



Open in app

Get started



Published in Towards Data Science

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)



Yong Cui

Follow

Sep 30, 2021 · 7 min read ★ · Listen

Save



Deploy a Public Streamlit Web App for Free — Here's How

Google Sheets as its backend and hosted by Streamlit Sharing



Photo by [Tetiana Syrova](#) on [Unsplash](#)





Open in app

Get started

development requires Python users' sufficient familiarity with relatively complicated web frameworks, such as Flask and Django. Although it's not the hardest thing for any experienced Python users to learn these frameworks, it might be an overkill for data scientists to showcase their data science projects.

Fortunately, we data scientists have a better option now — the [Streamlit](#) framework. It simply converts simple Python scripts to functional, interactive, and shareable web apps within minutes. Because it's designed toward showcasing data science products, it natively supports the data models and graphs with major data science libraries, such as pandas and matplotlib. Thus, it requires little learning curve if you already know Python well. If you've not tried that, here's a couple of articles that I published before to get you started with streamlit.

Build Your First Interactive Data Science Web App With Streamlit

Highlighting some of the most common features

towardsdatascience.com

8 Simple and Useful Streamlit Tricks You Should Know

Write better Streamlit apps

towardsdatascience.com

In this article, I want to take you guys a step further by showing you how to publish your web app such that others can access your data science projects publicly. More importantly, I'll show you how to integrate Google Sheets as a free database host for our project. Please note that Google Sheets isn't an industry-level database, but it's good enough to host a small dataset. What's more — it's free.



[Open in app](#)[Get started](#)

Step 1. Build Web App Locally

First, I want to make sure that you've installed streamlit on your computer and you're able to launch a web app locally.

1.1. Installation of Streamlit

You can install the streamlit framework using the pip tool: `pip install streamlit`.

You can check if it's installed by simply viewing the version: `streamlit --version`. This command should tell you what version the installed streamlit package is. The version information displayed is `Streamlit, version 0.89.0` while I'm writing this tutorial.

1.2. Create the App's Script

Create a Python script file and let's simply call it `streamlit_sharing.py`. Because the goal of this tutorial is to show you how to publish the web app, we'll keep the stuff in the app minimum. Suppose that the script includes the following lines of code.

```
1 import streamlit as st
2 import pandas as pd
3
4 st.write("My First Streamlit Web App")
5
6 df = pd.DataFrame({"one": [1, 2, 3], "two": [4, 5, 6], "three": [7, 8, 9]})
7 st.write(df)
```

streamlit_sharing_part1.py hosted with ❤ by GitHub

[view raw](#)

Streamlit Sharing — The Basic Setup

1.3. Launch the App Locally

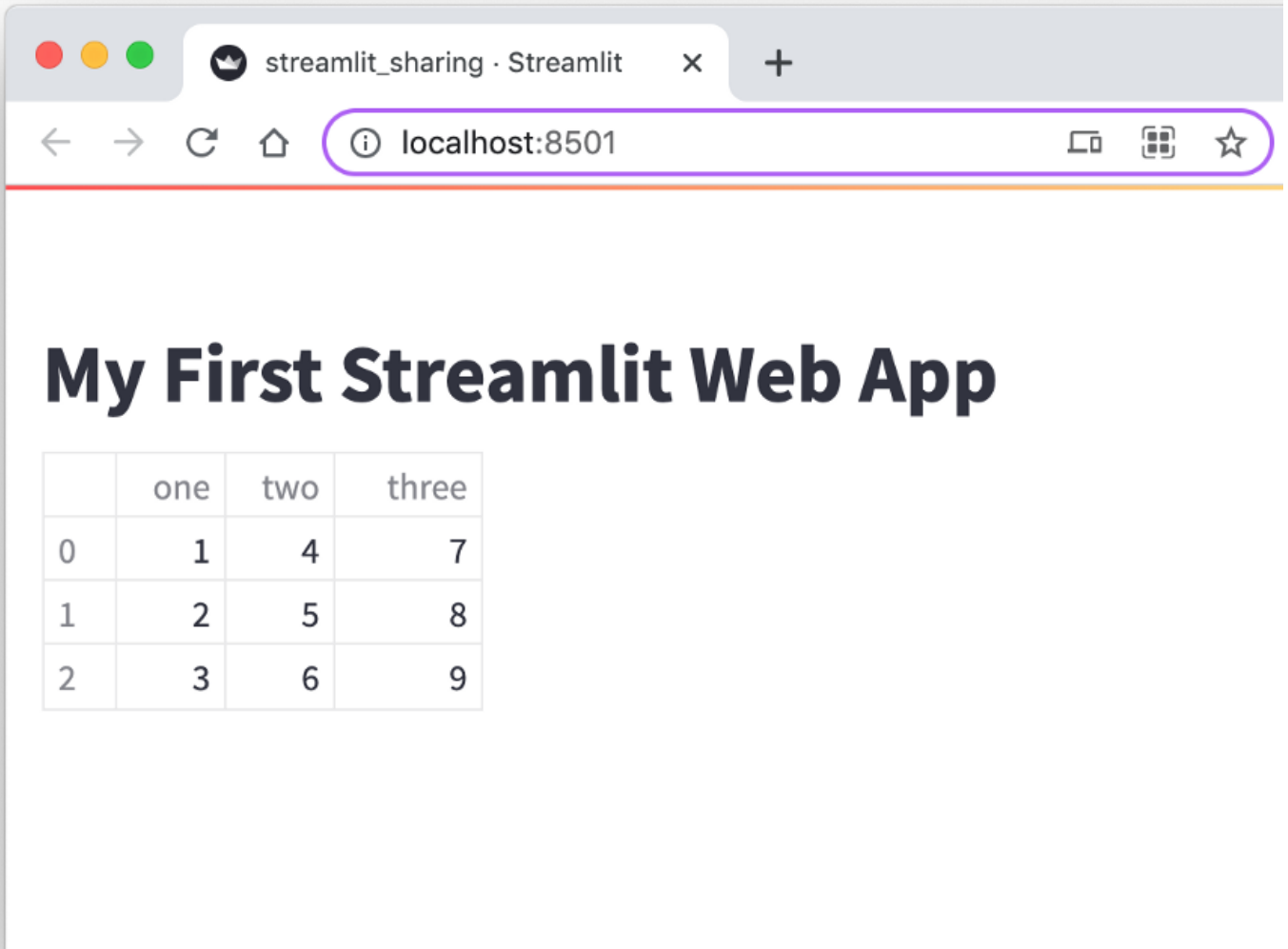
To launch the app locally, in the terminal, you can run the following command: `streamlit run streamlit_sharing.py`. Please make sure that you need to navigate to the directory where the Python script is saved. Otherwise, you'll have to specify the full path to the file.





Open in app

Get started



Local Streamlit Web App

The above screenshot shows you the web app running on your local host. Great! You've simply launched your first Streamlit web app.

Step 2. Connect to a Google Sheet

We should understand that Streamlit supports the connection to many different kinds of database, but online database hosts are not typically free. To provide you with a proof of concept, we'll use a public Google Sheet as the data source for our web app. Suppose that our sheet has the following data.





Open in app

Get started

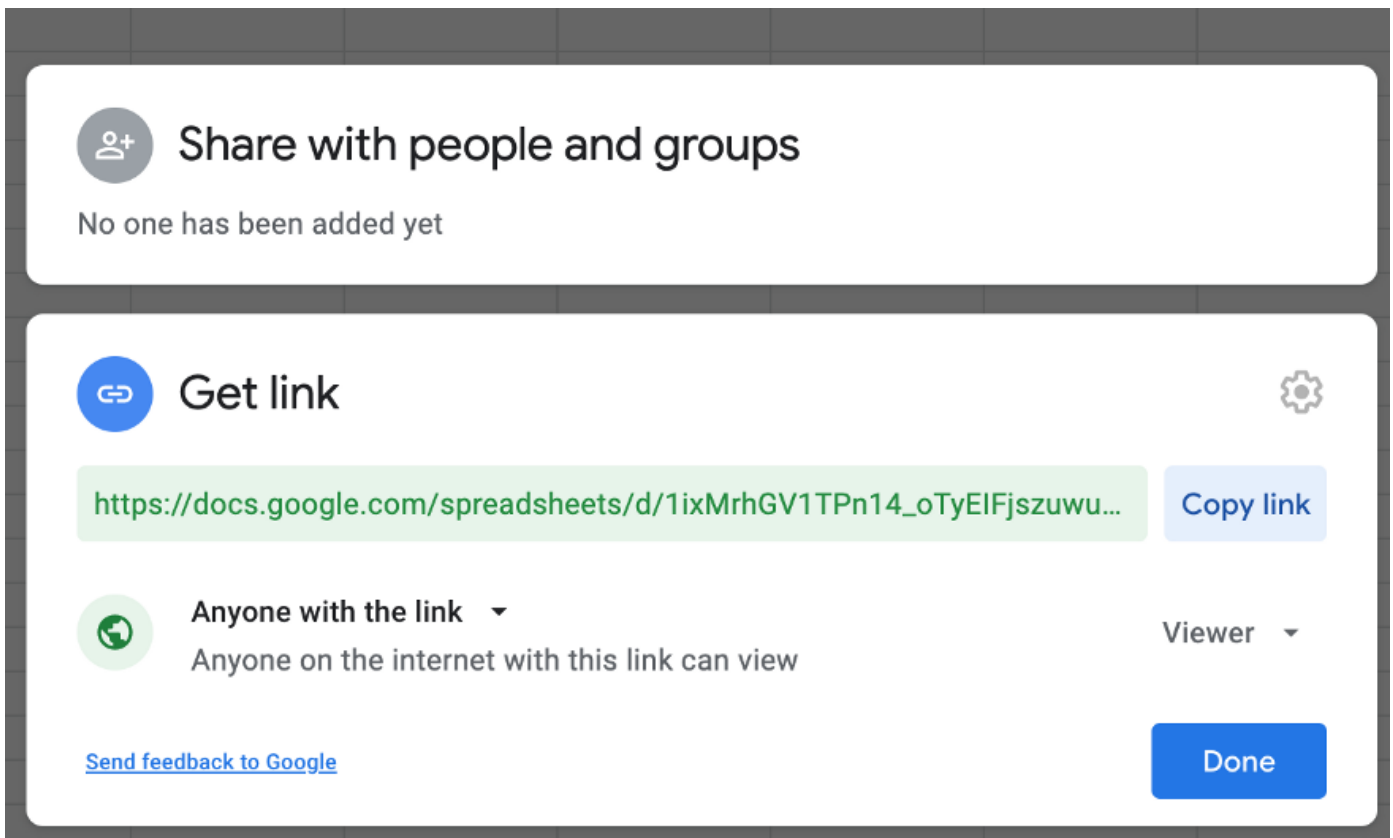
2	Dog	1000	03/02/2020
3	Cat	1500	04/01/2020
4	Tiger	1740	05/08/2020
5	Lion	1302	08/02/2020

google_sheets_data.csv hosted with ❤️ by GitHub [view raw](#)

Data in the Google Sheet

2.2. Share the Sheet

Once you create the Google sheet, please turn on sharing publicly such that our web app can access it directly.



Enable Public Sharing

IMPORTANT: To keep the app simple, we'll only consider the set-up that uses a public Google Sheet. If you want to use a private Google Sheet, it requires enabling Google APIs





Open in app

Get started

To access the Google sheet programmatically, we need to install the package `gsheetsdb` by running `pip install gsheetsdb`. Once it's installed, we can update our script to the following version:

```
1  import streamlit as st
2  import pandas as pd
3  from gsheetsdb import connect
4
5  st.title("My First Streamlit Web App")
6
7  df = pd.DataFrame({"one": [1, 2, 3], "two": [4, 5, 6], "three": [7, 8, 9]})
8  st.write(df)
9
10 st.title("Connect to Google Sheets")
11 gsheet_url = "https://docs.google.com/spreadsheets/d/1ixMrhGV1TPn14_oTyEIFjszuuw09xkbsc1WEBJH3N0/"
12 conn = connect()
13 rows = conn.execute(f'SELECT * FROM "{gsheet_url}"')
14 df_gsheet = pd.DataFrame(rows)
15 st.write(df_gsheet)
```

streamlit_sharing_part2.py hosted with ❤ by GitHub

[view raw](#)

Integrate Google Sheet

The above code involves the following changes:

- We use the `gsheetsdb` for connecting to the Google sheet.
- To use the sheet, we use a SQL-like syntax to retrieve the data. Notably, instead of specifying the sheet's name, we need to specify the URL to the sheet. Don't forget about the double quotes for the sheet's URL.
- We create a `DataFrame` object from the retrieved rows.

With all these changes, our web app looks like the following now. As you can see, we're indeed able to retrieve the data from the created Google sheet.





Open in app

Get started



streamlit_sharing · Streamlit



localhost:8501



My First Streamlit Web App

	one	two	three
0	1	4	7
1	2	5	8
2	3	6	9

Connect to Google Sheets

	user	followers	registration_date
0	Dog	1,000.0000	2020-03-02
1	Cat	1,500.0000	2020-04-01
2	Tiger	1,740.0000	2020-05-08
3	Lion	1,302.0000	2020-08-02

Updated Web App

Step 3. Deploy the App Publicly

3.1. Create a Streamlit Share Account

There are different options to host a Streamlit app, such as Heroku (You can find a tutorial in a [Medium article](#)). Here, I want to show you to use Streamlit Share — a free Streamlit





Open in app

Get started

1. Because the Streamlit team is still developing the platform, you need to request an invite to be able to deploy the app on Streamlit Share. You can submit a request here: <https://streamlit.io/cloud>
2. Once you submit your request, it won't take too long for them to approve your request. In my case, it took less than one day to receive the invite.
3. You can now login into the Streamlit Share (<https://share.streamlit.io/>) with your GitHub authentication.

3.2. Update Your Script File

As you'll see next, we'll share our app in a public GitHub repository, and thus we don't want to expose the Google sheet URL, which can be a potential security concern. Certainly, using a private Google Sheet or other databases with authentications is more secure.

Instead of specifying the URL in the script directly, you'll save the URL in a shared app's settings, from which you can access using the setting's name.

```
# Original
gsheet_url = "the_link"


# Updated, when the URL is saved in the settings
gsheet_url = st.secrets["public_gsheets_url"]
```

3.3. Create a Public GitHub Repository

Once you update your script, it's finally the time to make your app public (to some extent). What you'll do is to create a public GitHub repository. I'm not going to expand how you can do that here. I use PyCharm, and I just need to use the built-in Git tool to create a repository easily.

3.4. Share the Web App!





Open in app [Get started](#)

[← Back](#)

Deploy an app

Apps are deployed directly from their GitHub repo. Enter the location of your app below.
Or [click here to fork and deploy a sample app](#).

Repository

[Paste GitHub URL](#)

user/repo

Branch

master

Main file path

streamlit_app.py

[Advanced settings...](#)

Deploy!

←

Advanced settings

×

Python version

3.9

Secrets

Provide environment variables and other secrets to your app using TOML format. This information is encrypted and served securely to your app at runtime. Learn more about Secrets in our [docs](#). Changes take around a minute to propagate.

```
public_gsheets_url = "the_link"
```

Save

If you've already linked your GitHub account, Streamlit is smart enough to pull out the list of repositories and you can simply choose the one and specify the file path.

Importantly, you click the “Advanced settings” and specify the link there, because as mentioned previously, this is how our streamlit script file accesses the confidential information. When needed, you need to choose a proper Python version for your project, because by default, it's set to 3.7.

After that, click the “Deploy!” button, and wait for a couple of moments, and you should be able to see your app is up and running!





Open in app

Get started

share.streamlit.io/ycui1/medium_python_tutorials/streamlit_sharing_tutorial/streamlit_sharing.py

My First Streamlit Web App

	one	two	three
0	1	4	7
1	2	5	8
2	3	6	9

Connect to Google Sheets

	user	followers	registration_date
0	Dog	1,000.0000	2020-03-02
1	Cat	1,500.0000	2020-04-01
2	Tiger	1,740.0000	2020-05-08
3	Lion	1,302.0000	2020-08-02

Public Web App

Because it's hosted by Streamlit Share as a public website, you can share the link to your teammates or clients and they can access it from anywhere they want.

Final Words

Streamlit has made web app developments easy for data scientists, because it takes care of the layout of the web elements, and we data scientists are just responsible for the core part of our app — the data.



[Open in app](#)[Get started](#)

Thanks for reading this article. Stay connected by [signing up my newsletter](#). Not a Medium member yet? Support my writing by [using my membership link](#).

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

[Get this newsletter](#)

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

