

Setting up google spreadsheet access.

To set app bot with access to our spreadsheets, we will have to enable google drive and google sheet APIs, create service account, set up keys and allow access to certain spreadsheet.

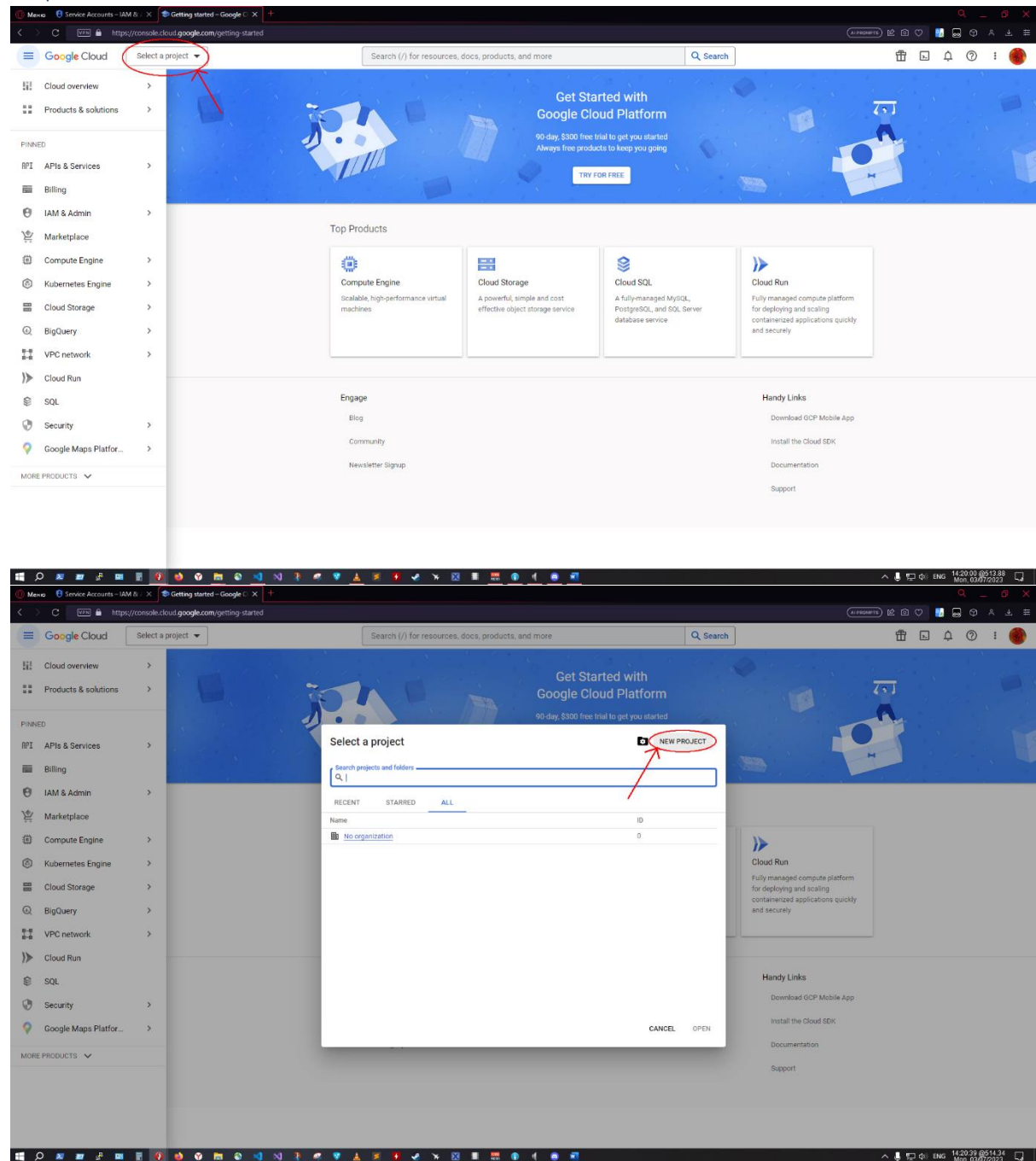
This guide is a walkthrough with all steps needed to set it up.

Enabling APIs

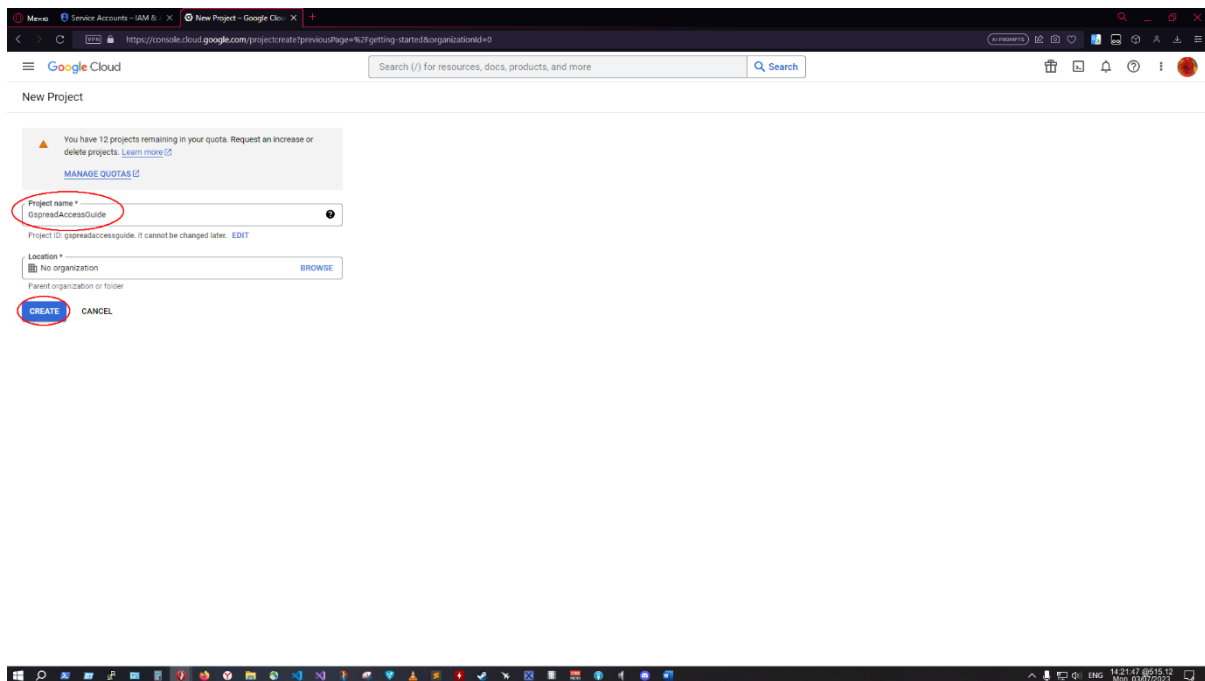
Go to <https://console.cloud.google.com>

First, we need to create the project. If you already have a project, skip to step 2.

Step 1

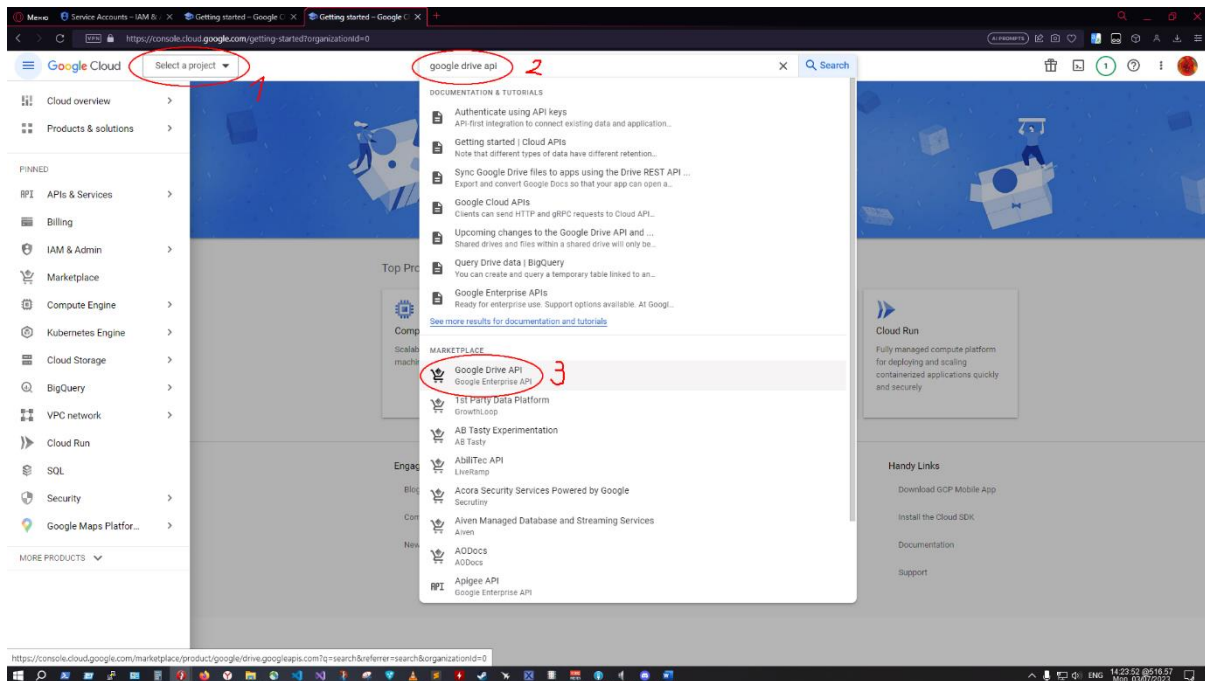


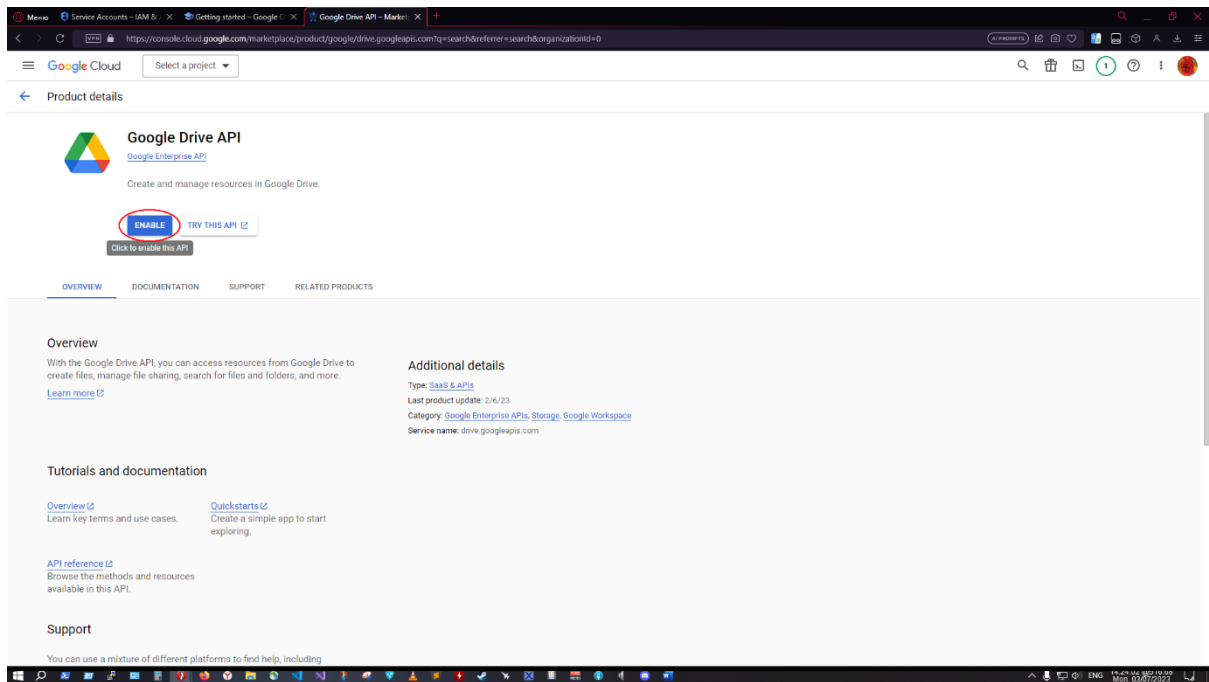
Choose the project name and click “Create”.



Step 2

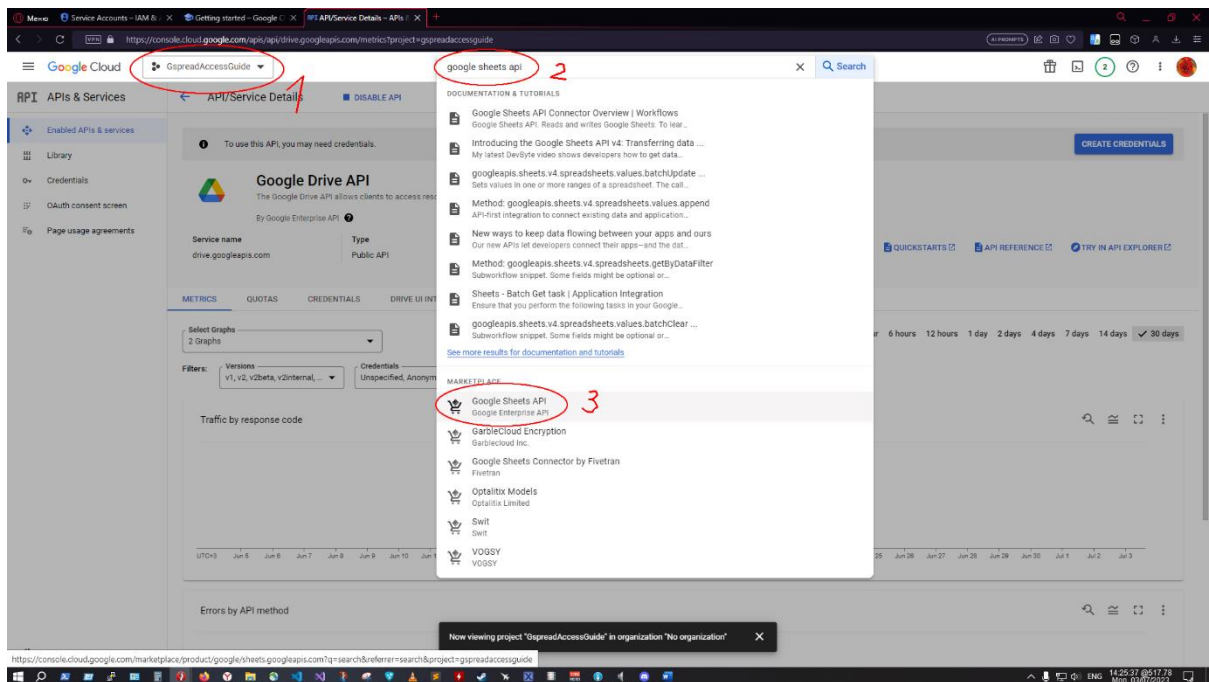
Now we need to enable both APIs for the created project. It's better to select the project before searching for them (1), in order not to get prompted with project selection upon API enabling. Enter “google drive api” in the search field and select “Google Drive API” from the dropdown hints in “Marketplace” section. Then just enable it.

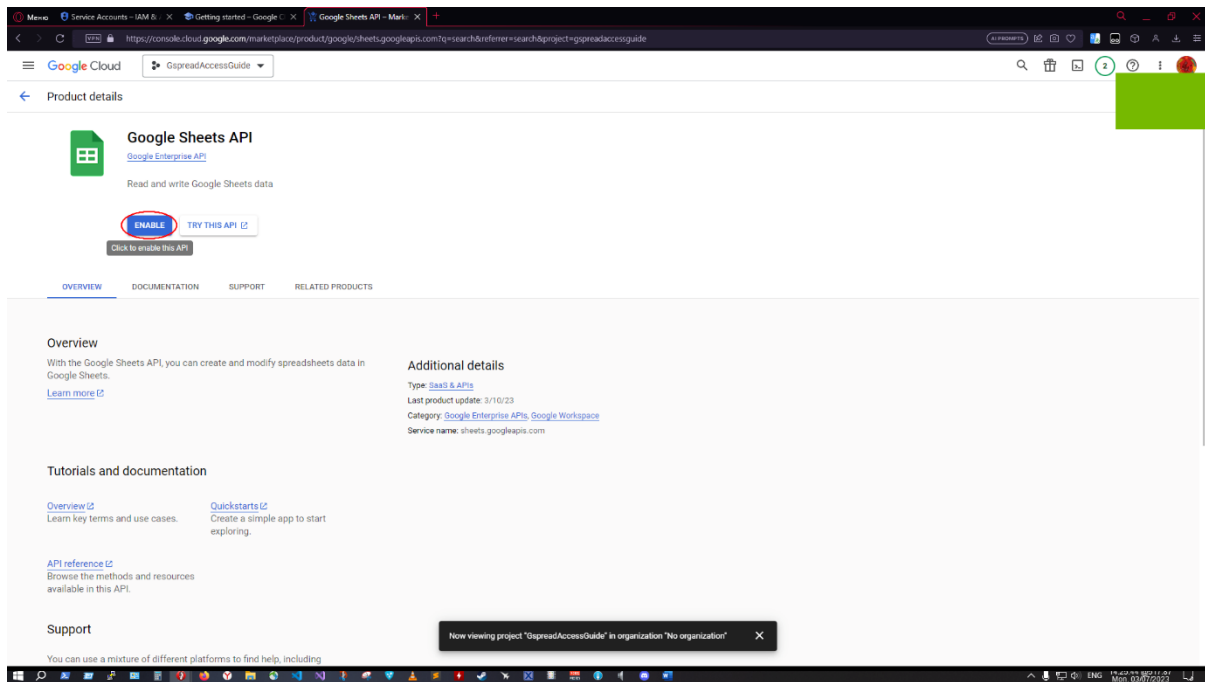




Step 3

Repeat Step 2 with “google sheets api” and select “Google Sheets API”.

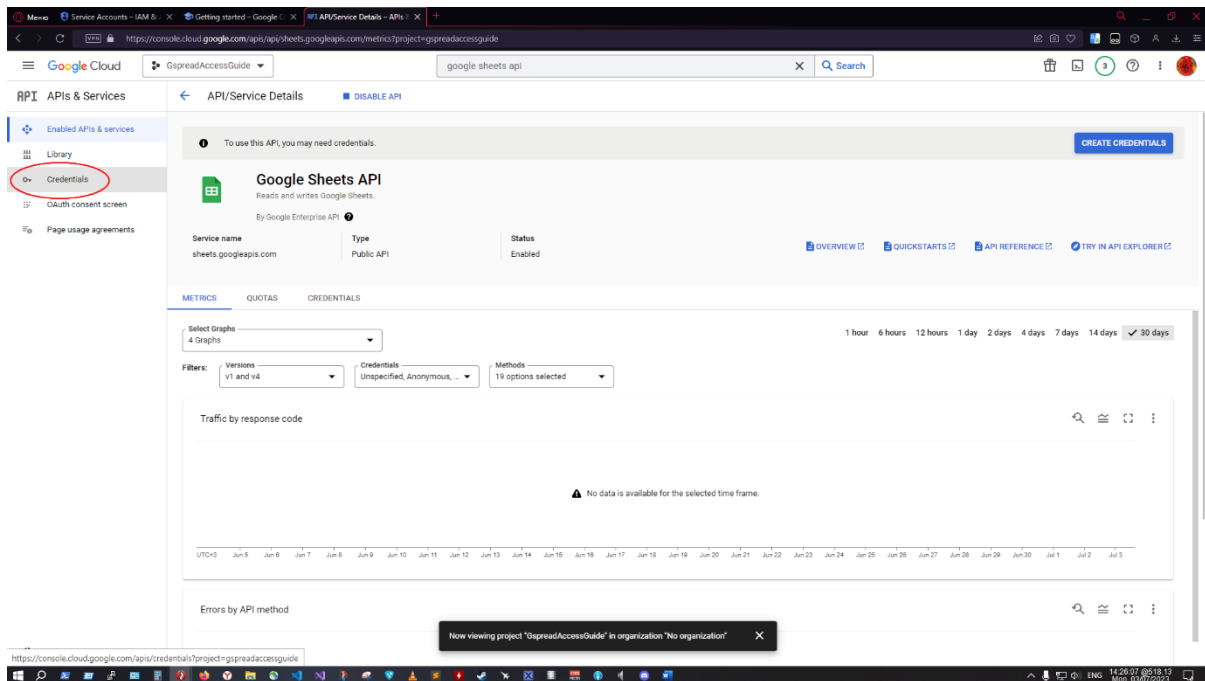


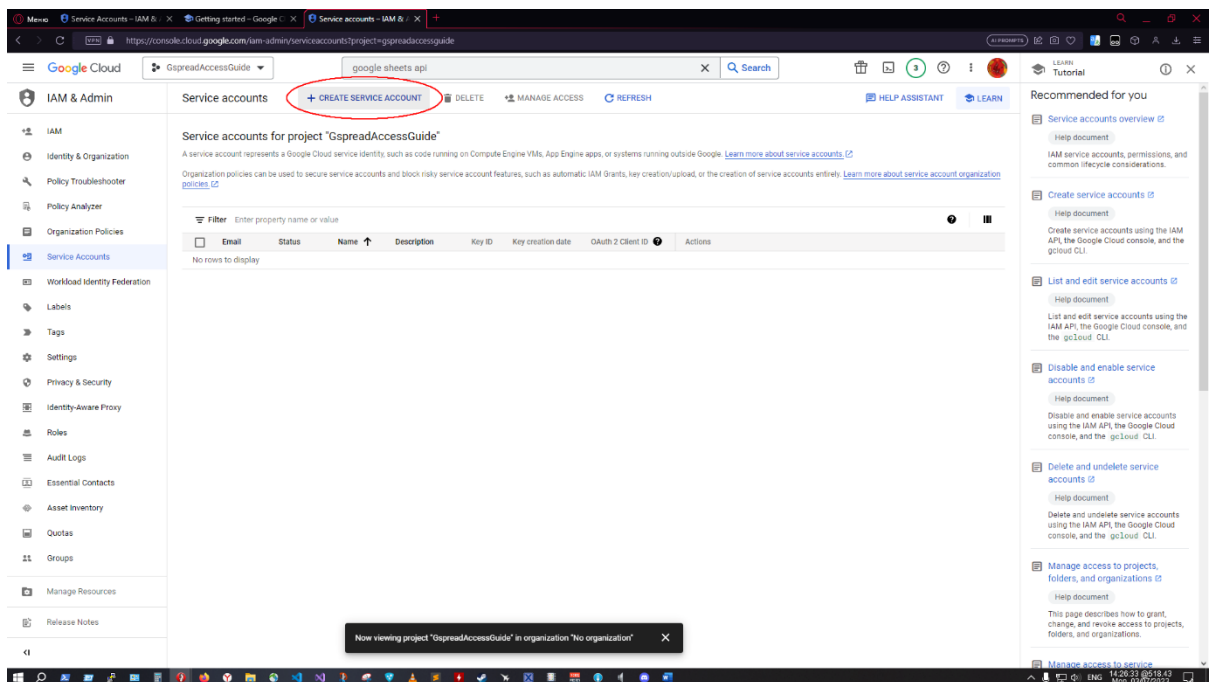
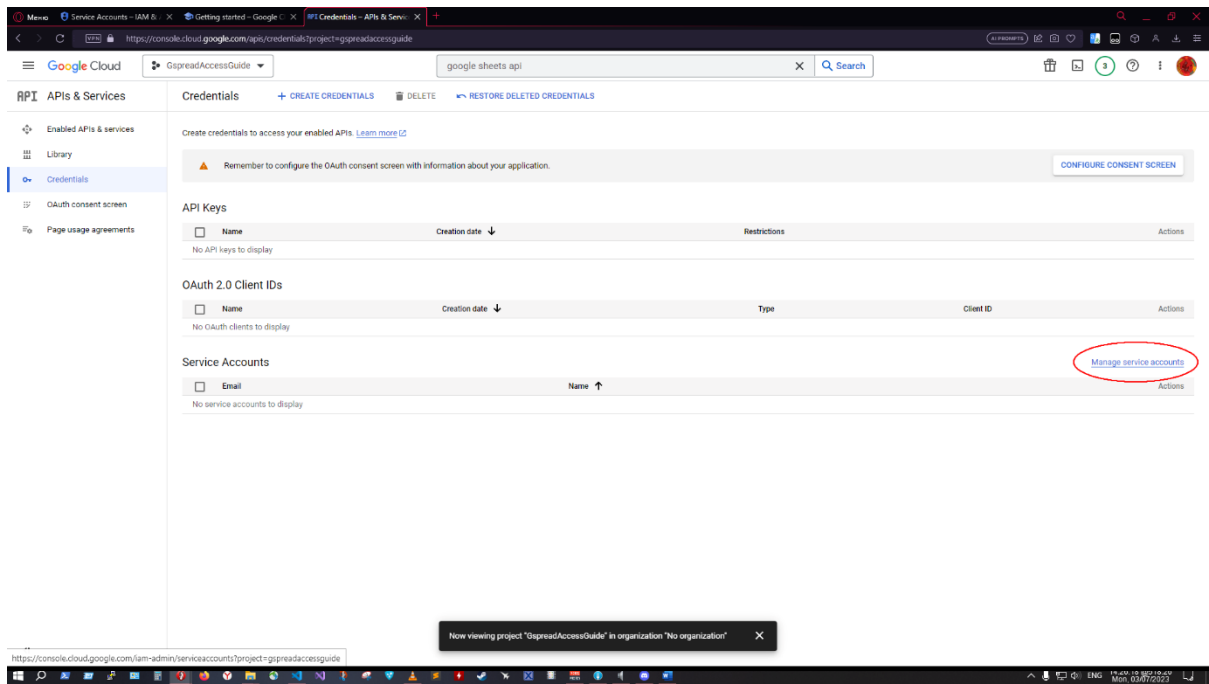


Service account and keys

Step 4

Now we need to set up service account. Go to “Credentials” and click on “Manage service accounts”. Then click “Create service account”.





Choose a name and leave 2 and 3 as is.

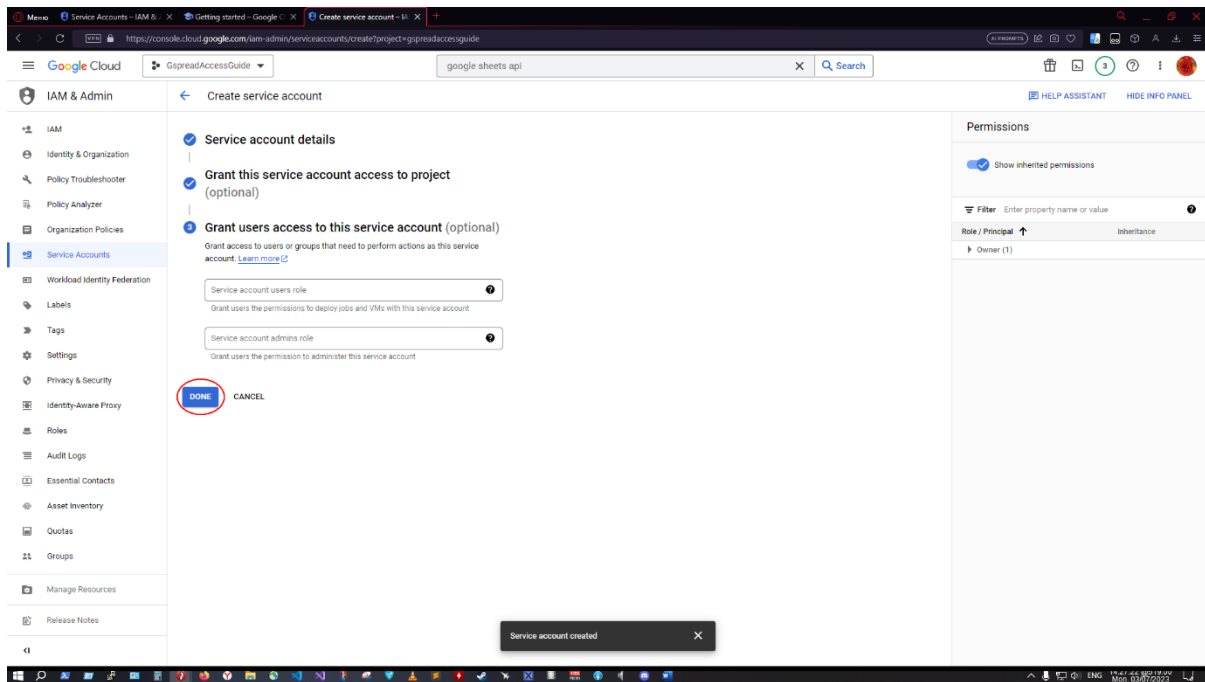
The image displays two sequential screenshots of the Google Cloud IAM & Admin console, specifically the 'Create service account' wizard.

Top Screenshot (Step 1: Service account details):

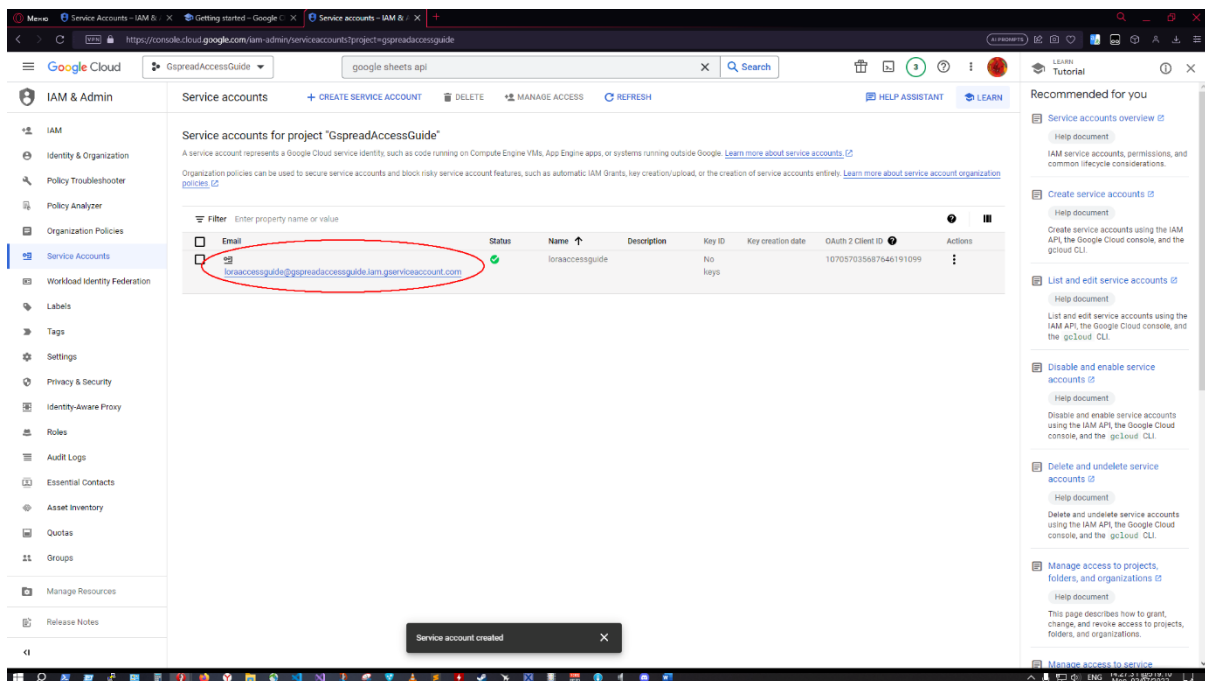
- The 'Service account name' field is populated with 'loraccessguide' and is circled in red.
- The 'Service account ID' field is populated with 'loraccessguide'.
- The 'Email address' field is populated with 'loraccessguide@gspreadaccessguide.iam.gserviceaccount.com'.
- The 'Service account description' field is empty.
- The 'CREATE AND CONTINUE' button is visible at the bottom of the form.
- A notification banner at the bottom states: 'Now viewing project "GspreadAccessGuide" in organization "No organization"'.

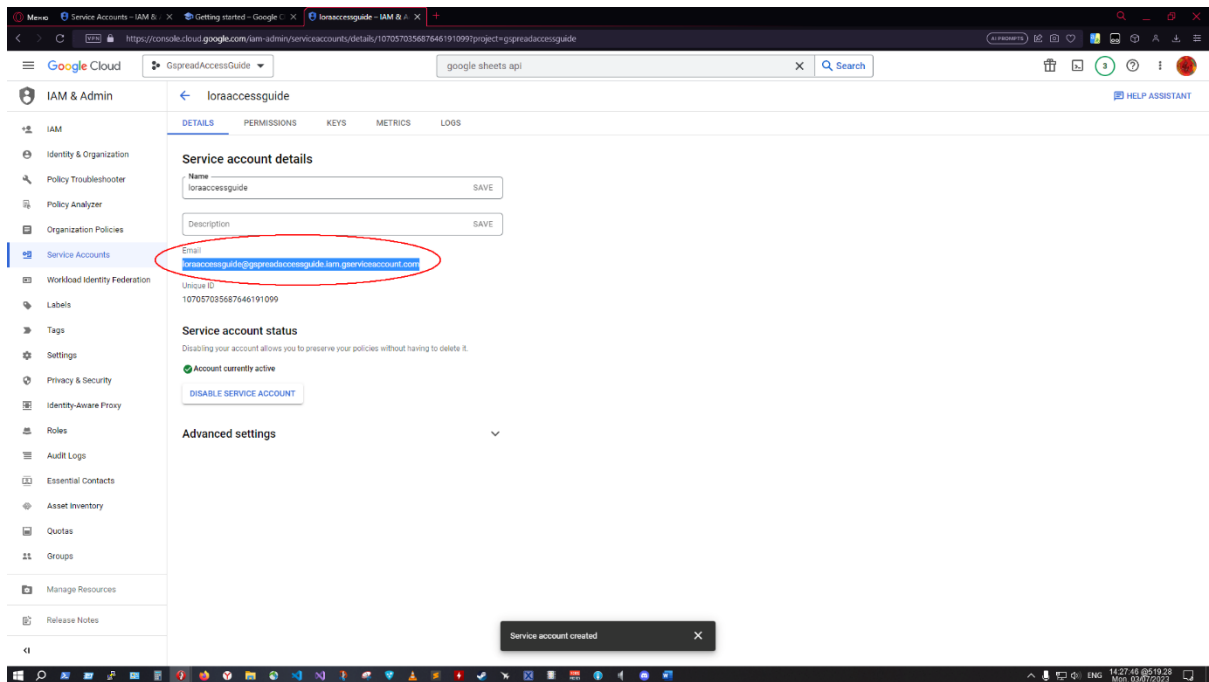
Bottom Screenshot (Step 2: Grant this service account access to project):

- The 'CONTINUE' button is circled in red.
- The 'Grant this service account access to project (optional)' section is expanded, showing a 'Select a role' dropdown and an 'IAM condition (optional)' link.
- The 'Grant users access to this service account (optional)' section is also visible.
- A notification banner at the bottom states: 'Service account created'.



Now we successfully created a service account. You may enter it and save the email. We will need it later on for giving access to a certain spreadsheet.

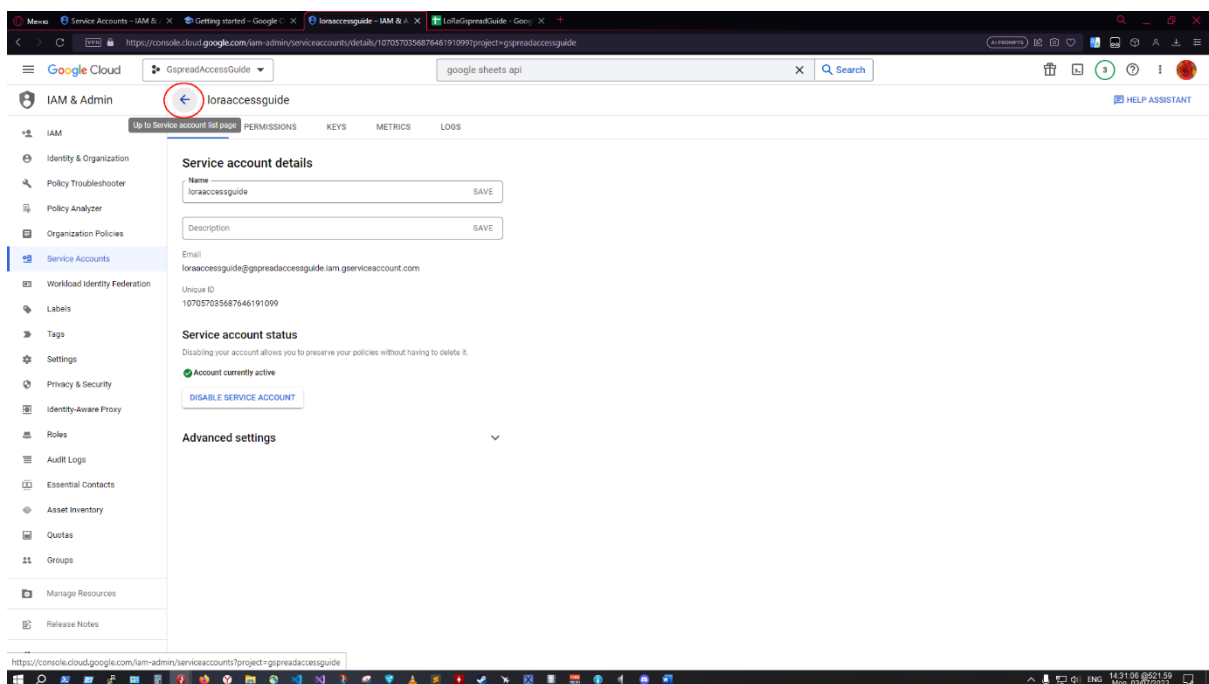


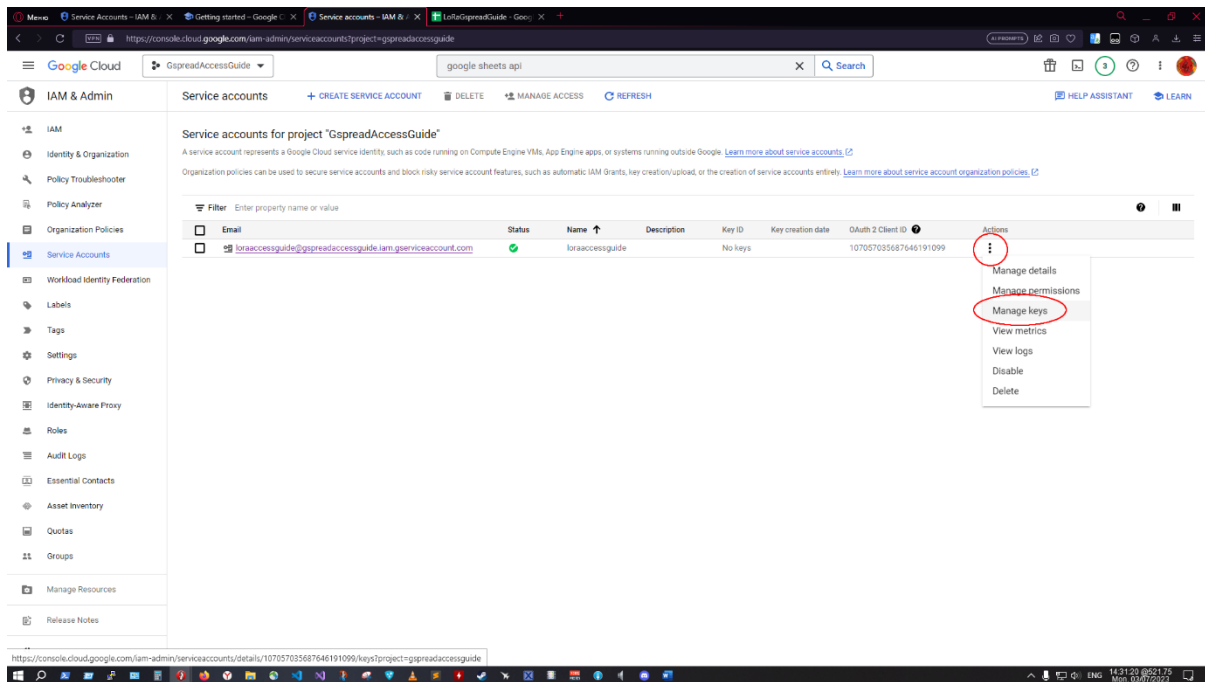


Step 5

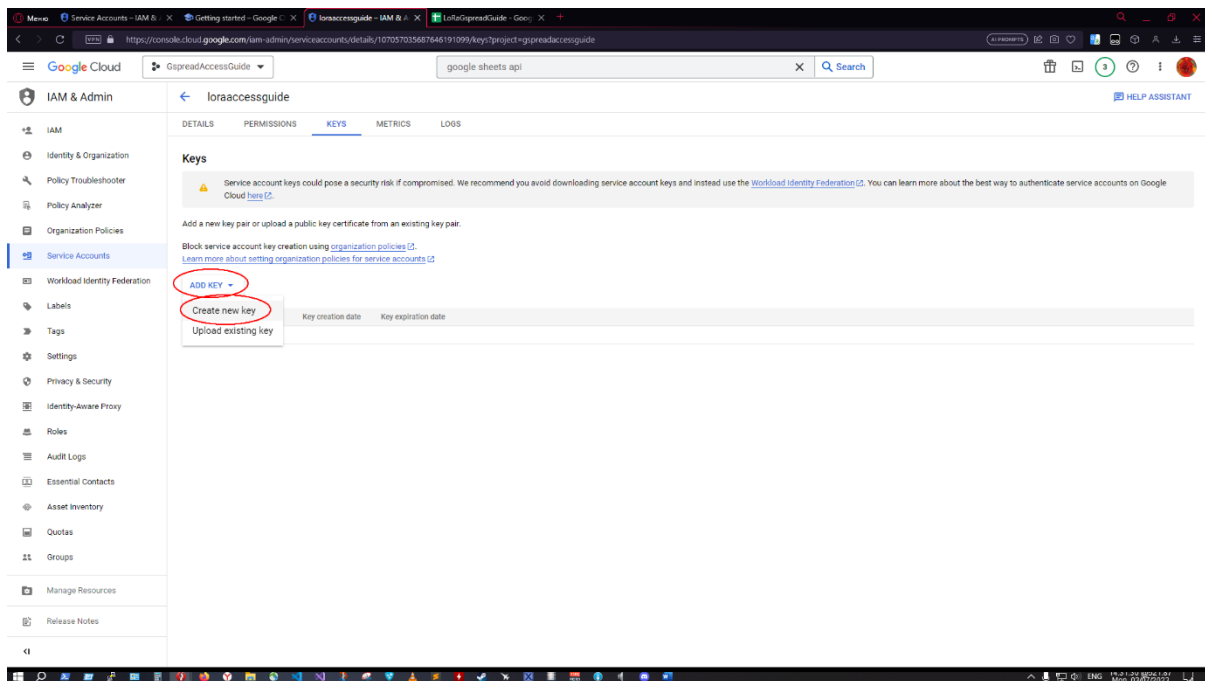
Service account still needs keys to be able to access drive and spreadsheet data from a certain machine. Thus, we have to set up keys.

Go back and click 3 dots under “Actions” column. Choose “Manage keys”.

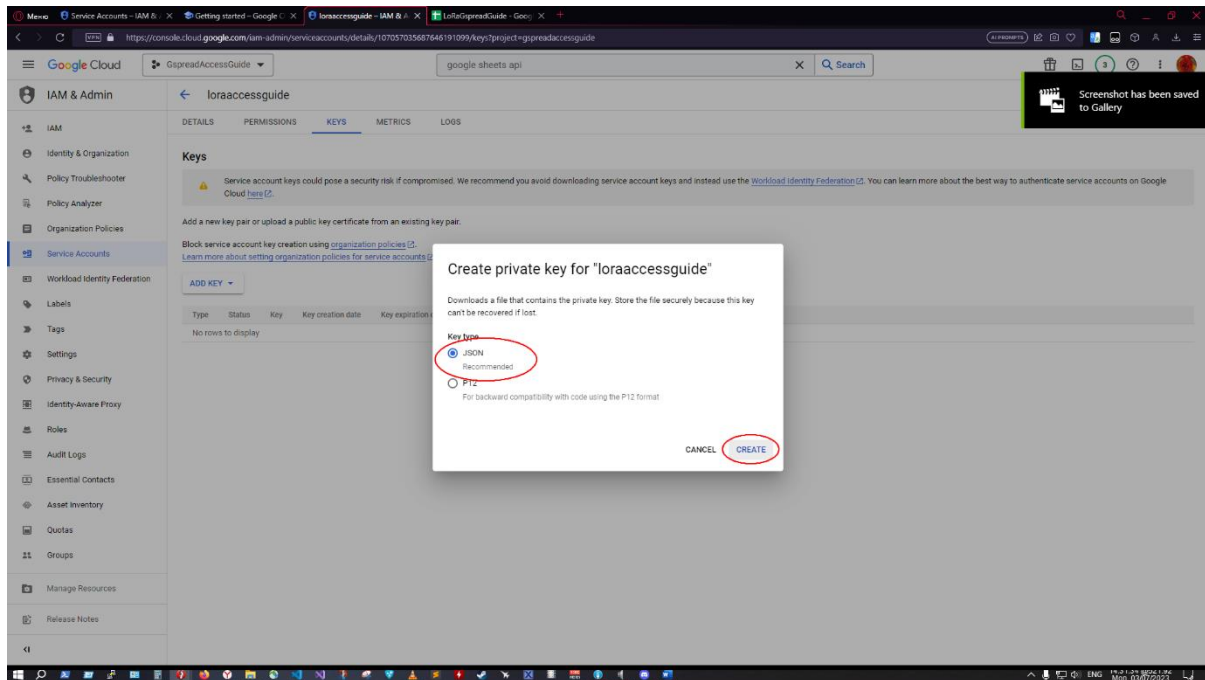




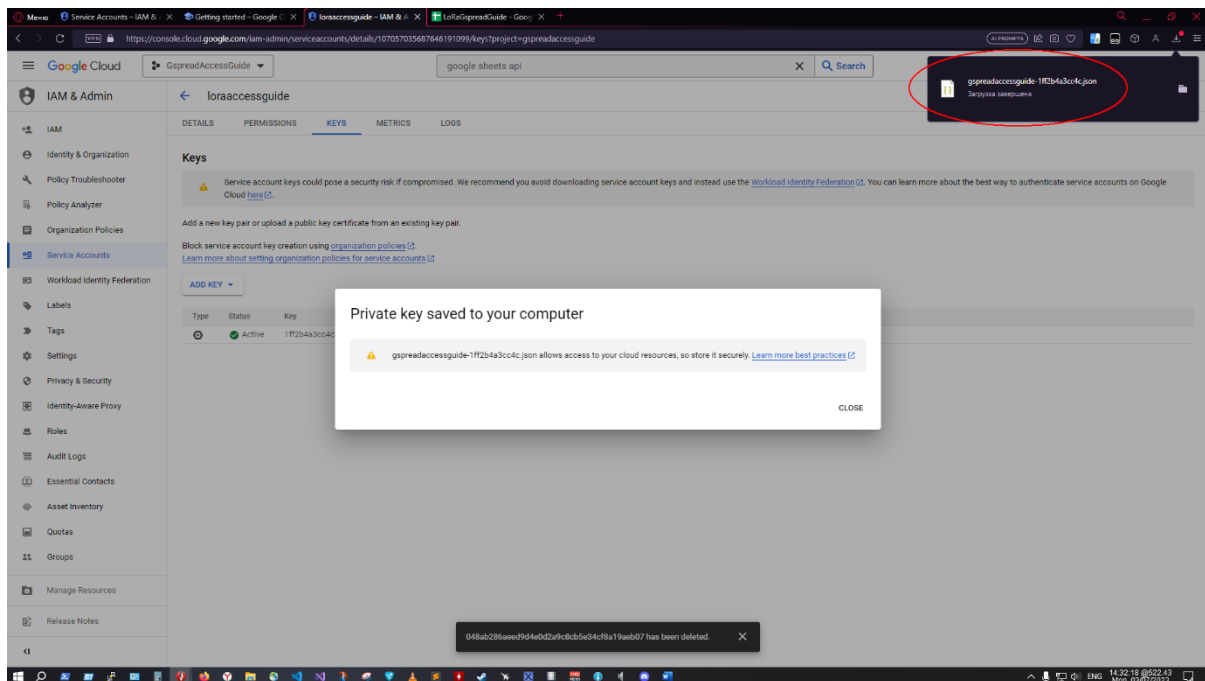
Click “Add key” and choose “Create new key”.



Make sure that JSON type is selected and click “Create”

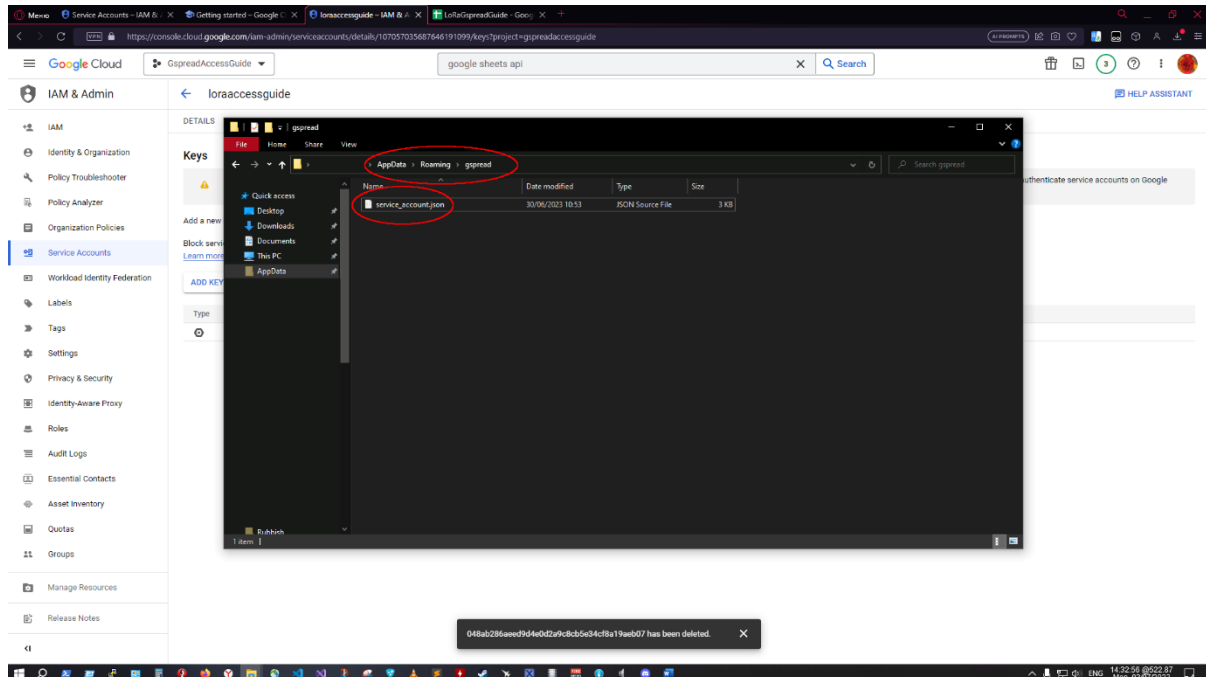


Private key will be saved to your device. It is better to move this key to a certain folder that gspread defaults to.



For Windows: [User]\AppData\Roaming\gsread

For Linux: ~/.config/gspread

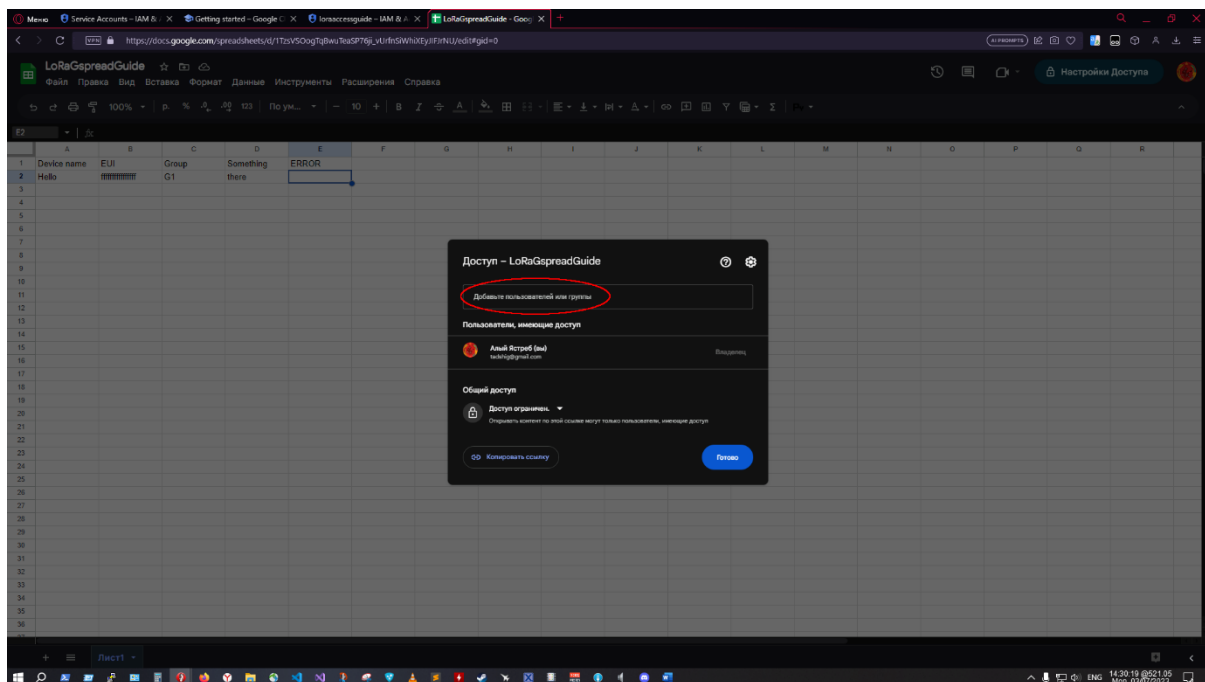
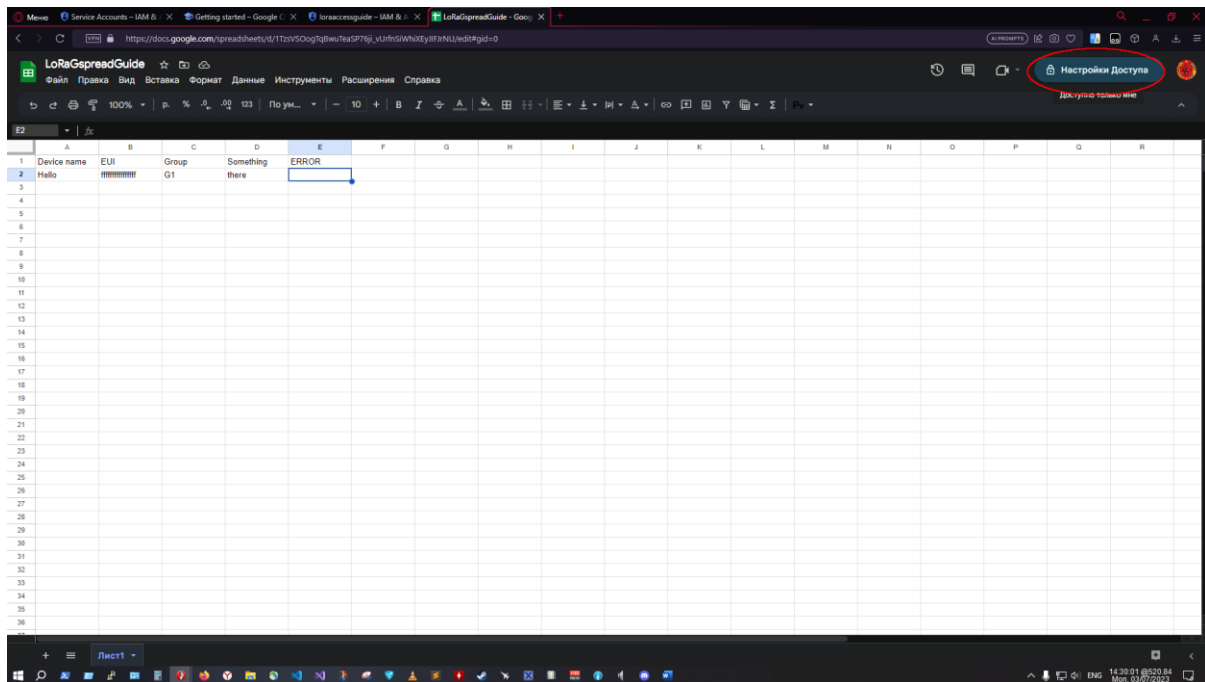


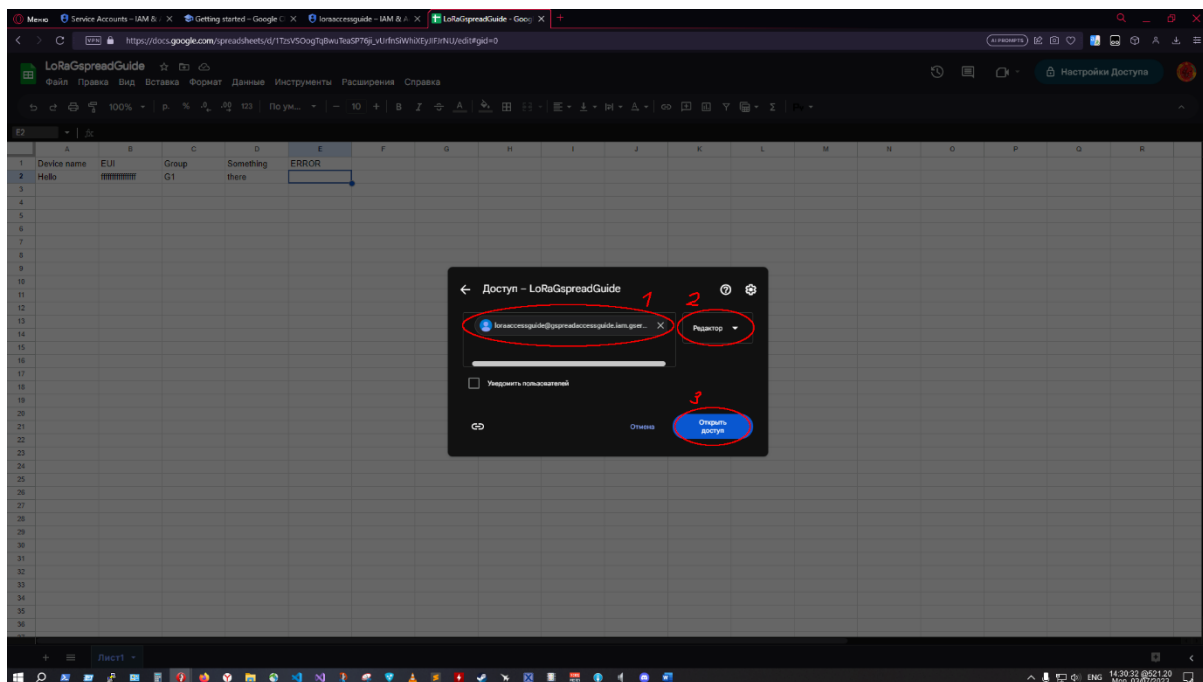
By default gspread will search for “service_account.json”, but you may rename it and modify the config file to use your custom filename rather than default.

Giving access to spreadsheet

Step 6

Now that we set up service account and keys, the only thing left to do is to give it access to a spreadsheet we want to read with gspread. It is done the same way as for other person, just make sure to use the email we copied earlier and give it editor access, since gspread will have to do both, read and write to the spreadsheet.





Note the format of table on the spreadsheet, (it lacks “App key” column, so it will break things.) It is important that it has the first row for headers. All other rows are considered to be devices.

“GoogleSpreadsheetParser.py” script will only care about 4 headers “Device name”, “EUI”, “ERROR” and “App key”. If you don’t want to modify the script, make sure to have the same header names.

“Device name” and “EUI” are read-only, while “App key” and “ERROR” are overwritten after every run, (the app-key may stay the same if device already existed in the server database.)

Here is the example of how table may look like:

Device name	Group	EUI	App key	Comment	Or	What?	ERROR
Device1	G1	2c7f12042007df	2ab423e4651e3271c5fecb1076164a9	Something			Was not able to create device 'Device1' (2c7f12042007df). Hint: It is possible that EUI is occupied by the device in other application.
Device2	G2	2c7f1204200708d	b683b543431b41c9c5d22d99a9986b0	SMore something			
Device3	G3	2c7f120420036fe	278d53a0748b9a303b455ad908656dd0	anyway			
Device4	G4	2c7f12042007da2	69376c780ca707447862acc85d3b5d				
Device5	G5	2c7f12042007a39	57d9020a7e774d297346a97073041fa				
ice12	Ice Ice	2c7f12042007f8	32cca9183d74a947ba9a9b08da8ffae5				
ice32	G6	2c7f1204200708d	1e486dcca7e02226661a91b53ba29c				
Devc	G7	12f7120420036fe	371a7abde4246da823d71f36d4e6a			doesn't	
Dice	D&D	2c7f12042007da2	16d4215he42b7acc5e55c2929844d8f				
Device43	43	2a6712042007a90	6644620a160295cab825523c2642b2e				
De	Dead	2c7f12042007a1a	10a4e456a4084a20a25a49d21c909				
ice7	IceIceIce7	2a6712042007a56	86ba4b18133451133c3b5a651087a7			malter	
ice1	Cold	2c7f12042007df	b767a907d98ab7560b700815dwa23				
ice2	Colder	2a671204200708d	3c9755459010ab6b4be9afccad3610	save			
Device3	G8	1c7f120420036fe	cb1322130a8938a0277a1842a761916				
Dice4	Dungeons&Dragons	2c7f12042007da2	697c595072a95c7c95792a7f5d381c2			me	
Device	Device	2a6712042007a90	c28130965299980118a4a5d48a4c5			now	
Devs	Under construction	2c7f12042007a1a	a097219361006569662c44da977ba40			please	
ice7	Baby	2c7f12042007a56	760a6706a934807e4d1090604a3d196			I want	
art	Piece of art	2a671204200708d	6377402bca3a4116b743986c0a3c3c			c	
abba	Povner	2abba1abba00708d	8623ffcca0a0a04b67a07773a66422af			prog	
Device200	Q200	2c7f12042007a1a	98a9e8b80bc3387052c95c0f5c4ef417			now	
f	Press F to pay respect	mmmm07da2		Stop it		Get	Some
DD	DD	4ef167a58442bba	5718079c554800854a8a9ba307163a	Help			
Device1		afafafafafafaf					ERROR: device is a duplicate of device in a row 2. Same Device name
Impostor		2a6712042007a90					ERROR: device is a duplicate of device in a row 11. Same EUI
		HEHEHEHE					ERROR: device has empty device name.
blazuytoy		Nope					ERROR: device has corrupted eui (Nope). Has to consist of 16 characters from "0123456789abcde"
i am new		23a756b894ab589c					ERROR: device has corrupted name (i am new). Only alphanumeric characters are allowed
Niau		tatatata					ERROR: device has corrupted eui (tatata). Has to consist of 16 characters from "0123456789abcde"
							ERROR: device has empty eui.
Korzina		2a4f5e694ab589c	d2eaf652ceb941ab63561940da82b135				
Pipka		2a4f5e694ab589c					ERROR: device is a duplicate of device in a row 33. Same EUI
Tbilqi		2a4f5e68cab589c					