

# Google Spreadsheets and Python



---

by Greg Baugues (<https://www.twilio.com/blog/author/greg-baugues>) on February 16, 2017  
(<https://www.twilio.com/blog/2017/02/an-easy-way-to-read-and-write-to-a-google-spreadsheet-in-python.html>)



Like

Tweet

Follow @twilio

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.

9:52 PM - Jul 4, 2014

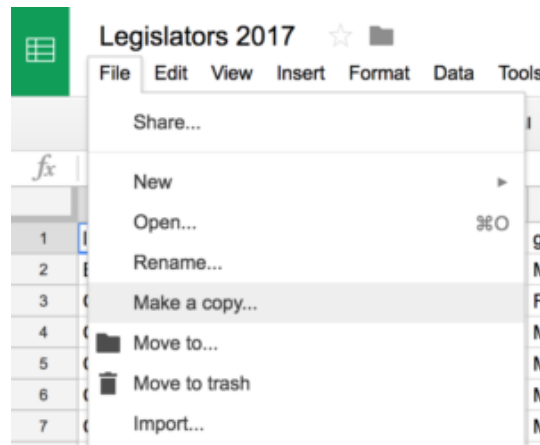
96    39 people are talking about this

In this tutorial, we'll use [Anton Burnashev's \(https://twitter.com/burnash\)](https://twitter.com/burnash) excellent [gspread \(https://github.com/burnash/gspread\)](https://github.com/burnash/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](#)

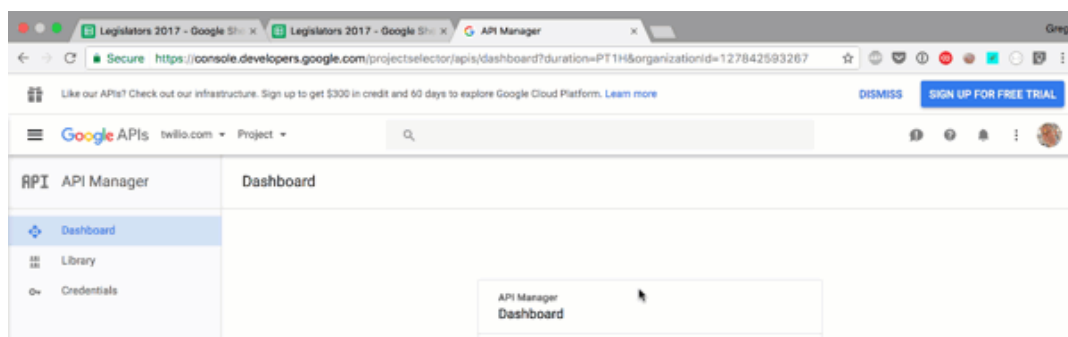
[legislators \(https://docs.google.com/spreadsheets/d/1KK432FFgQ18IP-juSq10ae1ZX1lfa7FTELQFY9bDC38/edit#gid=152787366\)](https://docs.google.com/spreadsheets/d/1KK432FFgQ18IP-juSq10ae1ZX1lfa7FTELQFY9bDC38/edit#gid=152787366)? (Side note: Ian Webster uses this data in conjunction with Twilio to make it easy for citizens to call congress (<https://www.twilio.com/blog/2016/06/call-congress-using-one-phone-number.html>)).

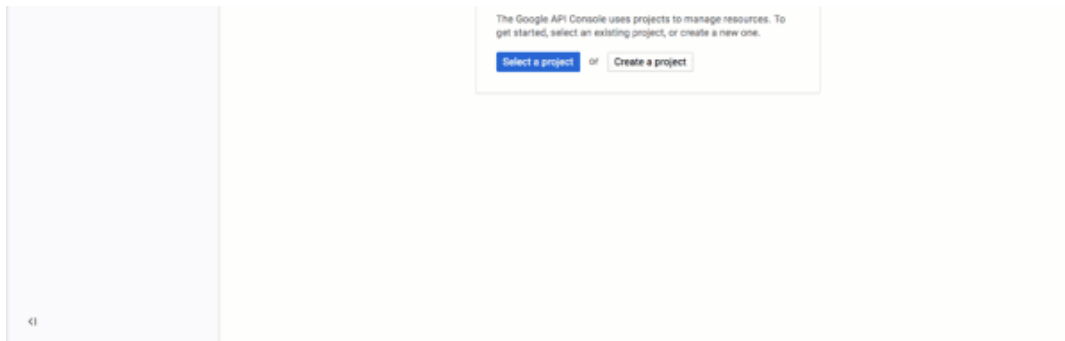


To programmatically access your spreadsheet, you'll need to create a service account and OAuth2 credentials from the [Google API Console \(https://console.developers.google.com/\)](https://console.developers.google.com/). If you've been traumatized by OAuth2 development before, don't worry; service accounts 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 \(https://console.developers.google.com\)](https://console.developers.google.com).
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 `gsread.exceptions.SpreadsheetNotFound` error when you try to access the spreadsheet from Python.

A screenshot of a code editor showing the contents of a file named `client_secret.json`. The file contains a JSON object with the following fields: `type` (service\_account), `project_id` (legislators-157916), `private_key_id` (f3d43d89a5c19ea8a05317270025613ebb5cc8af), `private_key` (-----BEGIN PRIVATE KEY-----\nMIIIEvQIBADANBgkqhkiG9w0BAQEFAA), `client_email` (legislators@legislators-157916.iam.gserviceaccount.com), `client_id` (102880681777284010484), `auth_uri` (https://accounts.google.com/o/oauth2/auth), `token_uri` (https://accounts.google.com/o/oauth2/token), `auth_provider_x509_cert_url` (https://www.googleapis.com/oauth2/v1/certs), and `client_x509_cert_url` (https://www.googleapis.com/robot/v1/metadata/x509/).

```
1 {
2   "type": "service_account",
3   "project_id": "legislators-157916",
4   "private_key_id": "f3d43d89a5c19ea8a05317270025613ebb5cc8af",
5   "private_key": "-----BEGIN PRIVATE KEY-----\nMIIIEvQIBADANBgkqhkiG9w0BAQEFAA",
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
```

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](https://github.com/google/oauth2client) (<https://github.com/google/oauth2client>) – to authorize with the Google Drive API using OAuth 2.0
2. [gspread](https://github.com/burnash/gspread) (<https://github.com/burnash/gspread>) – to interact with Google Spreadsheets

Install these packages with:

```
Python
1 pip install gspread oauth2client
```

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

```
Python
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': '', 'gender': 'M', 'washington_post_id': '', 'youtube': '', 'ballotpedia_id': '', 'youtube_id': ''}, {'cspan_id': '', 'last_name': 'Cheney', 'twitter': '', 'icpsr_id': '', 'wikipedia_id': '', 'thomas_id': '', 'first_name': 'Liz', 'district': 0, 'state': 'WY', 'votesmart_id': '', 'facebook_id': '', 'party': 'Republican', 'type': 'rep', 'rss_url': '', 'lis_id': '', 'govtrack_id': 412732, 'bioguide_id': 'C001109', 'birthday': '7/28/1966', 'facebook': '', 'address': '416 Cannon HOB; Washington DC 20515-5000', 'contact_form': '', 'opensecrets_id': '', 'phone': '202-225-2311', '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 gspread's [well documented](http://gsread.readthedocs.io/en/latest/) (<http://gsread.readthedocs.io/en/latest/>) 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:

Python
1 <code>sheet.get_all_values()</code>

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

Python
1 <code>sheet.row_values(1)</code>
2
3 <code>sheet.col_values(1)</code>
4
5 <code>sheet.cell(1, 1).value</code>

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

Python
1 <code>sheet.update_cell(1, 1, "I just wrote to a spreadsheet using Python!")</code>

Or you can insert a row in the spreadsheet:

Python	
1	<code>row = ["I'm", "inserting", "a", "row", "into", "a", "Spreadsheet", "with", "Python"]</code>
2	<code>index = 1</code>
3	<code>sheet.insert_row(row, index)</code>

You can also delete a row from the spreadsheet:

Python	
1	<code>sheet.delete_row(1)</code>

And find out the total number of rows:

Python	
1	<code>sheet.row_count</code>

Check the [gsread](http://gsread.readthedocs.io/en/latest/) API reference (<http://gsread.readthedocs.io/en/latest/>) 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](https://www.twilio.com/docs/quickstart/python) (<https://www.twilio.com/docs/quickstart/python>).

If you build something cool, please let me know. You can find me at [gb@twilio.com](mailto:gb@twilio.com) (<mailto:gb@twilio.com>) or [@greggyb](http://twitter.com/greggyb) (<http://twitter.com/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 (<https://twitter.com/burnash>) for gsread.*

[Tweet](#)[Follow @twilio](#)

by Greg Baugues (<https://www.twilio.com/blog/author/greg-baugues>) [gbaugues@twilio.com](mailto:gbaugues@twilio.com)  
(<mailto:gbaugues@twilio.com>) [@greggyb](http://twitter.com/greggyb) (<http://twitter.com/greggyb>)

## Build More With Python

SMS and MMS Notifications with Python

(<https://www.twilio.com/docs/tutorials/walkthrough/server-notifications/python/django>)

Tutorials from Twilio: Live Coding on Twitch This Week

(<https://www.twilio.com/blog/2016/03/tutorials-from-twilio-live-coding-on-twitch-this-week.html>)

**(<https://www.twilio.com/blog/2016/03/tutorials-from-twilio-live-coding-on-twitch-this-week.html>)**

Bulk Delete your Twilio Recordings with Python

**(<https://www.twilio.com/blog/2016/05/bulk-delete-your-twilio-recordings-with-python.html>)**

**(<https://www.twilio.com/blog/2016/05/bulk-delete-your-twilio-recordings-with-python.html>)**

How to Build an SMS Slack Bot in Python

**(<https://www.twilio.com/blog/2016/05/build-sms-slack-bot-python.html>)**

**(<https://www.twilio.com/blog/2016/05/build-sms-slack-bot-python.html>)**

Comments for this thread are now closed.



52 Comments Twilio Blog

Login

Recommend 11 Share

Sort by Oldest

**Jed** • a year 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 • Share &gt;

**Jebb** → Jed • a year ago

Looks like the author of gspread plans to move to v4 too [https://github.com/burnash/...](https://github.com/burnash/)

1 • Share &gt;

**Greg Baugues** → Jebb • a year ago

This is good info. Thanks @Jebb.

• Share &gt;

**Bob Belderbos** → Greg Baugues • a year ago

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 &gt;

**Te ENe Te** → Bob Belderbos • 7 months ago

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

• Share &gt;

**Bob Belderbos** → Te ENe Te • 7 months ago

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

• Share &gt;

**pizzapanther** • a year 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 &gt;



Power modern communications. Build the next generation of voice and SMS applications.

**Start Building For Free (<http://twilio.com/try-twilio>)**

## Posts By Stack

---

.NET (<https://www.twilio.com/blog/tag/net>)

Arduino (<https://www.twilio.com/blog/tag/arduino>)

Java (<https://www.twilio.com/blog/tag/java>)

JavaScript (<https://www.twilio.com/blog/tag/javascript>)

PHP (<https://www.twilio.com/blog/tag/php>)

Python (<https://www.twilio.com/blog/tag/python>)

Ruby (<https://www.twilio.com/blog/tag/ruby>)

Swift (<https://www.twilio.com/blog/tag/swift>)

## Posts By Product

---

MMS (<https://www.twilio.com/blog/tag/mms>)

Programmable Chat (<https://www.twilio.com/blog/tag/programmable-chat>)

SIP (<https://www.twilio.com/blog/tag/sip>)

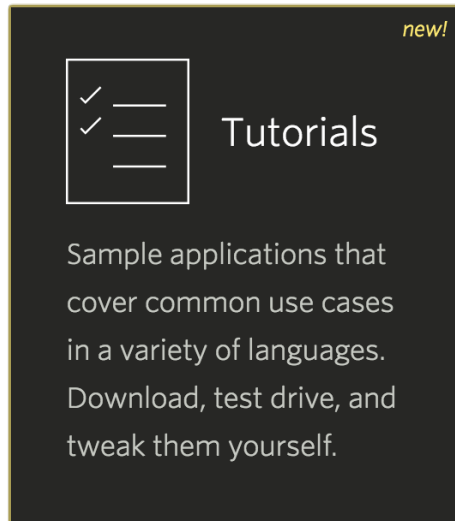
SMS (<https://www.twilio.com/blog/tag/sms>)

Task Router (<https://www.twilio.com/blog/tag/taskrouter>)

Twilio Client (<https://www.twilio.com/blog/tag/twilio-client>)

Twilio Video (<https://www.twilio.com/blog/tag/twilio-video>)

Voice (<https://www.twilio.com/blog/tag/voice>)



(<http://www.twilio.com/docs/tutorials>)

## Categories

---

Code, Tutorials and Hacks (<https://www.twilio.com/blog/all-things-code>)

Customer Highlights (<https://www.twilio.com/blog/customer-highlights>)

Developers Drawing The Owl (<https://www.twilio.com/blog/developers-drawing-the-owl>)

News (<https://www.twilio.com/blog/news>)

Stories From The Road (<https://www.twilio.com/blog/stories-from-the-road>)

The Owl's Nest: Inside Twilio (<https://www.twilio.com/blog/inside-twilio>)

RSS Feed (<http://www.twilio.com/blog/feed>)

**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 the digest email.  
See our [privacy policy \(https://www.twilio.com/legal/privacy\)](https://www.twilio.com/legal/privacy) for more information.

-----

## **COMPANY ([HTTP://WWW.TWILIO.COM/COMPANY](http://www.twilio.com/company))**

About Twilio (<http://www.twilio.com/company>)

Our Nine Values (<http://www.twilio.com/company/nine-values>)

Our Leadership Principles (<http://www.twilio.com/company/leadership-principles>)

The Team (<http://www.twilio.com/company/management>)

Press & Media (<http://www.twilio.com/press>)

Twilio.org (<http://www.twilio.org/>)

Careers (<http://www.twilio.com/company/jobs>)

## **PRODUCTS ([HTTP://WWW.TWILIO.COM/PRODUCTS](http://www.twilio.com/products))**

Voice (<http://www.twilio.com/voice>)

Video (<http://www.twilio.com/video>)

SMS (Text Messaging) (<http://www.twilio.com/sms>)

MMS (Picture Messaging) (<http://www.twilio.com/mms>)

Toll-Free SMS (<http://www.twilio.com/sms/toll-free>)

SIP Trunking (<http://www.twilio.com/sip-trunking>)

Client Mobile (<http://www.twilio.com/client/mobile>)

Client WebRTC (<http://www.twilio.com/webrtc>)

Network Traversal Service (<http://www.twilio.com/stun-turn>)

TaskRouter (<http://www.twilio.com/taskrouter>)

Phone Numbers (<http://www.twilio.com/sms/phone-numbers>)

Short Codes (<http://www.twilio.com/sms/shortcodes>)

Support Plans (<http://www.twilio.com/support-plans>)

Platform (<http://www.twilio.com/platform>)

Lookup (<http://www.twilio.com/lookup>)

Monitor (<http://www.twilio.com/monitor>)

Messaging Copilot (<http://www.twilio.com/copilot>)

Global Conference (<http://www.twilio.com/voice/conference>)

Authy (<http://www.twilio.com/authy>)

## **PRICING ([HTTP://WWW.TWILIO.COM/PRICING](http://www.twilio.com/pricing))**

Voice Pricing (<http://www.twilio.com/voice/pricing>)

Messaging Pricing (<http://www.twilio.com/sms/pricing>)

SIP Trunking Pricing (<http://www.twilio.com/sip-trunking/pricing>)

Client Pricing (<http://www.twilio.com/client/pricing>)

Network Traversal Pricing (<http://www.twilio.com/stun-turn/pricing>)

TaskRouter (<http://www.twilio.com/taskrouter/pricing>)

Lookup (<http://www.twilio.com/lookup#pricing>)

International Coverage (<http://www.twilio.com/international>)

## **USE CASES ([HTTP://WWW.TWILIO.COM/USE-CASES](http://www.twilio.com/use-cases))**

Customer Stories (<http://www.twilio.com/customers>)

Showcase (<http://www.twilio.com/showcase>)

Two-Factor Authentication (<http://www.twilio.com/use-cases/two-factor-authentication>)

Appointment Reminders (<http://www.twilio.com/use-cases/appointment-reminders>)

Dispatch Notifications (<http://www.twilio.com/use-cases/dispatch-notifications>)

ETA Alerts (<http://www.twilio.com/use-cases/eta-alerts>)

Automated Surveys (<http://www.twilio.com/use-cases/automated-surveys>)

Masked Phone Numbers (<http://www.twilio.com/use-cases/masked-phone-numbers>)

Visual Estimates (<http://www.twilio.com/use-cases/visual-estimates>)

Developer Network (<http://www.twilio.com/doers>)

Webinars (<http://www.twilio.com/webinars>)

White Papers (<http://www.twilio.com/white-papers>)

## **API & DOCS ([HTTP://WWW.TWILIO.COM/API](http://www.twilio.com/api))**

API Documentation (<http://www.twilio.com/docs/api>)

Quickstarts (<http://www.twilio.com/docs/quickstart>)

How Tos (<http://www.twilio.com/docs/howto>)

Helper Libraries (<http://www.twilio.com/docs/libraries>)

Security (<http://www.twilio.com/docs/security>)

© 2018 Twilio. All rights reserved. | [Terms of Service \(http://www.twilio.com/legal/tos\)](http://www.twilio.com/legal/tos) |

---

[Privacy Policy \(http://www.twilio.com/legal/privacy\)](http://www.twilio.com/legal/privacy)