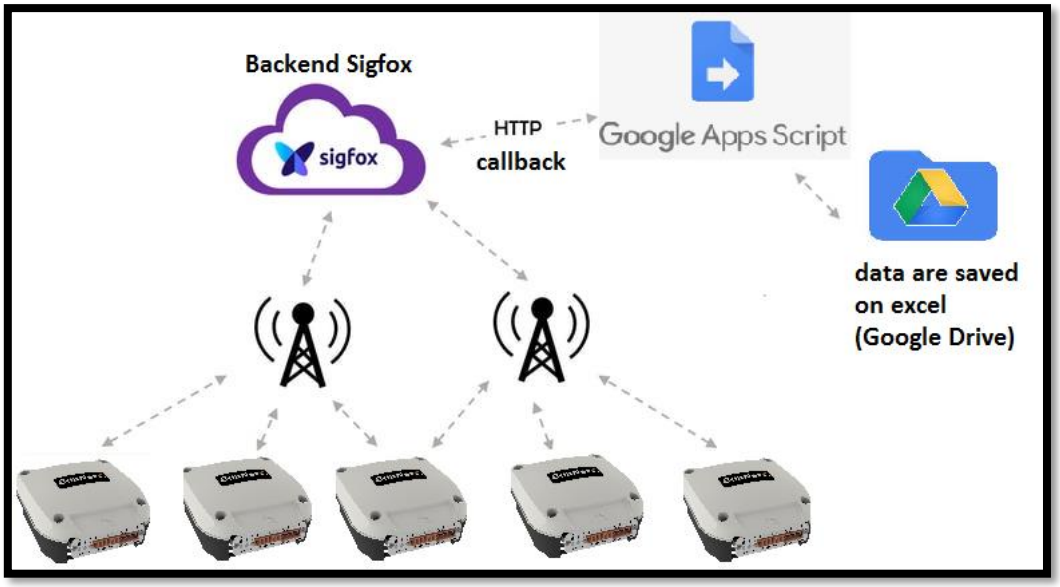
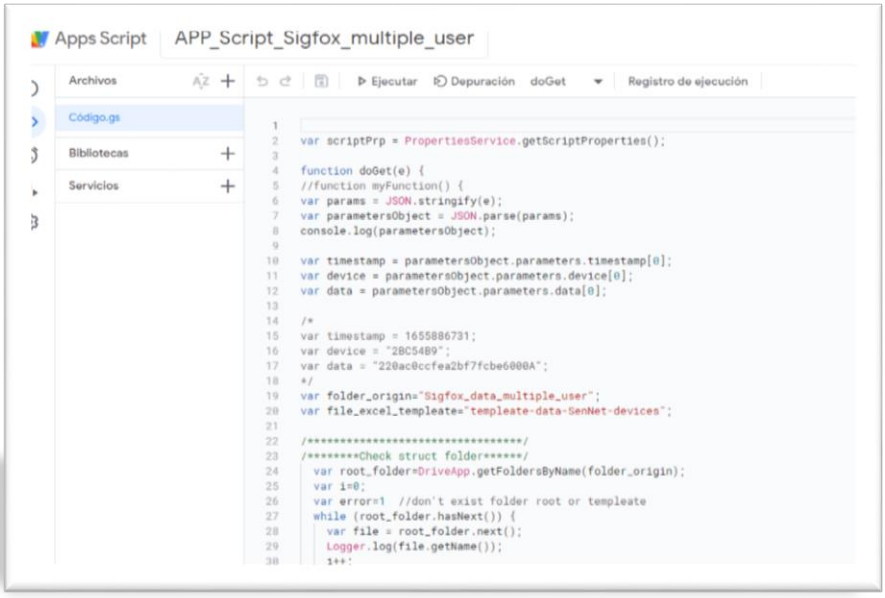
SenNet IoT Sigfox devices saving data on Google Drive

How use Google Apps Script to develop a script to save our data belong a SenNet IoT Sigfox devices.

Google Apps Script is a very powerful and freely feature belong to Google.

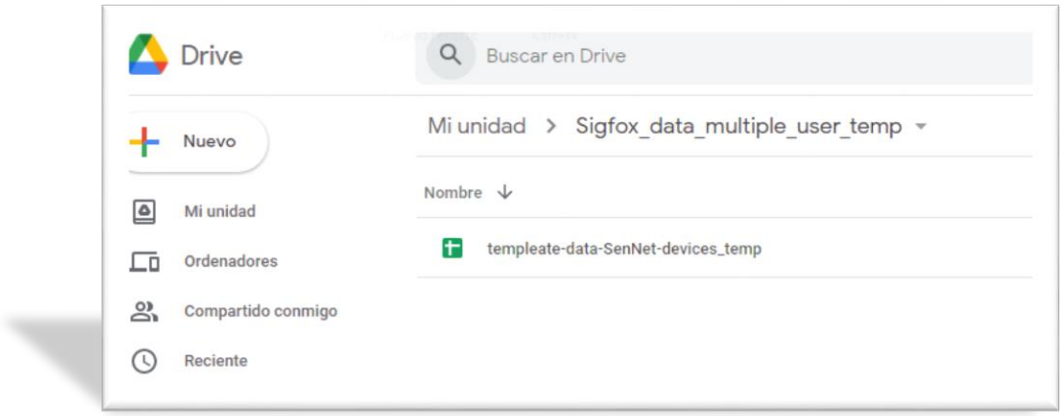


- Create Google Apps Script on your google account.
- Copy code of SenNet Apps Script parser, belong to Satel Spain.

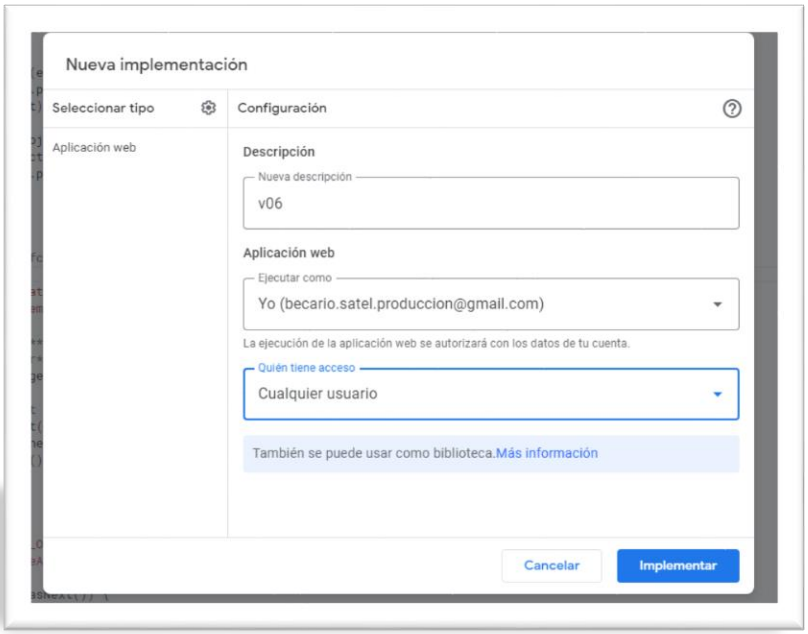


Inside of this script you must define an folder where is going to save data/excel, and templeate excel that it'll use to copy data from devices with right format.

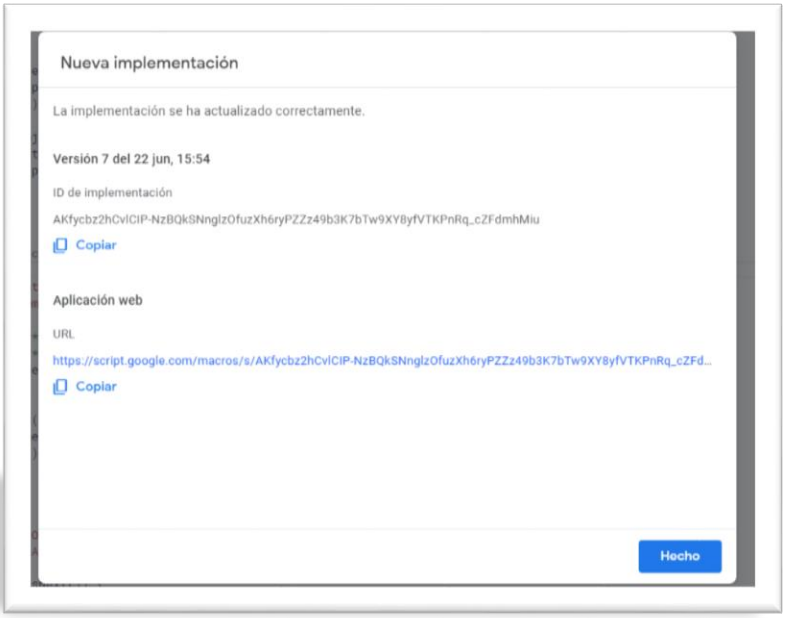
```
var folder_origin="Sigfox_data_multiple_user_temp";
var file_excel_templeate="templeate-data-SenNet-devices_temp";
```



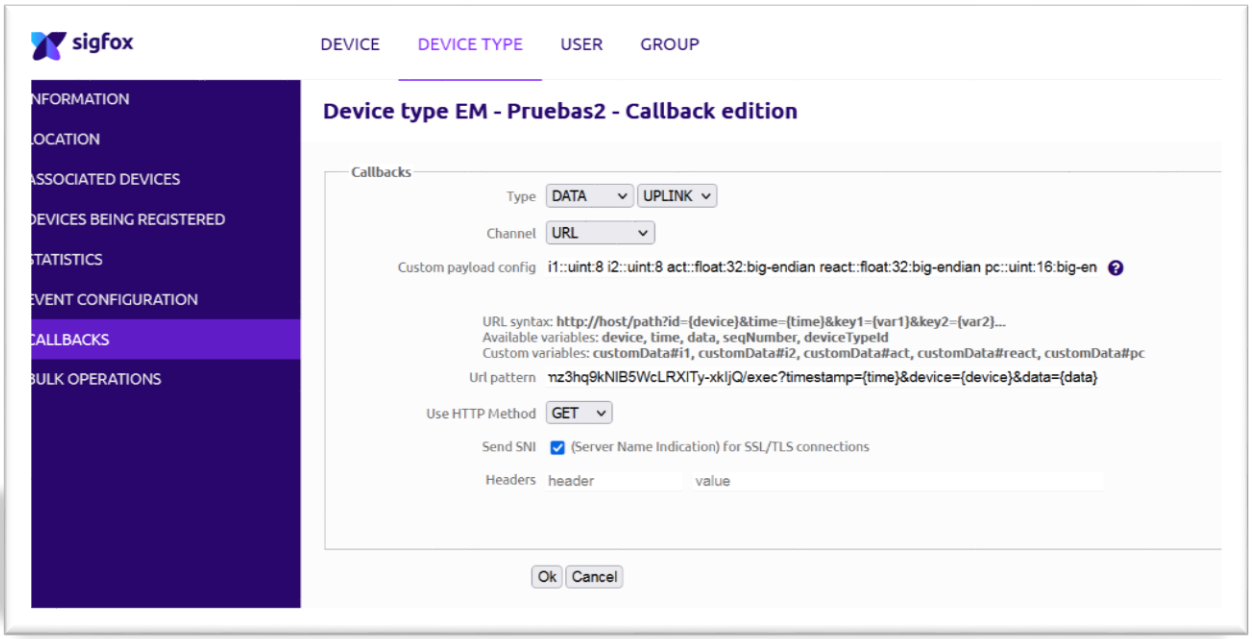
- Define a new implementation



- Give all right to execute this Script to other user outside of your google account.
- Copy URL link

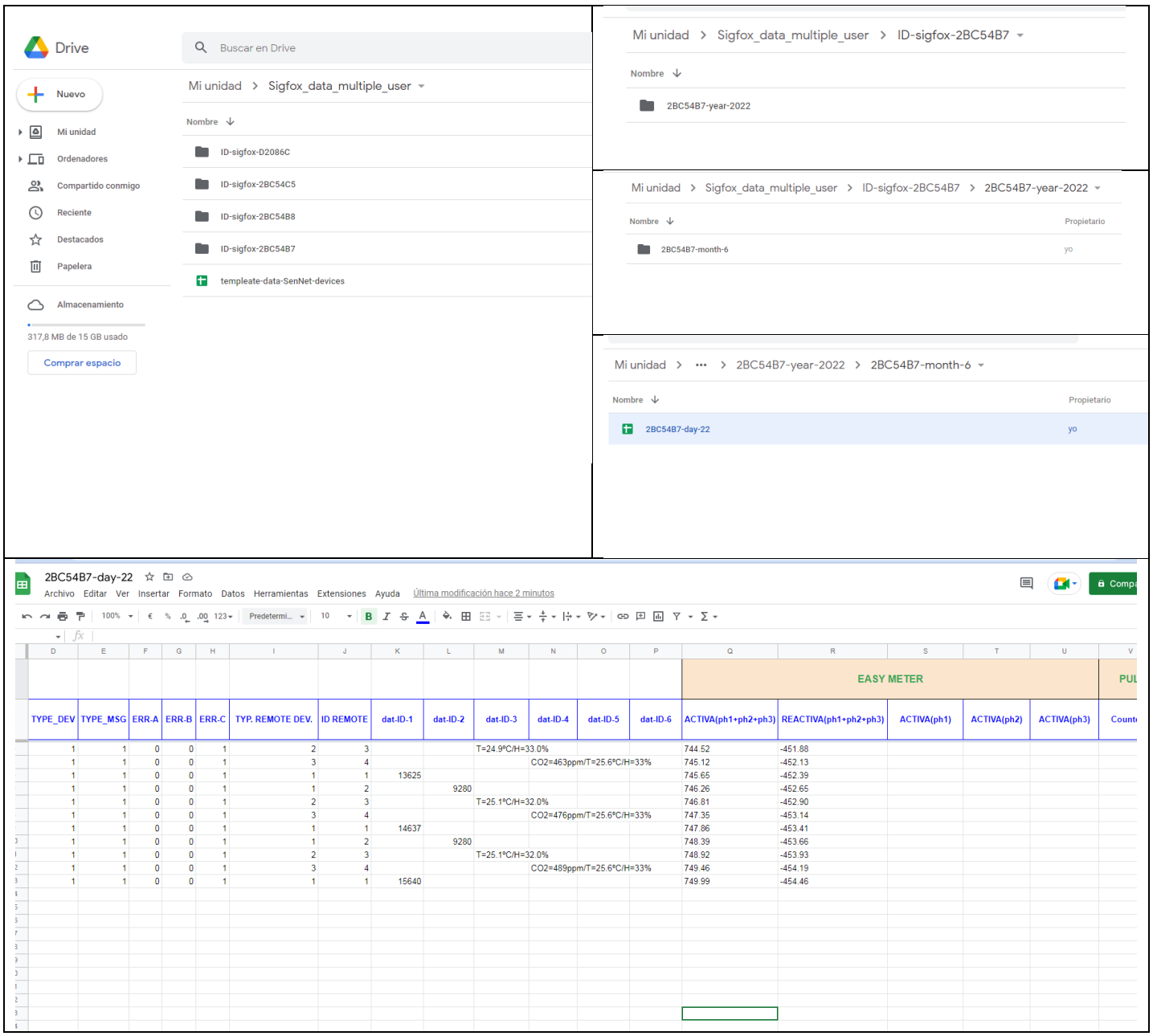


- Go to backend website , and define custom callback for your type of device, you must add these fields:



"ULR APP_web" + ?timestamp={time}&device={device}&data={data}

- This is the sctuct that will be define by Apps Script.



2BC54B7-day-22

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda Última modificación hace 2 minutos

													EASY METER					
TYPE_DEV	TYPE_MSG	ERR-A	ERR-B	ERR-C	TYP. REMOTE DEV.	ID REMOTE	dat-ID-1	dat-ID-2	dat-ID-3	dat-ID-4	dat-ID-5	dat-ID-6	ACTIVA(ph1+ph2+ph3)	REACTIVA(ph1+ph2+ph3)	ACTIVA(ph1)	ACTIVA(ph2)	ACTIVA(ph3)	Count
1	1	0	0	1	2	3			T=24.9ºC/H=33.0%				744.52	-451.88				
1	1	0	0	1	3	4				CO2=463ppm/T=25.6ºC/H=33%			745.12	-452.13				
1	1	0	0	1	1	1	13625						745.65	-452.39				
1	1	0	0	1	1	2		9280					746.26	-452.65				
1	1	0	0	1	2	3			T=25.1ºC/H=32.0%				746.81	-452.90				
1	1	0	0	1	3	4				CO2=476ppm/T=25.6ºC/H=33%			747.35	-453.14				
1	1	0	0	1	1	1	14637						747.86	-453.41				
1	1	0	0	1	1	2		9280					748.39	-453.66				
1	1	0	0	1	2	3			T=25.1ºC/H=32.0%				748.92	-453.93				
1	1	0	0	1	3	4				CO2=489ppm/T=25.6ºC/H=33%			749.46	-454.19				
1	1	0	0	1	1	1	15640						749.99	-454.46				

Note:

Current limitations

Use the limits below to help test your scripts. All limits are subject to elimination, reduction, or change at any time, without notice.

Feature	Consumer (e.g., gmail.com) and G Suite free edition (legacy)	Google Workspace accounts
Script runtime	6 min / execution	6 min / execution
Custom function runtime	30 sec / execution	30 sec / execution
Simultaneous executions	30 / user	30 / user
Email attachments	250 / msg	250 / msg
Email body size	200 KB / msg	400 KB / msg
Email recipients per message	50 / msg	50 / msg
Email total attachments size	25 MB / msg	25 MB / msg
Properties value size	9 KB / val	9 KB / val
Properties total storage	500 KB / property store	500 KB / property store
Triggers	20 / user / script	20 / user / script
URL Fetch response size	50 MB / call	50 MB / call
URL Fetch headers	100 / call	100 / call
URL Fetch header size	8 KB / call	8 KB / call
URL Fetch POST size	50 MB / call	50 MB / call
URL Fetch URL length	2 KB / call	2 KB / call

https://www.steegle.com/google-products/google-apps-script-faq

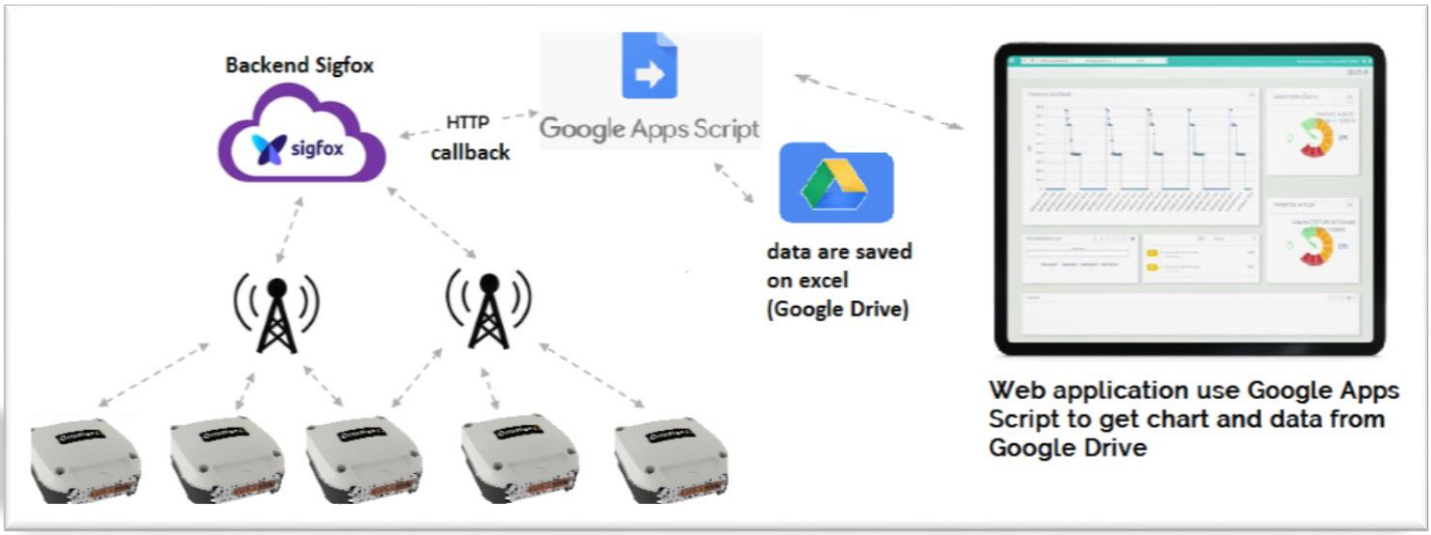
How many users can access my web-app simultaneously?

It depends on a lot of things. It may fail with one user or work for a million. The key issue to this question regards the various quotas we have to live with in Apps Script. And most of them we're not aware of until we hit them.

It's a hard thing to analyze, but let's try the bare minimum. Does your app run as the user or as the developer? If it's the developer, than it's going to be much less users, mainly because everything counts against your quota (instead of each user having its own). But just to give you some reference, I'll give a wild guess based on my experience... if you're a great developer and already worked-around the various quotas, maybe a 100 users, probably less. The only way to know would be testing, but as you can imagine, this is hard to do. If you app is standalone running with access to anyone even anonymous, then you might have a reasonable shot at stress testing it with tools like JMeter. But the sole acting of stress testing your app can make it break so hard that it might be difficult to recover from. Your account may get blocked for a day or more until your quotas are refreshed, etc.

If your app runs as the user, like an add-on, then it might work of an "infinity" amount of users. And even if it breaks, it's going to be individually, since the quotas are counted against each user and not you (the developer). You just have to pay attention in your script for resources that are shared and therefore count against you. Usually app-centric storage, like ScriptDb, Cloud-Sql or your custom external backend (e.g. App Engine). If one of these goes wrong your app may fail to all your users at once!

Under development:



Web application use service of Google Apps Script to serve chart and specific data saved of each client on Google Drive, fast and freely service.