

Zapier Import Integration

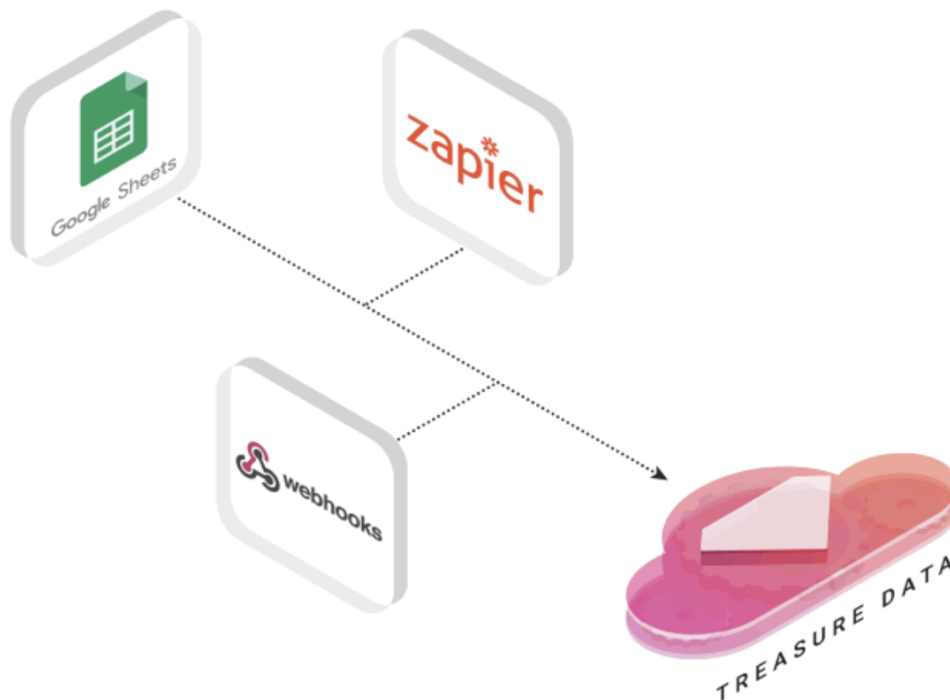
If you're familiar with IFTTT.com and the simple "If This Then That" task automation "recipes", then you already also have a pretty good idea of how Zapier's trigger-response mechanism works: you perform a simple manipulation in one web app, and this triggers a response or behavior in another.

Using webhooks, you can do a simple data import to Treasure Data. Given Zapier's hundreds of integrations, a "Zap" containing a webhook to Treasure Data's REST API opens the door to numerous self-service integrations between any of these data sources and Treasure Data.

While this integration should not replace a full-fledged import to Treasure Data's Data Engineering as a Service using one of our Data Connectors, it does serve as a good basis for modeling your data or building exploratory queries and analysis on data sources that are not yet supported.

Are you working in an organization where some folks are still keeping data in spreadsheets and want a quick and easy way to use their data in your analytics pipeline?

This tutorial will demonstrate this with a simple integration between Google Sheets and Treasure Data using a Zapier Zap.



- [Prerequisites](#)
- [Set up your Google Sheet](#)
- [Set up your Treasure Data Database and Table](#)
- [Set up your Zapier Zap](#)
- [Query your Data](#)
- [More Integration Options](#)

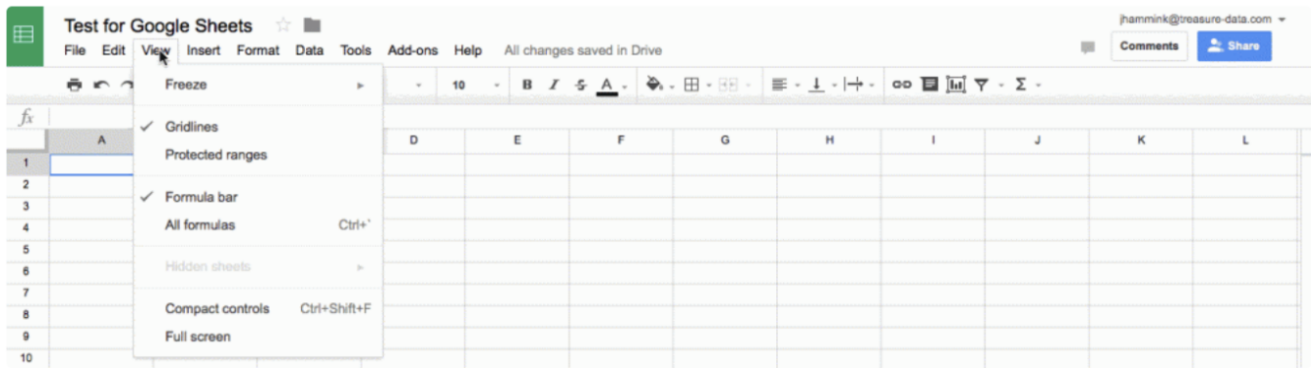
Prerequisites

- A [Google Docs](#) account.
- A [Treasure Data account](#) and your Treasure Data Write API key. We recommend you also install Treasure Data Toolbelt.
- A [Zapier](#) account. For our purposes, the free basic plan will do.

Set up your Google Sheet

Set up a Google Sheet as you normally would (for use as a data table). Be sure to include a header row at the top.

Select the GIF to view the animation.

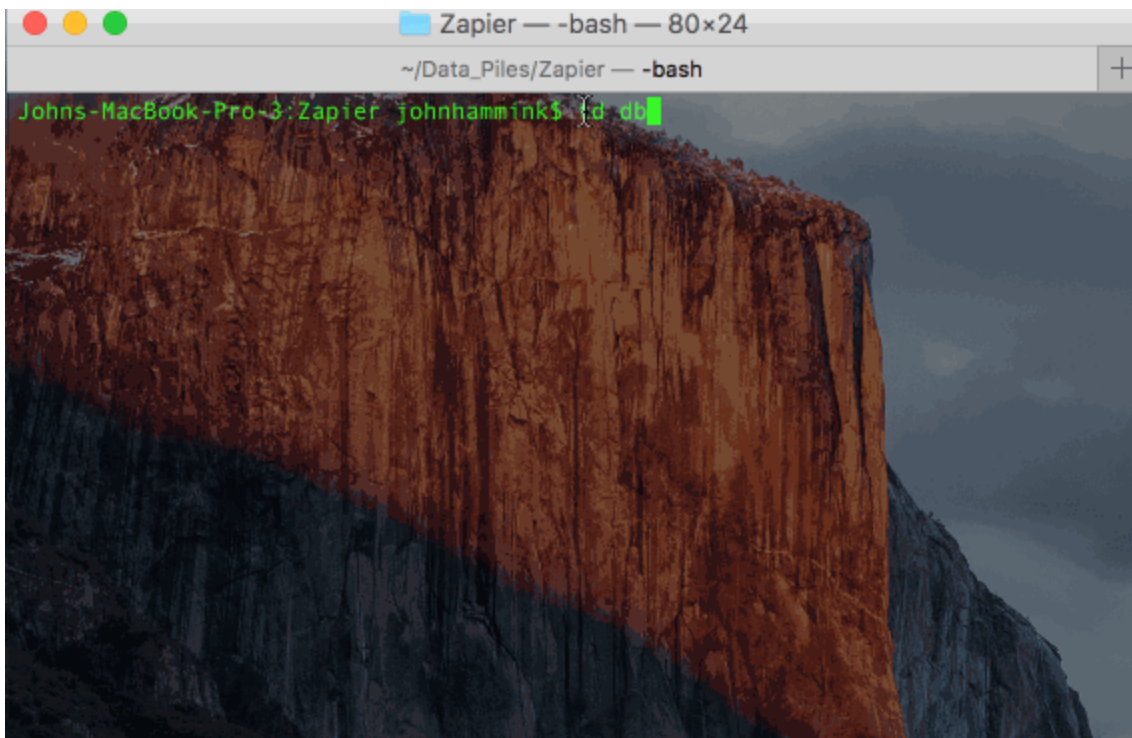


Set up your Treasure Data Database and Table

This is most easily done from the command line using TD Toolbelt.

```
td db:create <database_name>td table:create <database_name> <table_name>  
view rawCreate DB and Table (generic) hosted with by GitHub
```

https://blog.treasuredata.com/wp-content/uploads/2016/04/td_create_db_and_table.gif



Set up your Zapier Zap

1. At the top of the Zapier Dashboard, select

MAKE A ZAP!

2. Choose Google Sheets as your trigger app.



Google Sheets

3. Choose New Spreadsheet Row as your trigger and select Save and Continue.



Select Google Sheets Trigger

Search Google Sheets Triggers...



New Spreadsheet Row

Triggered when a new row is added to the bottom of a spreadsheet.



Updated Spreadsheet Row

Triggered when a new row is added or modified in a spreadsheet.

[show less common options](#)

Continue

244px200

1. Select your Google Sheets account. Select Test and Save and Continue.
2. Select the Spreadsheet and Worksheet you want to pull data from. Select Save and Continue.
3. On the next screen, after you've added a row of data to your spreadsheet, select Fetch + Continue.
4. Next, select Webhooks as your Action app.

5. Choose Post action. Hit Save + Continue.
6. Enter the following information

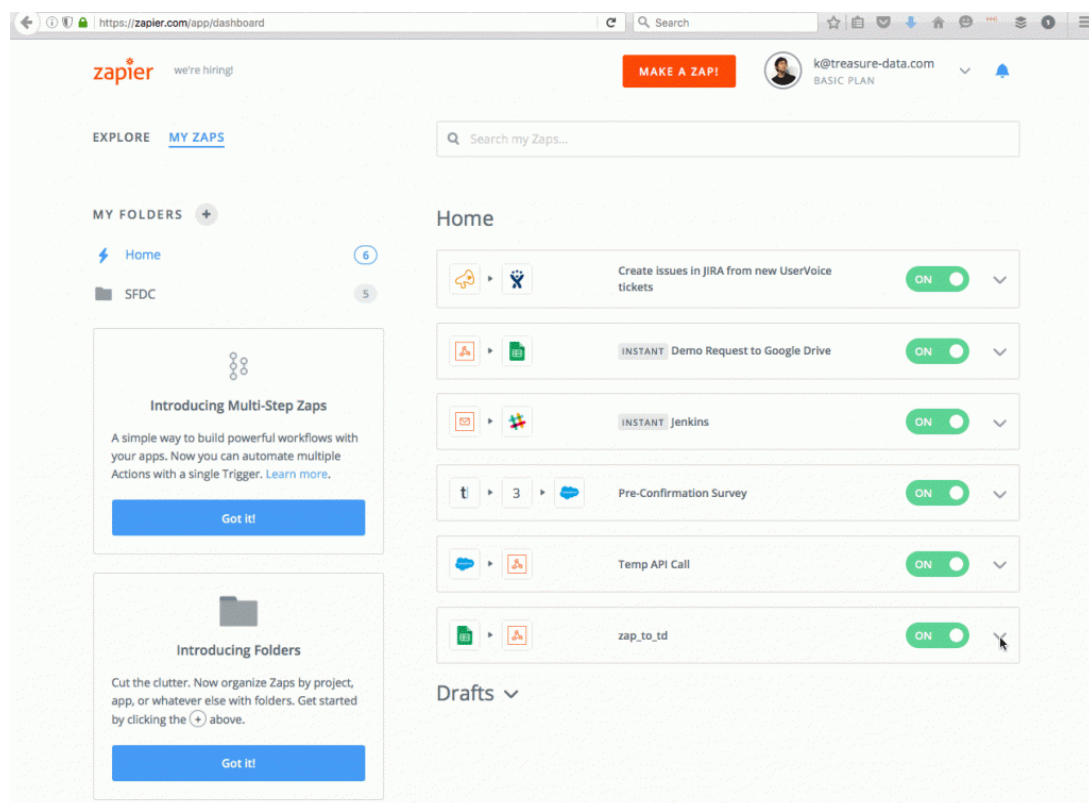
```
URL: http://in.treasuredata.com/postback/v3/event/<database_name>/<table_name>Payload Type: JsonWrap Request
in Array: NoUnflatten: yesHeaders:Content-Type: application/jsonX-TD-Write-Key:
<your_Treasure_Data_Write_API_Key>
view rawwebhook post hosted with by GitHub
```

1. Select Continue, add a new row to your spreadsheet and test it in the next step by selecting Create & Continue.
2. Select Finish and turn the Zap on. You can view all Zaps with the See it in your Dashboard button.



It takes up to 15 minutes for Zapier to recognize new rows in the sheet and run the webhook, although in some cases you can easily prompt the Zap to run by selecting the dropdown menu.

Select the GIF to view the animation.



Query your Data

Once your data is in Treasure Data (it was ingested when the webhook ran), you can query it using Treasure Data CLI and SQL. Our SQL engine is powered by Hive and Presto and scales to billions of rows.

The column headers won't match exactly to those in your original spreadsheet. Zapier adds "gsx_" prefix to the headers you've defined in your original sheet, as well as a few other Zapier-specific fields.

However, you can get the same data out of your query as on your original sheet by excluding the Zapier-added columns in your queries.

There are a few different ways you can do this from the CLI. You can run the query as a plaintext string:

```
td query -d googledocs "select gsx_userid, gsx_lastname, gsx_streetname, gsx_city, gsx_zip from sheet"
view rawquery_googledocs hosted with by GitHub
```

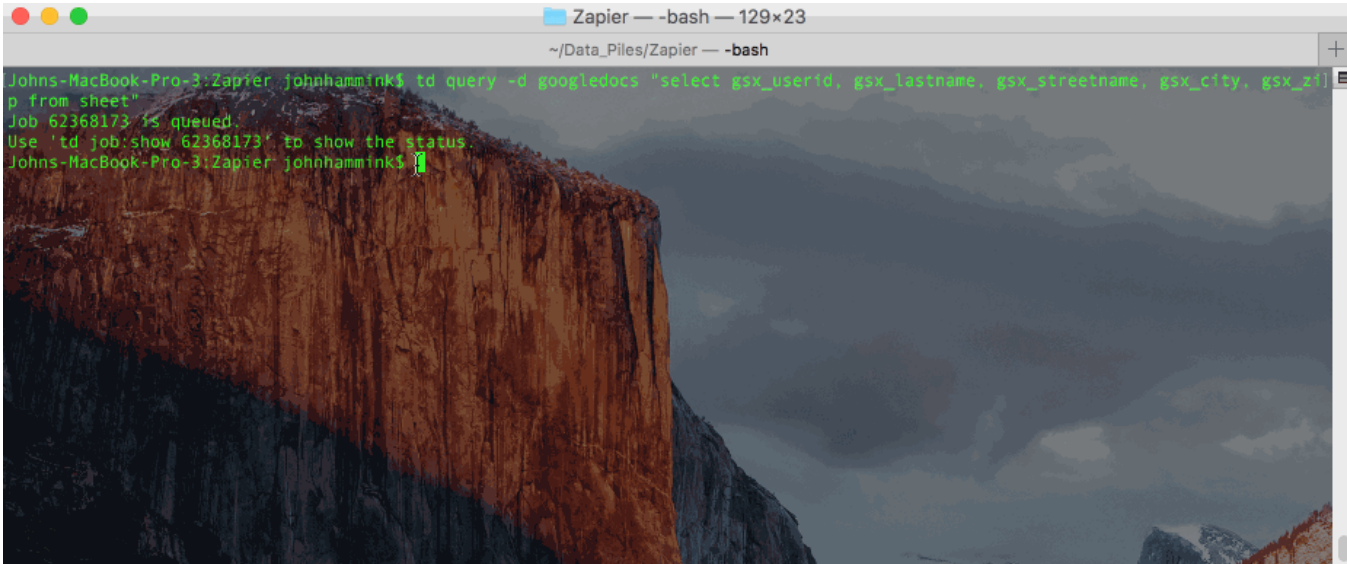
Or, if you prefer, you can also store the query itself in a text file, and call it from the CLI (accessing results by job ID, as above).

query.txt

```
select gsx_userid, gsx_lastname, gsx_streetname, gsx_city, gsx_zip from sheet  
view rawquery.txt hosted with by GitHub
```

Run the query from the command line:

```
td query -d googledocs -q query.txt  
view rawTD CLI Query from textfile hosted with by GitHub
```

A screenshot of a terminal window titled "Zapier -- -bash -- 129x23" with a background image of a mountain range. The terminal shows the following text:

```
Johns-MacBook-Pro-3:Zapier johhamminK$ td query -d googledocs "select gsx_userid, gsx_lastname, gsx_streetname, gsx_city, gsx_zip  
p from sheet"  
Job 62368173 is queued.  
Use 'td job:show 62368173' to show the status.  
Johns-MacBook-Pro-3:Zapier johhamminK$
```

More Integration Options

Today, we showed you a Zapier Zap that pulls data from Google Sheets and sends it to Treasure Data, but it's quite trivial to build Zaps to bring in Data from numerous other sources.

Check out Zapier's [Zapbook](#) for the full list of available Zapier integrations (and possible data sources).

You can even use SQL (via Hive or Presto) to periodically run an inner join on two separate tables imported from separate sources (assuming they share a common field) as you would in any other relational database. Try it out!