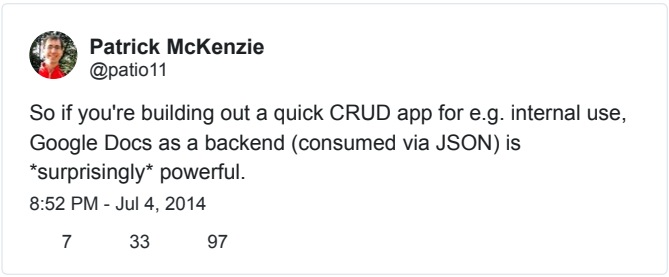
Google Spreadsheets and Python

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)

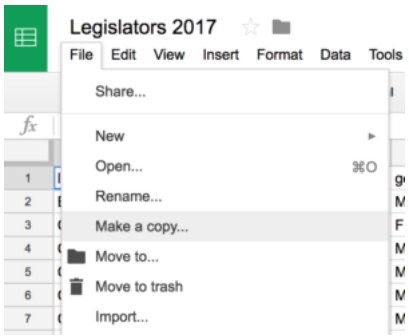
This post is inspired by Patrick McKenzie's reminder that sometimes you don't need a database:



In this tutorial, we'll use Anton Burnashev's (https://twitter.com/burnash) excellent 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

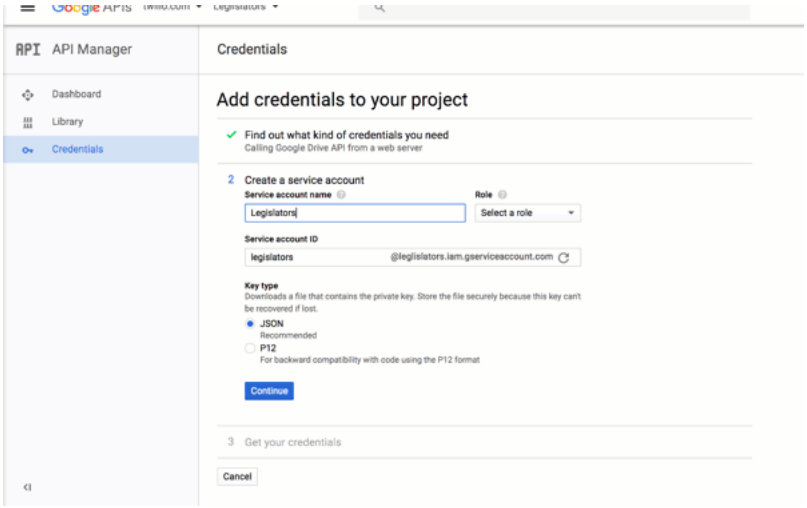
At the risk of being Captain Obvious, you're going to need a spreadsheet if you want to follow along with this post. If you don't have one on hand that's full of juicy data, might I suggest you make a copy of this spreadsheet with contact information for all United States legislators (https://docs.google.com/spreadsheets/d/1KK432FFgQ18lP-juSq10ae1ZX1lfa7FTELQFY9bDC38/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/). 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).

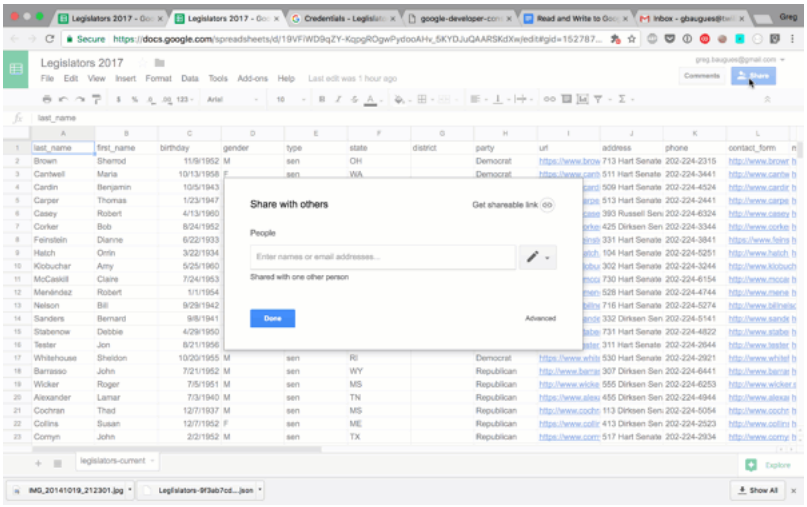
1. Go to the Google APIs Console (https://console.developers.google.com).
2. Create a new project.
3. Click *Enable API*. Search for and enable the Google Drive API.
4. *Create credentials* for a *Web Server* to access *Application Data.*
5. Name the service account and grant it a *Project* Role of *Editor.*
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.



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

Install these packages with:

```
Python
pip install gspread oauth2client
```

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

```
Python
import gspread
from oauth2client.service_account import ServiceAccountCredentials


# use creds to create a client to interact with the Google Drive API
scope = ['https://spreadsheets.google.com/feeds']
creds = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json', scope)
client = gspread.authorize(creds)

# Find a workbook by name and open the first sheet
# Make sure you use the right name here.
sheet = client.open("Copy of Legislators 2017").sheet1

# Extract and print all of the values
list_of_hashes = sheet.get_all_records()
print(list_of_hashes)
```

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

Insert, Update, and Delete from a Spreadsheet with Python

We've just scratched the surface of gspreads' well documented (http://gspread.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
sheet.get_all_values()
```

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

```
Python
```

```
sheet.row_values(1)

sheet.col_values(1)

sheet.cell(1, 1).value
```

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

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

Or you can insert a row in the spreadsheet:

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

You can also delete a row from the spreadsheet:

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

And find out the total number of rows:

```
sheet.row_count
```

Check the gspread API reference (http://gspread.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).

If you build something cool, please let me know. You can find me at gb@twilio.com (mailto:gb@twilio.com) or @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 gspread.

 by Greg Baugues (https://www.twilio.com/blog/author/greg-baugues) gbaugues@twilio.com (mailto:gbaugues@twilio.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)

How to Verify Phone numbers in Python with the Twilio Lookup API

(https://www.twilio.com/blog/2016/02/how-to-verify-phone-numbers-in-python-with-the-twilio-lookup-api.html)

(https://www.twilio.com/blog/2016/02/how-to-verify-phone-numbers-in-python-with-the-twilio-lookup-api.html)

iOS and Web Browser Video Calls with Python and Swift

(https://www.twilio.com/blog/2016/02/ios-and-web-browser-video-calls-with-python-and-swift-2.html)
(https://www.twilio.com/blog/2016/02/ios-and-web-browser-video-calls-with-python-and-swift-2.html)

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)

47 Comments Twilio Blog Login

Recommend 10 Share Sort by Oldest

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Jed • 10 months 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 • Reply • Share ›

Jebb ➚ Jed • 10 months ago

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

1 • Reply • Share ›

Greg Baugues ➚ Jebb • 10 months ago

This is good info. Thanks **@Jebb**.

• Reply • Share ›

Bob Belderbos ➚ Greg Baugues • 9 months 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 • Reply • Share ›

Te ENe Te ➚ Bob Belderbos • 4 months ago

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

• Reply • Share ›

Bob Belderbos ➚ Te ENe Te • 4 months ago

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

• Reply • Share ›

pizzapanther • 10 months 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.

• Reply • Share ›

ZoublWah ➚ pizzapanther • 10 months 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.

• Reply • Share ›

Andrey Fedorov ➚ pizzapanther • 10 months ago

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

wat

Reply • Share ›

Michael R. Bernstein ➚ Andrey Fedorov • 10 months 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 Reply • Share ›

Dilshod • 10 months 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.

Reply • Share ›

Greg Baugues ➚ Dilshod • 10 months ago

@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...

Reply • Share ›

James Abela ➚ Greg Baugues • 10 months 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.

4 Reply • Share ›

Excellion • 10 months ago

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

Reply • Share ›

Alan ➚ Excellion • 10 months 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

Reply • Share ›

Dustin ➚ Alan • 9 months ago

Or use Pandas to save it as Excel.

Reply • Share ›

Chris Emery • 10 months 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.

Reply • Share ›

IssaKnife • 10 months ago

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

Reply • Share ›

James Abela ➚ IssaKnife • 10 months 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)

Reply • Share ›

Jur • 10 months ago

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 identifier of the tab inside the sheet}&format=csv
```

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

1 ^ | v • Reply • Share ›

Jan Oglop ➚ Jur • 9 months ago

Let me add that line separators are "\r\n" ... I've been fighting with it whole day

1 ^ | v • Reply • Share ›

zhangjingqiang • 10 months ago

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

^ | v • Reply • Share ›

Bruno • 10 months 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?

^ | v • Reply • Share ›

Lee • 10 months 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....

^ | v • Reply • Share ›

David Okwii • 10 months 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'

^ | v • Reply • Share ›

Von Foerster ➚ David Okwii • 9 months ago

I've got the same one.
Did you solve it? Can you help?
Thanks for the tutorial!

^ | v • Reply • Share ›

David Okwii ➚ Von Foerster • 9 months 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.

^ | v • Reply • Share ›

Gery Greyhound O1G • 8 months 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?

^ | v • Reply • Share ›

John Mathews • 8 months ago

Thanks for this. Super useful and clear!

^ | v • Reply • Share ›

Ron Serruya • 7 months ago

Thanks ! This helped me a lot

Reply • Share ›

Manikandan K • 7 months ago

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

1 Reply • Share ›

Brandon Moore • 6 months ago

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. :)

Reply • Share ›

Wenhao Xuan • 6 months ago

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

Reply • Share ›

Sunny ke • 6 months 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

Reply • Share ›

gif ➚ Sunny ke • a month ago

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

Reply • Share ›

Fernando Simao • 6 months ago

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

Reply • Share ›

Erik • 5 months ago

Get an error when trying to import ServiceAccountCredentials:

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

Reply • Share ›

Curtis Salisbury ➚ Erik • 4 months ago

I am getting the same error as well.

Reply • Share ›

omer • 4 months ago

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?

Reply • Share ›

Charles Cuthbert ➚ omer • 3 months ago

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

Other. 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

Reply • Share ›

Oliv698 ➚ Charles Cuthbert • 3 months 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?

Reply • Share ›

Oliv698 ➚ Oliv698 • 3 months ago

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

Reply • Share ›

Steve Shaw • 4 months ago

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 Reply • Share ›

Razvan Zaharia • 4 months 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

Reply • Share ›

Razvan Zaharia ➚ Razvan Zaharia • 3 months 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.
???

Reply • Share ›

Kyaw Thura Maung • a month ago

Hello!

Reply • Share ›

sidharth bolar • 11 days ago

This is very helpful
Thanks!

Reply • Share ›

Search ...

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)

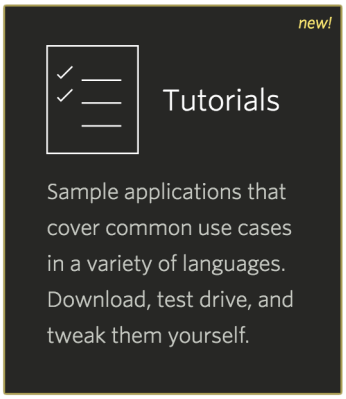
Posts By Product

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)

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)

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)

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)

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)

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