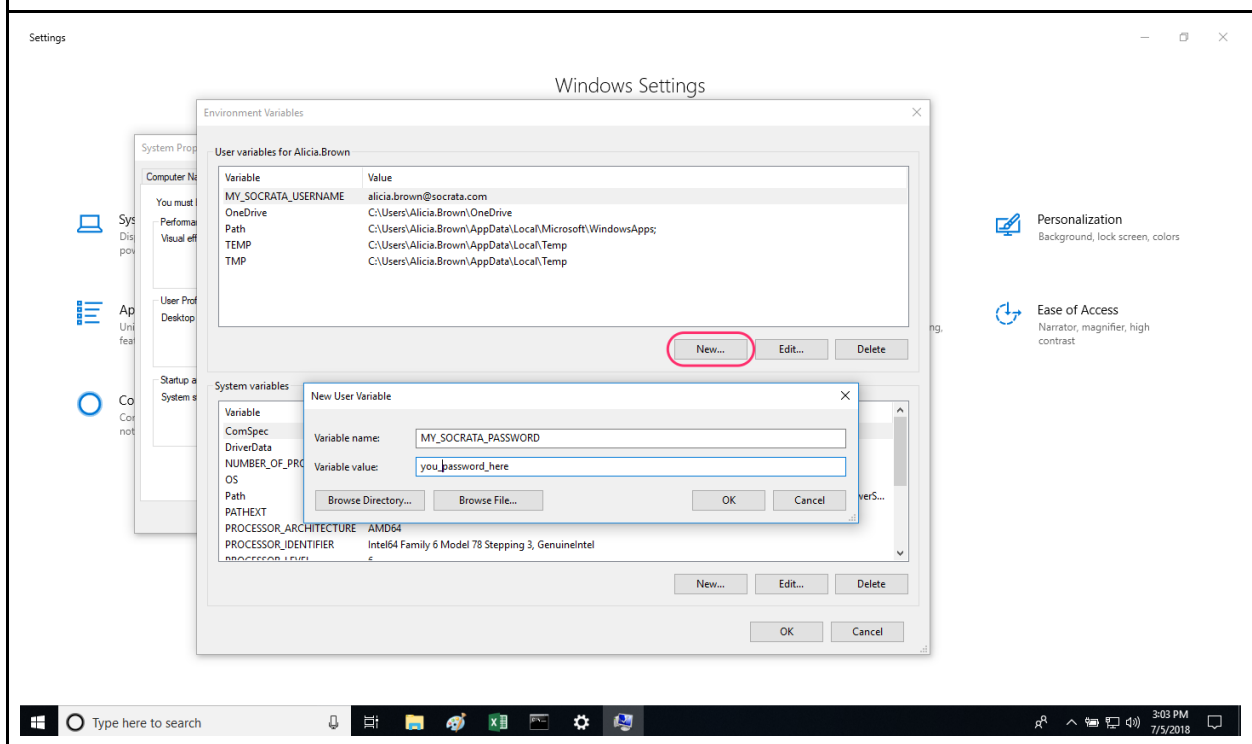
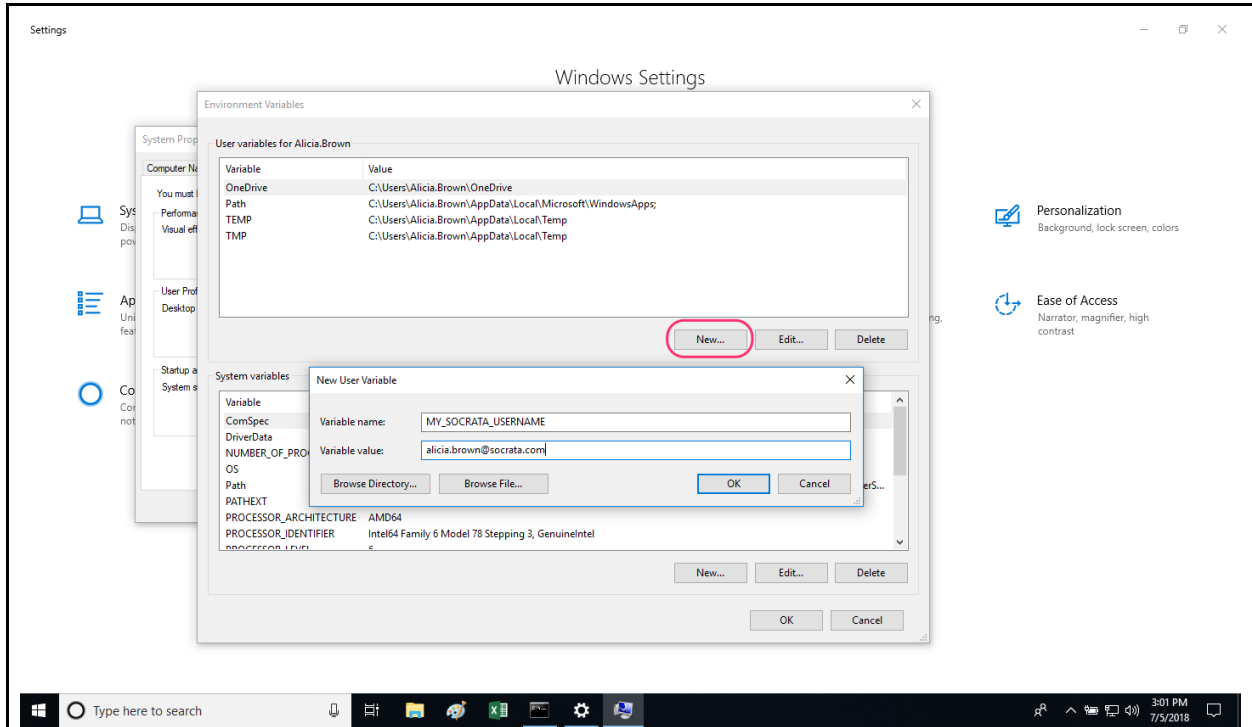
9. Run script from command prompt/terminal/task scheduler: `python stops_update.py`
10. Voila!

Screenshots

Windows

Create New Environment Variable and Set Variable Value





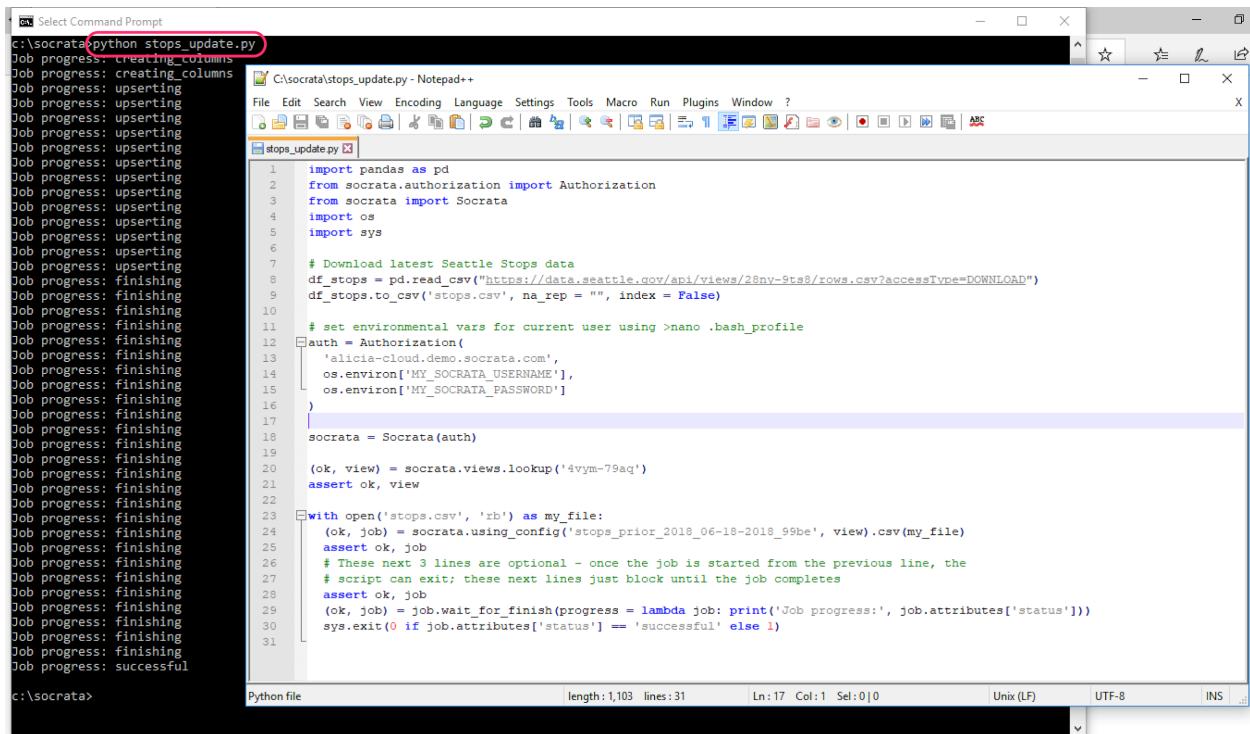
Install socrata-py

pip3 install socrata-py

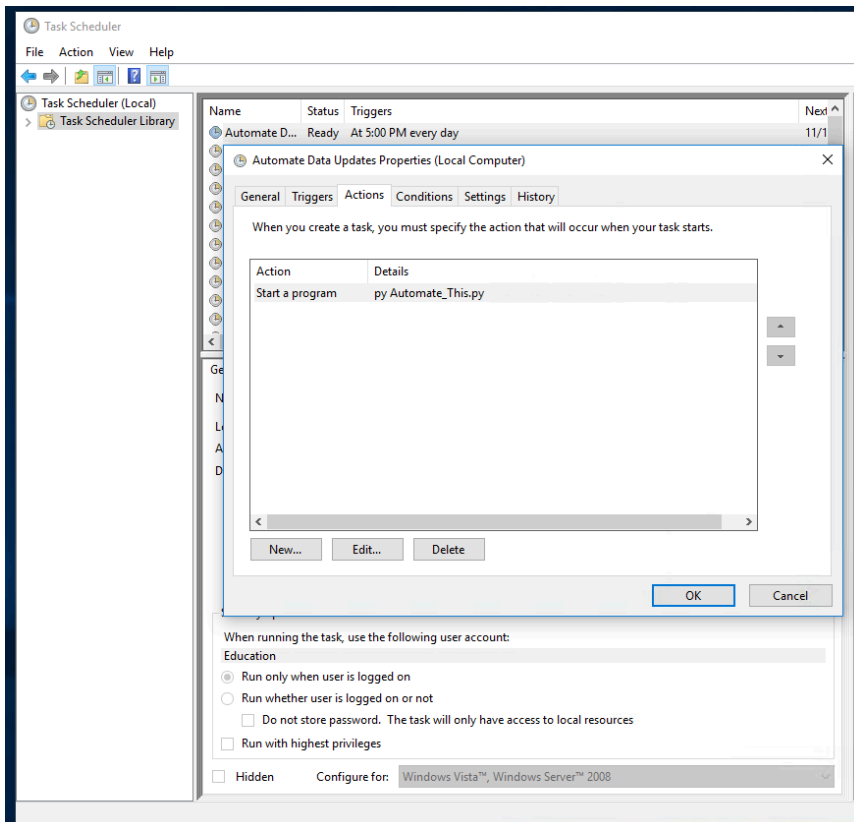
```
cmd Select Command Prompt
C:\Users\Alicia.Brown>pip3 install socrata-py
Collecting socrata-py
  Downloading https://files.pythonhosted.org/packages/68/77/29bc782e0b9729ac6181d4e6edba5d99f2d5406453aa973e09afaba92a4a/socrata-py-0.4.12.tar.gz
Collecting requests (from socrata-py)
  Downloading https://files.pythonhosted.org/packages/65/47/7e02164a2a3db50ed6d8a6ab1d6d60b69c4c3fdf57a284257925dfc12bda/requests-2.19.1-py2.py3-none-any.whl (91kB)
  100% |#####| 92kB 2.9MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests->socrata-py)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
  100% |#####| 143kB 6.6MB/s
Collecting certifi>=2017.4.17 (from requests->socrata-py)
  Downloading https://files.pythonhosted.org/packages/7c/e6/92ad559b7192d846975fc916b65f667c7b8c3a32bea7372340bfe9a15fa5/certifi-2018.4.16-py2.py3-none-any.whl (150kB)
  100% |#####| 153kB 6.6MB/s
Collecting urllib3<1.24,>=1.21.1 (from requests->socrata-py)
  Downloading https://files.pythonhosted.org/packages/bd/c9/6fdd990019071a4a32a5e7cb78a1d92c53851ef4f56f62a3486e6a7d8ffb/urllib3-1.23-py2.py3-none-any.whl (133kB)
  100% |#####| 143kB 6.6MB/s
Collecting idna<2.8,>=2.5 (from requests->socrata-py)
  Downloading https://files.pythonhosted.org/packages/4b/2a/0276479a4b3caeb8a8c1af2f8e4355746a97fab05a372e4a2c6a6b876165/idna-2.7-py2.py3-none-any.whl (58kB)
  100% |#####| 61kB 2.0MB/s
Installing collected packages: chardet, certifi, urllib3, idna, requests, socrata-py
  The script chardetect.exe is installed in 'c:\users\alicia.brown\appdata\local\programs\python\python37\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  Running setup.py install for socrata-py ... done
Successfully installed certifi-2018.4.16 chardet-3.0.4 idna-2.7 requests-2.19.1 socrata-py-0.4.12 urllib3-1.23

C:\Users\Alicia.Brown>pip3 install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/ee/c2/fb027ab321657acb42d270e26a4f35795075f394dbc30f67e19883fdb375/pandas-0.23.2-cp37-cp37m-win_amd64.whl (7.9MB)
  100% |#####| 7.9MB 1.6MB/s
Collecting numpy>=1.9.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/94/80/c49b01d8632f58aef25f8e9a05be56339b7bb94b1eefd4f5d8c087d002b5/numpy-1.14.5-cp37-cp37m-win_amd64.whl (13.4MB)
  100% |#####| 13.4MB 128kB/s
Collecting pytz>=2011k (from pandas)
  Downloading https://files.pythonhosted.org/packages/30/4e/27c34b62430286c6d59177a0842ed90dc789ce5d1ed740887653b898779a/pytz-2018.5-py2.py3-none-any.whl (510kB)
  100% |#####| 512kB 3.3MB/s
Collecting python-dateutil>=2.5.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/cf/f5/af2b09c957ace60dcfac112b669c45c8c97e32f94aa8b56da4c6d1682825/python_dateutil-2.7.3-py2.py3-none-any.whl (211kB)
  100% |#####| 215kB 3.3MB/s
```

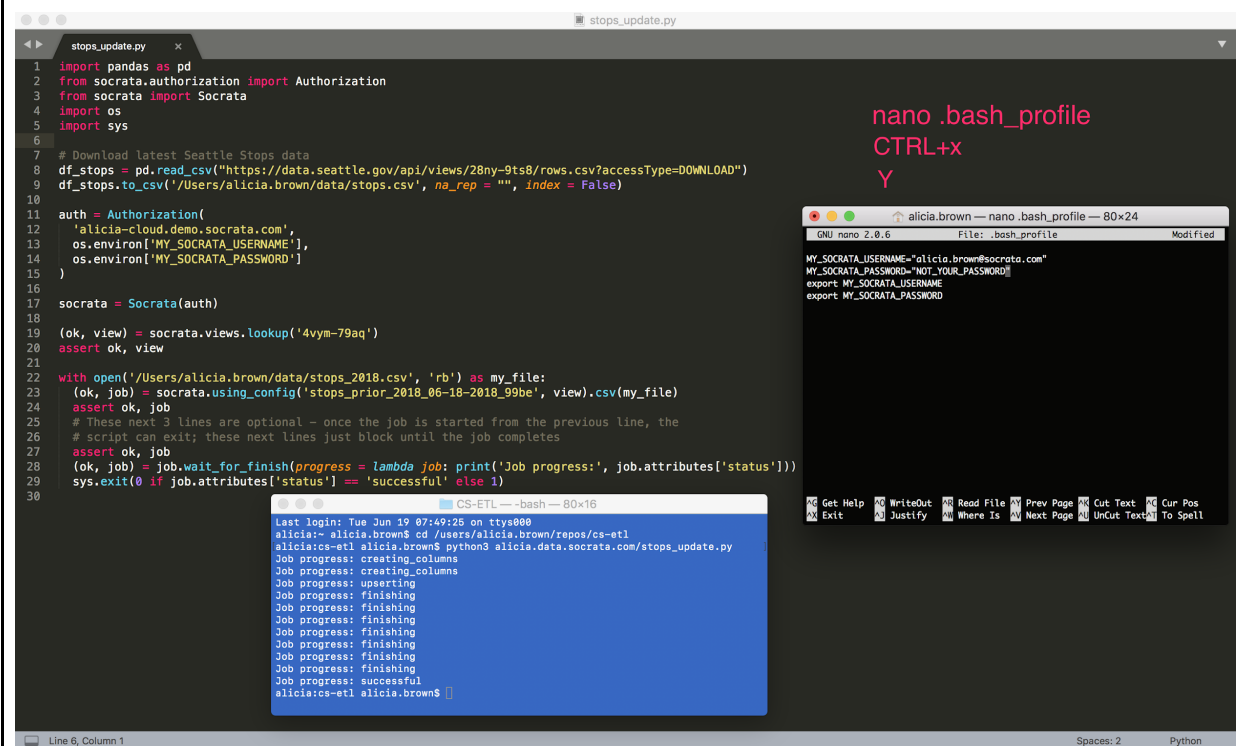
Run Automate This script



From Windows Task Scheduler



Mac



The screenshot displays a Mac terminal window with two active windows. The primary window, titled 'stops_update.py', contains a Python script that uses the Socrata API to fetch and process data. The script includes imports for pandas, Socrata, os, and sys. It downloads data from a Socrata view, authenticates with a demo account, and then uses the Socrata SDK to create a job for processing CSV data. The script includes progress reporting and a lambda function to handle job completion. The output of the script shows the job being created, columns being processed, and the job finishing successfully.

```
stops_update.py
1 import pandas as pd
2 from socrata.authorization import Authorization
3 from socrata import Socrata
4 import os
5 import sys
6
7 # Download latest Seattle Stops data
8 df_stops = pd.read_csv("https://data.seattle.gov/api/views/28ny-9ts8/rows.csv?accessType=DOWNLOAD")
9 df_stops.to_csv('/Users/alicia.brown/data/stops.csv', na_rep = '', index = False)
10
11 auth = Authorization(
12     'alicia-cloud.demo.socrata.com',
13     os.environ['MY_SOCRATA_USERNAME'],
14     os.environ['MY_SOCRATA_PASSWORD']
15 )
16
17 socrata = Socrata(auth)
18
19 (ok, view) = socrata.views.lookup('4vym-79aq')
20 assert ok, view
21
22 with open('/Users/alicia.brown/data/stops_2018.csv', 'rb') as my_file:
23     (ok, job) = socrata.using_config('stops_prior_2018_06-18-2018_99be', view).csv(my_file)
24     assert ok, job
25     # These next 3 lines are optional - once the job is started from the previous line, the
26     # script can exit; these next lines just block until the job completes
27     assert ok, job
28     (ok, job) = job.wait_for_finish(progress = lambda job: print('Job progress:', job.attributes['status']))
29     sys.exit(0 if job.attributes['status'] == 'successful' else 1)
30
```

The second window, titled 'alicia.brown - nano .bash_profile', shows the nano editor editing the .bash_profile file. The file content is as follows:

```
GNU nano 2.0.6 File: .bash_profile Modified
MY_SOCRATA_USERNAME="alicia.brown@socrata.com"
MY_SOCRATA_PASSWORD="NOT_YOUR_PASSWORD"
export MY_SOCRATA_USERNAME
export MY_SOCRATA_PASSWORD
```

Red text annotations in the terminal window indicate the nano editor shortcuts: 'nano .bash_profile', 'CTRL+x', and 'Y'.

Environment Variables on a Mac are stored in .bash_profile