

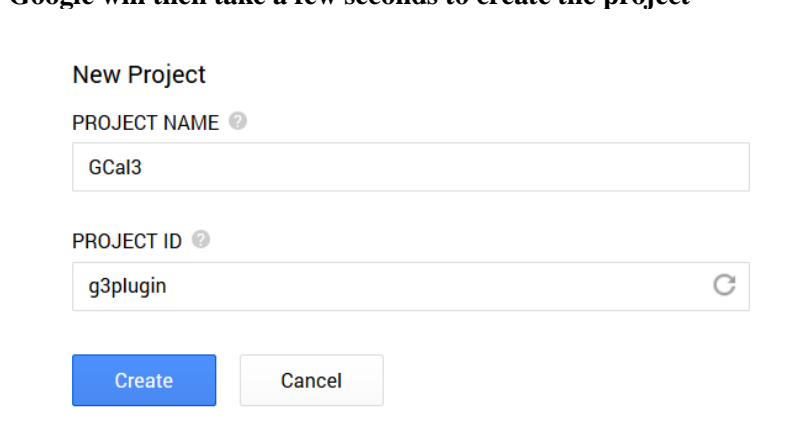
Setting up a Google V3 API Service Account

The new, V3, google API's require more rigorous security than was previously the case. The V2 APIs will apparently be turned off in Nov 2014, so any plugin that accesses google services will need to use an authentication mechanism. Several are available but often require a user interaction at each use.

This note describes how to set up a service account for google calendar (as an example). A service account allows credentials to be passed by an application (e.g. a plugin) without user interaction and is therefore suited to use by plugins.

I will describe a procedure that will later allow you to use the credentials with the Google Calendar 3 (GCal3) plugin but the process will be more-or-less the same for others.

1. Goto the google developer console projects page here
2. <https://console.developers.google.com/project>
3. Press the **Create Project** Button
Enter a project name – lets use 'GCal3'
Enter a project id – lets use 'g3plugin'
Press **Create**
Google will then take a few seconds to create the project



The screenshot shows the 'New Project' dialog box in the Google Developers Console. It has two input fields: 'PROJECT NAME' with the value 'GCal3' and 'PROJECT ID' with the value 'g3plugin'. Below the fields are two buttons: 'Create' (blue) and 'Cancel' (grey). The dialog box is titled 'New Project' and has a question mark icon next to the 'PROJECT NAME' and 'PROJECT ID' labels.



You should now be at a page that looks like this:



4. In the left hand pane - click on **APIS & AUTH** then **APIs**
You will see a list of google services – some are turned on by default (usually 4). You can leave these on, it does not really matter – what it's saying is that these are the API's that are available to this project and therefore will be accessible when the credentials (which we get later) are used. We'll leave these defaults alone in this example – but we want to include the API's for google calendar.

Scroll down to the 'Calendar API' (which shows as Off) and click the Off button. There may be a popup telling you the api is being enabled.

Scroll back up to the top and you should see 'Calendar API' added to the list of API's that are 'On'

NAME	QUOTA	STATUS
BigQuery API 	0%	<input checked="" type="checkbox"/> ON
Calendar API	0%	<input checked="" type="checkbox"/> ON
Google Cloud SQL		<input checked="" type="checkbox"/> ON
Google Cloud Storage		<input checked="" type="checkbox"/> ON
Google Cloud Storage JSON API 		<input checked="" type="checkbox"/> ON
Ad Exchange Buyer API	1,000 requests/day	<input type="checkbox"/> OFF

5. In the left hand pane - click on **Credentials**
Click on **Create New Client ID** and a popup will appear
Select the 'Service Account' radio button
Click on **Create Client ID** and then wait a few seconds while google creates the credentials
You will be asked if you want to open a file called 'somefilename.json'
Select 'Save' and the file will be downloaded to your machine
You will get a popup about the private key being downloaded to your machine – say ok

Your page should now look like this

Create new Client ID

Service Account

CLIENT ID	apps.googleusercontent.com
EMAIL ADDRESS	@developer.gserviceaccount.com
PUBLIC KEY FINGERPRINTS	

Generate new JSON key

Generate new P12 key

Delete

6. For GCal3 –the JSON key file (saved earlier) contains all the info needed so
Open the download folder that your browser uses

Find the json file. It will have a name like GCal3-12ab34bblabla58.json

Copy this file to a suitable folder on your machine and rename it to GCal3.json

You have now created a google service account that will allow access to your calendar and you have downloaded the credentials that will be used by the GCal3 plugin.

BUT – you also need to:

- (1) configure the calendar you want to access so that it knows which service accounts are allowed access, and**
- (2) Get the calendar ID**

7. Open the json credentials file created in step 6.

It will look something like this

```
{
  "private_key_id": "-----BEGIN PRIVATE KEY-----",
  "private_key": "-----BEGIN PRIVATE KEY-----",
  "client_email": "123456789-5ab5d6jab5s0g5pkgr9tb952jdkueb1h@developer.gserviceaccount.com",
  "client_id": "123456789-5ab5d6jab5s0g5pkgr9tb952jdkueb1h.apps.googleusercontent.com",
  "type": "service_account"
}
```

Copy the client_email value (the portion between the quotes) that looks like this

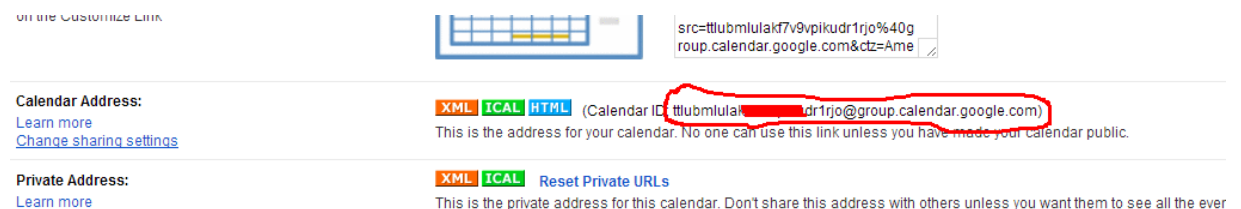
123456789-5ab5d6jab5s0g5pkgr9tb952jdkueb1h@developer.gserviceaccount.com

Be careful not to alter / save the json file !

8. Go to the google calendar that you want to access and do the following

Settings → Calendars → [your calendar] →

Copy the Calendar ID from here:



The screenshot shows the Google Calendar settings page. At the top, there's a 'Calendar Address' section with a URL that includes the Calendar ID: 'src=tlubmlulakf7v9vpikudr1rjo%40group.calendar.google.com&ctz=Ame'. Below this, the 'Calendar Address' is displayed as 'tlubmlulakf7v9vpikudr1rjo@group.calendar.google.com', which is circled in red. The 'Private Address' section is also visible, showing a similar URL structure.

Save the CalendarID for later use

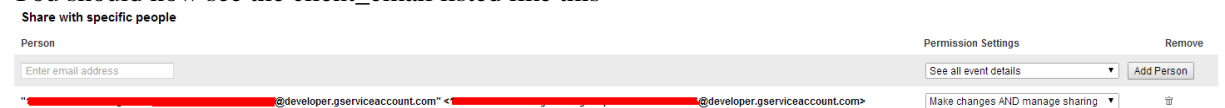
Next click on **Share this Calendar**

On the section that says “Share with specific people” Add the client-email from the step above.

Leave the permission at the default setting of “See all event Details”

Click on **Add Person**

You should now see the client_email listed like this



The screenshot shows the 'Share with specific people' dialog. It has a table with columns for 'Person', 'Permission Settings', and 'Remove'. The 'Person' column contains the email address '123456789-5ab5d6jab5s0g5pkgr9tb952jdkueb1h@developer.gserviceaccount.com'. The 'Permission Settings' column has a dropdown menu set to 'See all event details'. The 'Remove' column has a trash icon. There are also buttons for 'Add Person' and 'Make changes AND manage sharing'.

THAT's IT – the credentials and permsnsions have been set up