## Build the future of communications.

START BUILDING FOR FREE

BY **GREG BAUGUES** • 2017-02-16

🐦 TWITTER    📘 FACEBOOK    in LINKEDIN

# Google Spreadsheets and Python



This post is inspired by Patrick McKenzie's reminder that sometimes you don't need a database:

**Patrick McKenzie**
@patio11

So if you're building out a quick CRUD app for e.g. internal use, Google Docs as a backend (consumed via JSON) is *surprisingly* powerful.
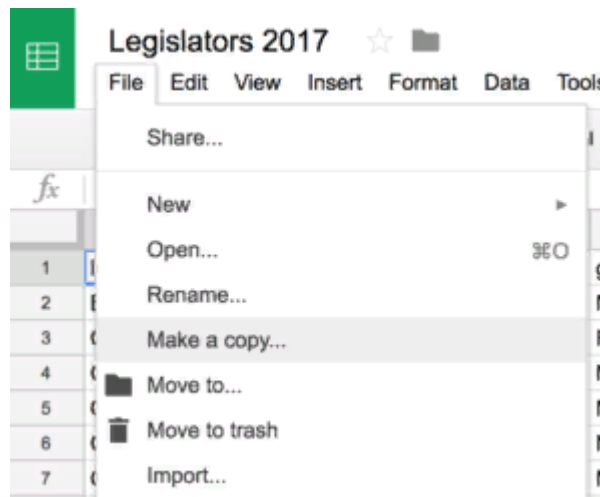
♡ 110    3:52 AM - Jul 5, 2014    ⓘ

In this tutorial, we'll use Anton Burnashev's excellent gspread Python package to read, write, and delete data from a Google Spreadsheet with just a few lines of code.

# Google Drive API and Service Accounts

At the risk of being Captain Obvious, you're going to need a spreadsheet if you want to follow along with this post. If you don't have one on hand that's full of juicy data, might I suggest you make a copy of this spreadsheet with contact information for all United States legislators? (Side note: Ian Webster uses this data in conjunction with Twilio to make it easy for citizens to call congress).
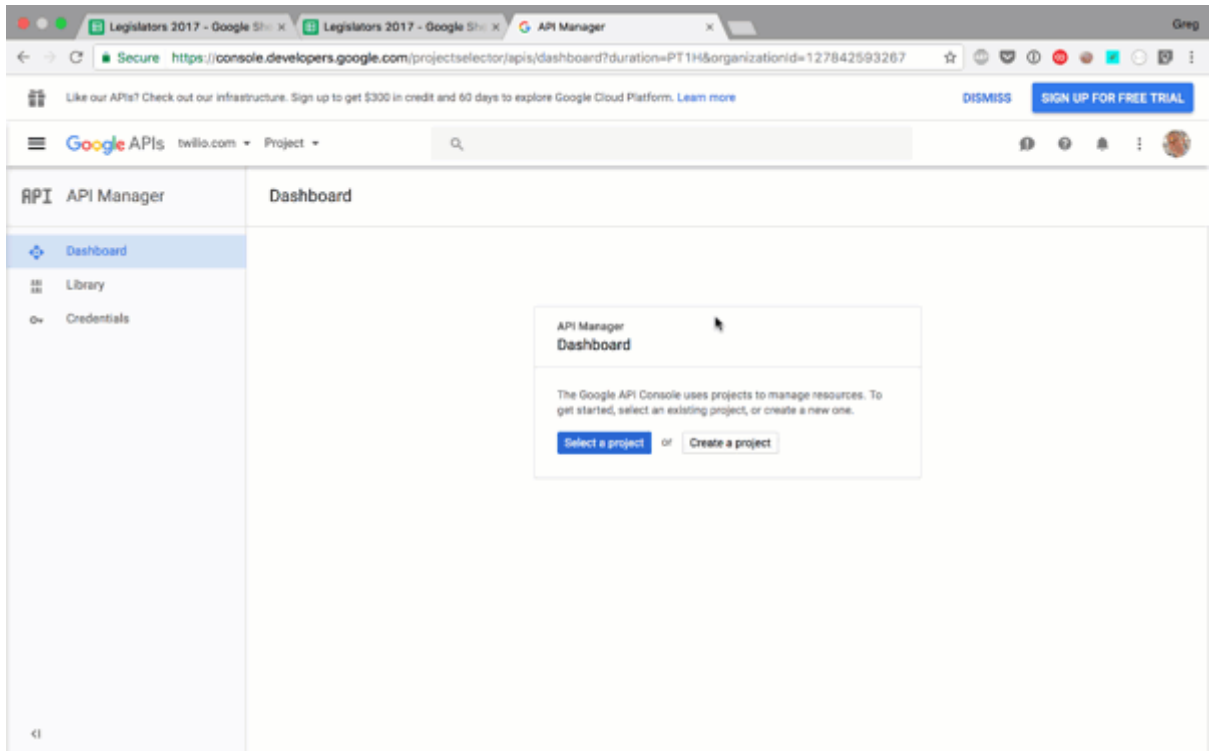


To programmatically access your spreadsheet, you'll need to create a service account and OAuth2 credentials from the Google API Console. If you've been traumatized by OAuth2 development before, don't worry; service accounts are *way* easier to use.

Follow along with the steps and GIF below. You'll be in and out of the console in 60 seconds (much like Nic Cage in your favorite Nic Cage movie).

1. Go to the Google APIs Console.

2. Create a new project.

3. Click *Enable API*. Search for and enable the Google Drive API.

4. *Create credentials* for a *Web Server* to access *Application Data.*

5. Name the service account and grant it a *Project* Role of *Editor.*

6. Download the JSON file.

7. Copy the JSON file to your code directory and rename it to client_secret.json



There is one last required step to authorize your app, and it's easy to miss!

Find the  client_email inside client_secret.json. Back in your spreadsheet, click the *Share* button in the top right, and paste the client email into the People field to give it edit rights. Hit Send.

If you skip this step, you'll get a gspread.exceptions.SpreadsheetNotFound error when you try to access the spreadsheet from Python.

```
    client_secret.json  ×
  1  {
  2    "type": "service_account",
  3    "project_id": "legislators-157916",
  4    "private_key_id": "f3d43d89a5c19ea8a05317270025613ebb5cc8af",
  5    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAA
  6    "client_email": "legislators@legislators-157916.iam.gserviceaccount.com",
  7    "client_id": "102880681777284010484",
  8    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  9    "token_uri": "https://accounts.google.com/o/oauth2/token",
 10    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs"
 11    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/
 12  }
 13
Line 8, Column 20                                                    Spaces: 2      JSON
```

We're done with the boring part! Now onto the code.

# Read Data from a Spreadsheet with Python

With credentials in place (you did copy them to your code directory, right?) accessing a Google Spreadsheet in Python requires just two packages:

1. oauth2client – to authorize with the Google Drive API using OAuth 2.0

2. gspread – to interact with Google Spreadsheets

Install these packages with:

```
1  pip install gspread oauth2client
```

Then paste this code into a new file called spreadsheet.py:

```
1  import gspread
2  from oauth2client.service_account import ServiceAccountCredentials
3
4
5  # use creds to create a client to interact with the Google Drive API
```

```
 6   scope = ['https://spreadsheets.google.com/feeds']
 7   creds = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json
 8   client = gspread.authorize(creds)
 9
10   # Find a workbook by name and open the first sheet
11   # Make sure you use the right name here.
12   sheet = client.open("Copy of Legislators 2017").sheet1
13
14   # Extract and print all of the values
15   list_of_hashes = sheet.get_all_records()
16   print(list_of_hashes)
```

Run python spreadsheet.py and marvel at the glorious, well-formatted data.

```
ok_id': '', 'party': 'Republican', 'type': 'rep', 'rss_url': '', 'lis_id': '', 'govtrack_id':
 412729, 'bioguide_id': 'G000580', 'birthday': '3/27/1972', 'facebook': '', 'address': '415 C
annon HOB; Washington DC 20515-4605', 'contact_form': '', 'opensecrets_id': '', 'phone': '202
-225-4711', 'url': '', 'gender': 'M', 'washington_post_id': '', 'youtube': '', 'ballotpedia_i
d': '', 'youtube_id': ''}, {'cspan_id': '', 'last_name': 'Jayapal', 'twitter': '', 'icpsr_id'
: '', 'wikipedia_id': '', 'thomas_id': '', 'first_name': 'Pramila', 'district': 7, 'state': '
WA', 'votesmart_id': '', 'facebook_id': '', 'party': 'Democrat', 'type': 'rep', 'rss_url': ''
, 'lis_id': '', 'govtrack_id': 412730, 'bioguide_id': 'J000298', 'birthday': '9/21/1965', 'fa
cebook': '', 'address': '319 Cannon HOB; Washington DC 20515-4707', 'contact_form': '', 'open
secrets_id': '', 'phone': '202-225-3106', 'url': '', 'gender': 'F', 'washington_post_id': '',
 'youtube': '', 'ballotpedia_id': '', 'youtube_id': ''}, {'cspan_id': '', 'last_name': 'Galla
gher', 'twitter': '', 'icpsr_id': '', 'wikipedia_id': '', 'thomas_id': '', 'first_name': 'Mik
e', 'district': 8, 'state': 'WI', 'votesmart_id': '', 'facebook_id': '', 'party': 'Republican
', 'type': 'rep', 'rss_url': '', 'lis_id': '', 'govtrack_id': 412731, 'bioguide_id': 'G000579
', 'birthday': '3/3/1984', 'facebook': '', 'address': '1007 Longworth HOB; Washington DC 2051
5-4908', 'contact_form': '', 'opensecrets_id': '', 'phone': '202-225-5665', 'url': '', 'gende
r': 'M', 'washington_post_id': '', 'youtube': '', 'ballotpedia_id': '', 'youtube_id': ''}, {'
cspan_id': '', 'last_name': 'Cheney', 'twitter': '', 'icpsr_id': '', 'wikipedia_id': '', 'tho
mas_id': '', 'first_name': 'Liz', 'district': 0, 'state': 'WY', 'votesmart_id': '', 'facebook
_id': '', 'party': 'Republican', 'type': 'rep', 'rss_url': '', 'lis_id': '', 'govtrack_id': 4
12732, 'bioguide_id': 'C001109', 'birthday': '7/28/1966', 'facebook': '', 'address': '416 Can
non HOB; Washington DC 20515-5000', 'contact_form': '', 'opensecrets_id': '', 'phone': '202-2
25-2311', 'url': '', 'gender': 'F', 'washington_post_id': '', 'youtube': '', 'ballotpedia_id'
: '', 'youtube_id': ''}]
(spreadsheet-python-review) → google-spreadsheet-python-review
```

# Insert, Update, and Delete from a Spreadsheet with Python

We've just scratched the surface of gspreads' well documented and comprehensive functionality.

For instance, we extracted the data into a list of hashes, but you can get a list of lists if you'd prefer:

```
1  sheet.get_all_values()
```

Or you could just pull the data from a single row, column, or cell:

```
1  sheet.row_values(1)
2
3  sheet.col_values(1)
4
5  sheet.cell(1, 1).value
```

You can write to the spreadsheet by changing a specific cell:

```
1  sheet.update_cell(1, 1, "I just wrote to a spreadsheet using Python!")
```

Or you can insert a row in the spreadsheet:

```
1  row = ["I'm","inserting","a","row","into","a,","Spreadsheet","with","Python"
2  index = 1
3  sheet.insert_row(row, index)
```

You can also delete a row from the spreadsheet:

```
1  sheet.delete_row(1)
```

And find out the total number of rows:

```
1  sheet.row_count
```

Check the gspread API reference for the full details on these functions along with a few dozen others.

Using Google Spreadsheets with Python opens possibilities like building a Flask app with a spreadsheet as the persistence layer, or importing data from a Google spreadsheet into Jupyter Notebooks and doing analysis in Pandas. If you want to start playing with Python and Twilio, check out our Python quickstarts.

If you build something cool, please let me know. You can find me at gb@twilio.com or @greggyb. And if this post was helpful, please share it with someone else who might dig it.

*Many thanks to Devin and Sam for the reviews, to Google for Sheets, and most of all, to Anton for gspread.*

AUTHORS  |  Greg Baugues

Build More With Python

- Secure your Python Amazon Lambda App by Validating Incoming Twilio Requests

- The Twilio Python Helper Library

- Developer Digest: Automating Wedding Stress Away in Python

- Mock it Til' You Make It: Test Django with mock and httpretty

- Selling 5,000+ Boxes of Girl Scout Cookies with a bit of Twilio Magic

**Comments for this thread are now closed**  ✕

**52 Comments**  **Twilio Blog**  💬 Сергей Кущенко  ▾

♡ **Recommend** 22    🐦 **Tweet**   f **Share**    Sort by Oldest ▾

**Jed** • 3 years ago
Keep an eye on https://github.com/nithinmu... which supports v4 of the sheets API. Might already help speed up inserts into large sheets, for example.
2 ⌃ | ⌄ 1 • Share ›

> **Jebb** ➜ Jed • 3 years ago • edited
> Looks like the author of gspread plans to move to v4 too https://github.com/burnash/...
> 1 ⌃ | ⌄ • Share ›

>> **Greg Baugues** ➜ Jebb • 3 years ago
>> This is good info. Thanks **@Jebb**.
>> ⌃ | ⌄ • Share ›

>>> **Bob Belderbos** ➜ Greg Baugues • 3 years ago • edited
>>> Confirmed: was using this in Flask and a POST request to insert a cell took almost 2 min, superslow. Moving to

pygsheets and it takes < 2 seconds, so thanks for this suggestion Jed/Jebb.

And thanks Greg for this excellent article. It inspired me to roll my own: http://pybit.es/flask-api-p...

2 ∧ | ∨ • Share ›

**Te ENe Te** ➦ Bob Belderbos • 2 years ago

**@Bob Belderbos**... can you share your code please?

∧ | ∨ • Share ›

**Bob Belderbos** ➦ Te ENe Te • 2 years ago

Hi, code is in the article I linked ->
https://github.com/pybites/...

∧ | ∨ • Share ›

**pizzapanther** • 3 years ago

The big downside I've found using this a lot in the past is that Google has no SLA for sheets. Sure its Google so it has high availability but things can quite often fail with no support from Google. So just be careful of building anything mission critical with this. The API can be wonky every once in a while. At one point it even took them like 6 months to fix an issue I had with the Apps Script API. If my business relied on that functionality, I would have been screwed.

The Jupyter Notebook use case sounds great but if you move to something that is more critical use a real data store.

∧ | ∨ • Share ›

**ZoublWah** ➦ pizzapanther • 3 years ago

same. you need to handle a lot of random error conditions in your scripts because google can go from basically cutting you off ("connection lost") to random 500 errors and even 404 for the exact same request.

And of course you then need to pool until the API works again.

In addition to that some of the queries are extremely slow.

Basically it feels like a 3rd class citizen at best. Now then again I fully use it for years and do not intend to stop.

∧ | ∨ • Share ›

**Andrey Fedorov** ➦ pizzapanther • 3 years ago

> just be careful of building anything mission critical with [Google Sheets as a backend]

wat

∧ | ∨ • Share ›

**Michael R. Bernstein** ➦ Andrey Fedorov • 3 years ago

Well, you know, sometime you build an MVP, PoC, or prototype, and it gets deployed to production by management fiat (also known as

protoduction).

2 ∧ | ∨ • Share ›

**Dilshod** • 3 years ago

Thank you Greg for the great tutorial. I would like to know what program did you use for gif animations. It would be useful if you share it too. Thanks again.

∧ | ∨ • Share ›

**Greg Baugues** ➔ Dilshod • 3 years ago • edited

**@Dilshod** I would love to share because Camtasia is perhaps my favorite piece of software installed on my machine! It's the "it just works" of screencasting. https://www.techsmith.com/c...

∧ | ∨ • Share ›

**James Abela** ➔ Greg Baugues • 3 years ago

I second that! Its an amazing piece of software... Once you learn it, you can knock out screencasts for any lesson you please with very little notice.

9 ∧ | ∨ • Share ›

**Excellion** • 3 years ago

Hello, great tutorial. Thanks. Is there any API to download an entire spreadsheet as an MS Excel?

∧ | ∨ • Share ›

**Alan** ➔ Excellion • 3 years ago

import pandas as pd

your_data = sheet.get_all_values()
your_data = pd.DataFrame(your_data)
your_data.to_csv('filename.csv')

Open csv in Excel and convert it to .xlsx if you like

∧ | ∨ • Share ›

**Dustin** ➔ Alan • 3 years ago

Or use Pandas to save it as Excel.

∧ | ∨ • Share ›

**Chris Emery** • 3 years ago

It seems it maybe a good use case for distributed development troubleshooting. Maybe using hexagonal programming structure would allow a development app. to switch between sample, problem and production data in a simplified and distributed fashion.

∧ | ∨ • Share ›

**IssaKnife** • 3 years ago

For some reason I am unable to install gspread and oauth2client. I get an error that says Exception:
Traceback

Traceback
then lists a bunch of file paths followed by:
Permission denied: '/usr/local/lib/python2.7/dist-packages/httplib2-0.9.2-
py2.7.egg/EGG-INFO/PKG-INFO'

Any advice?

⌃ | ⌄ • Share ›

**James Abela** ➜ IssaKnife • 3 years ago
This is a wonderful tutorial and thanks for making it.

To address some of the comments, if you want to be less dependent on
Google Sheets, but still want access to data on it. You can publish to CSV
file. Python can then pull this data in from the web. (Or of course you can
download a CSV file, which I've done in the past for static data)

⌃ | ⌄ • Share ›

**Jur** • 3 years ago • edited
If you just want to read data in as a CSV and don't mind your data being public,
your life can be a lot easier if you use this URL format to download a public Google
Sheet as CSV:

```
https://docs.google.com/spreadsheets/d/{google sheet identifier}/export?gid={gid i
```

I find this the easiest way to get Google Spreadsheet data into a Jupyter notebook
for data analysis with Pandas.

1 ⌃ | ⌄ • Share ›

**Jan Oglop** ➜ Jur • 3 years ago
Let me add that line separators are "\r\n" ... I've been fighting with it whole
day

1 ⌃ | ⌄ • Share ›

**zhangjingqiang** • 3 years ago
I also write a blog about write data to google spreadsheet. All of them using raw
google v4 api. -- http://qiang-blog.herokuapp...

⌃ | ⌄ • Share ›

**Bruno** • 3 years ago
Thanks for this quick tutorial, very useful!

> At the risk of being Captain Obvious, you're going to need a spreadsheet if you
want to follow along with this post

I looks like it's possible to create a spreadsheet with gspread:
https://github.com/burnash/...
Was it omitted on purpose? Is it because it needs to be shared by the script if done
that way?

⌃ | ⌄ • Share ›

**Lee** • 3 years ago

I've used Sheets for some simple data hosting before, but gave Fieldbook a try as the backend when knocking up a simple static site I found the API much simpler and the interface and functionality slightly better than Sheets (for my purpose at least). There's some details here: http://viewfinderdesign.co....

∧ | ∨ • Share ›

**David Okwii** • 3 years ago

Hello, I run into the following error ;

python spreadsheet.py
Traceback (most recent call last):
File "spreadsheet.py", line 12, in <module>
sheet = client.open("Cost of Foods In Kampala").sheet1
File "/usr/local/lib/python2.7/dist-packages/gspread/client.py", line 82, in open
feed = self.get_spreadsheets_feed()
File "/usr/local/lib/python2.7/dist-packages/gspread/client.py", line 155, in get_spreadsheets_feed
r = self.session.get(url)
File "/usr/local/lib/python2.7/dist-packages/gspread/httpsession.py", line 73, in get
return self.request('GET', url, params=params, **kwargs)
File "/usr/local/lib/python2.7/dist-packages/gspread/httpsession.py", line 65, in request
response = func(url, data=data, params=params, headers=request_headers, files=files, json=json)
File "/usr/lib/python2.7/dist-packages/requests/sessions.py", line 467, in get
return self.request('GET', url, **kwargs)
TypeError: request() got an unexpected keyword argument 'json'

∧ | ∨ • Share ›

> **Von Foerster** → David Okwii • 3 years ago
> I've got the same one.
> Did you solve it? Can you help?
> Thanks for the tutorial!
>
> ∧ | ∨ • Share ›

> > **David Okwii** → Von Foerster • 3 years ago
> > I had to upgrade the requests library which meant moving to Python 3.x from 2.7. That's how I solve the issue.
> >
> > ∧ | ∨ • Share ›

**Gery Greyhound O1G** • 3 years ago

Thanks for the awesome tutorial!

I'm mainly into motion graphic design, and a really a newbie to coding and stuff like APIs. I started learning Python some 2 weeks ago primarily to make automated data visualization animations in Cinema 4D, and now it's a huge step forward that I can use Google Spreadsheet data. I really apreciate this tutorial :)

My only problem is that while this stuff works like a charm on my standalone Python 3.6 installation, but I can't find out how to install oauth2client and gspread modules

into the Python of Cinema 4D. Anyone has some ideas?

∧ | ∨ • Share ›

**John Mathews** • 3 years ago

Thanks for this. Super useful and clear!

∧ | ∨ • Share ›

**Ron Serruya** • 3 years ago

Thanks ! This helped me a lot

∧ | ∨ • Share ›

**Manikandan K** • 3 years ago

hello..i want a webapp with google spreadsheet database where one can search a data and results mst be displayed as webpage

1 ∧ | ∨ • Share ›

**Brandon Moore** • 3 years ago • edited

Remember to edit the spreadsheet.py to reflect the name of your own spreadsheet. The tutorial didnt do that and cost me some valuable minutes. :)

∧ | ∨ • Share ›

**Wenhao Xuan** • 3 years ago

Will I need to create a new project in the Google APIs Console for a new spreadsheet?

∧ | ∨ • Share ›

**Sunny ke** • 3 years ago

Thank you for sharing. This is very helpful.
I tried to convert it to executable from py2exe. When I ran, I got error:

Traceback (most recent call last):
File "spreadsheet.py", line 8, in <module>
File "gspread\client.pyc", line 402, in authorize
File "gspread\client.pyc", line 61, in login
File "oauth2client\client.pyc", line 545, in refresh
File "oauth2client\client.pyc", line 749, in _refresh
File "oauth2client\client.pyc", line 780, in _do_refresh_request
File "oauth2client\transport.pyc", line 282, in request
File "httplib2\__init__.pyc", line 1659, in request
File "httplib2\__init__.pyc", line 1399, in _request
File "httplib2\__init__.pyc", line 1319, in _conn_request
File "httplib2\__init__.pyc", line 1069, in connect
File "httplib2\__init__.pyc", line 96, in _ssl_wrap_socket
IOError: [Errno 2] No such file or directory

∧ | ∨ • Share ›

**gif** ➜ Sunny ke • 2 years ago

did you find out something? I'm still suffering from the same bug

∧ | ∨ • Share ›

**Fernando Simao** • 3 years ago

Thanks a lot! Very good job!! Fernando Simão from Rio de Janeiro, Brazil.

∧ | ∨ • Share ›

**Erik** • 2 years ago

Get an error when trying to import ServiceAccountCredentials:

TypeError: unsupported operand type(s) for |: 'frozenset' and 'list'

∧ | ∨ • Share ›

**Curtis Salisbury** → Erik • 2 years ago

I am getting the same error as well.

∧ | ∨ • Share ›

**omer** • 2 years ago • edited

Hi thank for this tutorial. But I get the next error:

Traceback (most recent call last):
File "C:/Users/Drew/PycharmProjects/Real Estate/googlesheetstest.py", line 6, in <module>
creds = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json', scope)
File "C:\Python27\lib\site-packages\oauth2client\service_account.py", line 219, in from_json_keyfile_name
with open(filename, 'r') as file_obj:
IOError: [Errno 2] No such file or directory: 'client_secret.json'

I didn't understand where I need to save the json file so i saved on my desktop maybe this the problem?

∧ | ∨ • Share ›

**Charles Cuthbert** → omer • 2 years ago

Omer: I have the same problem. Did you manage to fix it?

I saved the 'client_secret.json' file in oauth2client directory, but it does not work:
C:\Users\charles\AppData\Local\Programs\Python\Python36-32\lib\site-packages\oauth2client"

Charles

∧ | ∨ • Share ›

**Oliv698** → Charles Cuthbert • 2 years ago

With iPython, my json file is located in the same repository as my ipynb file. with a Python script, it should be the same, no?

∧ | ∨ • Share ›

**Oliv698** → Oliv698 • 2 years ago

sorry, in the root repository of my iPython installation (with Anaconda, my user folder with W7

**Steve Shaw** • 2 years ago • edited

Can I use this to access all files in a specified folder as opposed to just a single sheet? Also how would I access a file within a folder in drive?

EDIT: I got it, just need to share the folder where the file is with service account email address, don't need to specify path in code

1 ∧ | ∨ • Share ›

**Razvan Zaharia** • 2 years ago

Following your youtube tutorial I have been able to upload sensors data from an Arduino to Google Sheets. Thank you! I have however encountered the following problem: after one hour of logging the authorization is no longer valid. If I restart it works like a charm. I tried to reauthorize at a shorter time interval but I don't seem to get it right!
Any advice?
Thanks again for the tutorial

∧ | ∨ • Share ›

**Razvan Zaharia** → Razvan Zaharia • 2 years ago

Forget that. It's not about one hour or any other time interval, it's just Google not responding some time. I got it working for 7 hours straight and then I got this - File "C:\Python27\lib\site-packages\requests\adapters.py", line 490, in send raise ConnectionError(err, request=request)
requests.exceptions.ConnectionError: ('Connection aborted.', error(10060, 'A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond'))
I guess I just have to ignore these errors and try to connect again.
???

∧ | ∨ • Share ›

| Search | 🔍 |
|---|---|

Build the future of communications. Start today with Twilio's APIs and services.

START BUILDING FOR FREE

POSTS BY STACK

| JAVA | PHP | RUBY | PYTHON | SWIFT | JAVASCRIPT | ARDUINO | .NET |
|---|---|---|---|---|---|---|---|

POSTS BY PRODUCT

## POSTS BY PRODUCT

SMS  AUTHY  VOICE  MMS  VIDEO  SIP  WIRELESS  STUDIO  FLEX  TASK ROUTER

PROGRAMMABLE CHAT  TWILIO CLIENT

## CATEGORIES

Code, Tutorials and Hacks

Customer Highlights

Developers Drawing The Owl

News

Stories From The Road

The Owl's Nest: Inside Twilio

TWITTER                    FACEBOOK

# Developer stories
# to your inbox.

Subscribe to the Developer Digest, a monthly dose of all things code.

Enter your email...

You may unsubscribe at any time using the unsubscribe link in the digest email.
See our privacy policy for more information.

NEW!

## Tutorials

Sample applications that cover common use cases in a variety of languages. Download, test drive, and tweak them yourself.

Get started

SIGN UP AND START BUILDING

Not ready yet? Talk to an expert.