

# CalendarMod documentation

## Introduction

This is the documentation for the software CalendarMod developed by [Nguyen Minh](#). This software is motivated by the “great” scheduler at Vaasa University of Applied Sciences (VAMK)

The software is used to automate the process of:

- Downloading the iCalendar (.ics) file from the page given by VAMK
- Modifying the calendar file (delete unuseful information)
- Clearing the primary Google calendar of the user (after being authorized)
- Inserting the modified events to Google calendar

This documentation assumes its users to:

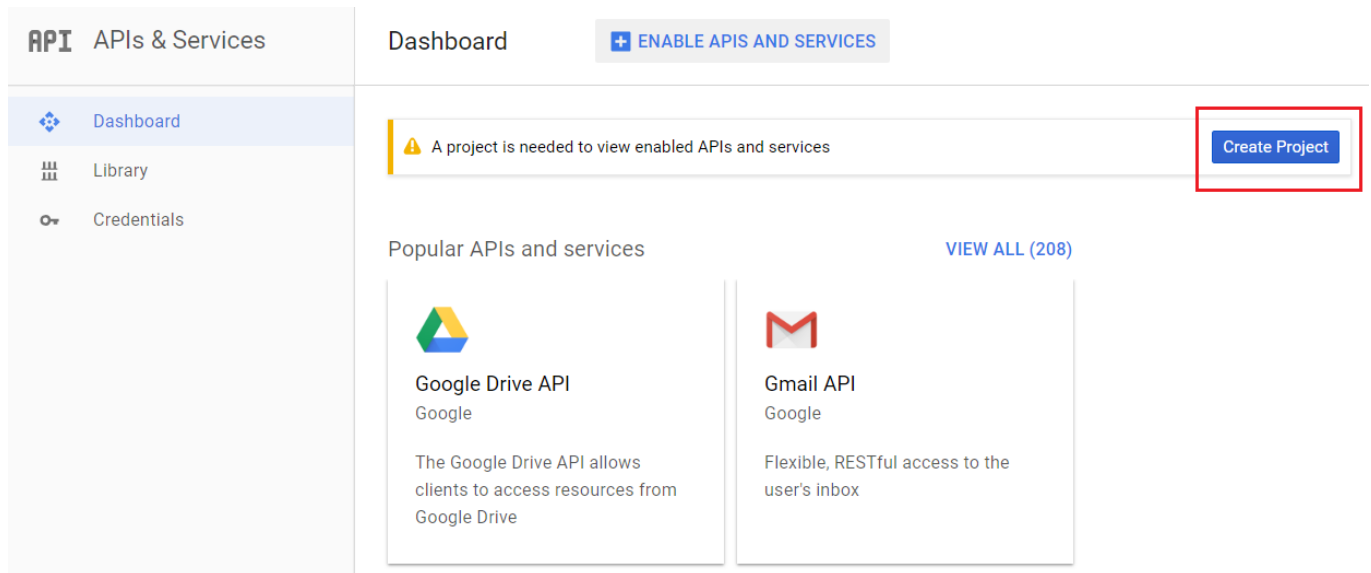
- Have already download and unzip the folder containing the executable file (.exe) of the program

## How to use

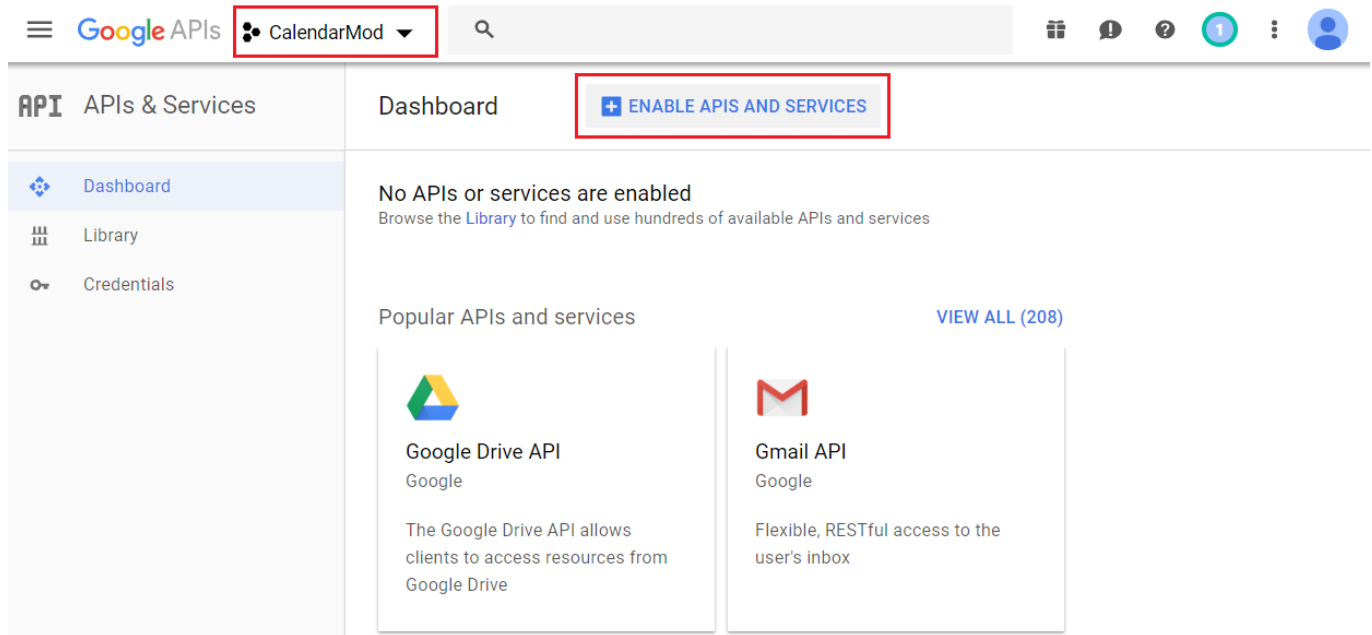
### Obtain Google’s client ID

To use the program, the user has to first obtain a client ID from Google. The file enables Google to identify a registered project and give the program access to the enabled APIs. The fact that this program is a desktop application prevents the developer from sharing his client ID (containing secret key) for the user to use.

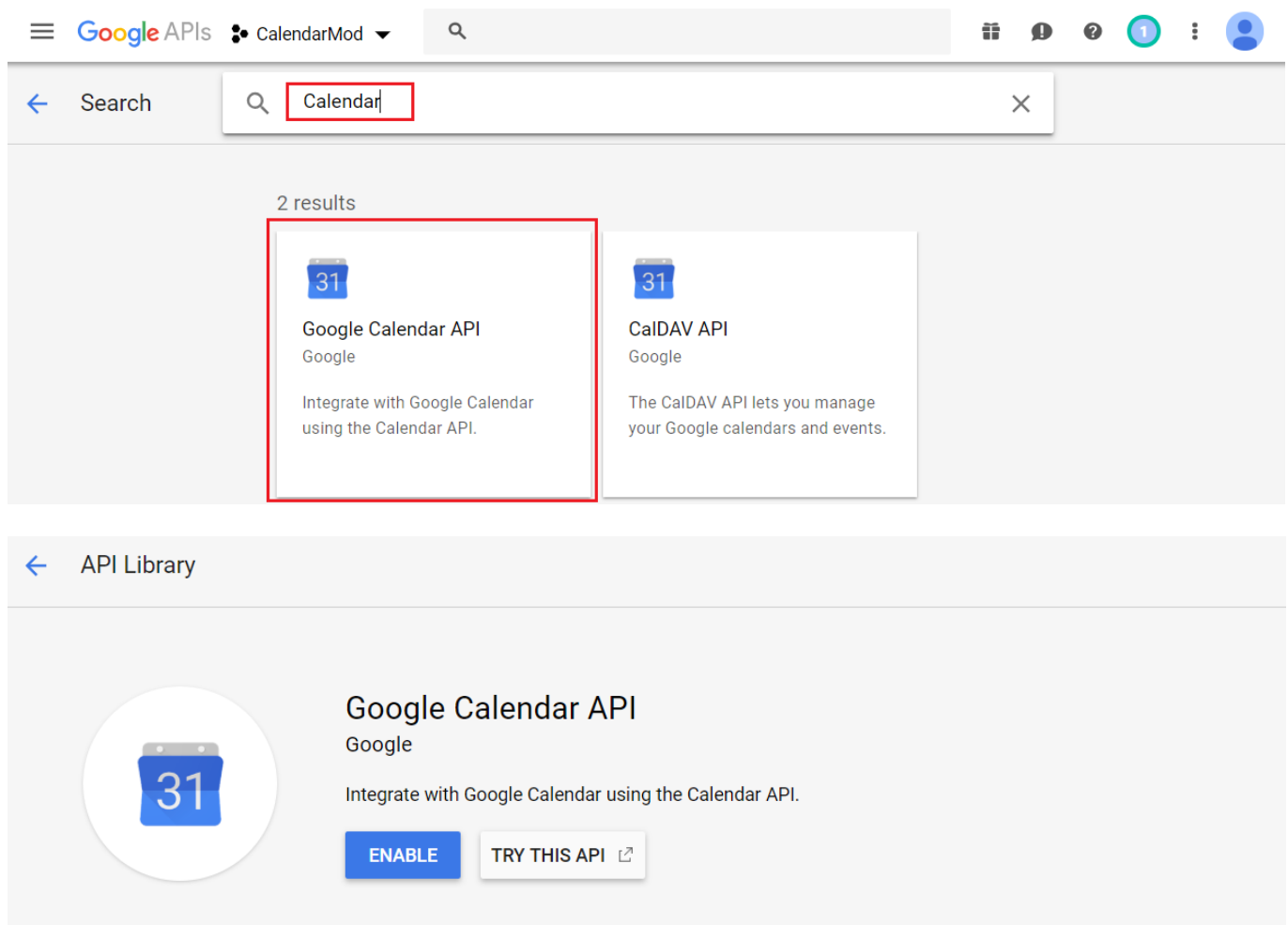
To obtain a client ID, one should visit [Google API Dashboard](#) and create a project.



The recommended project name is “CalendarMod”. Then, choose “ENABLE APIS AND SERVICES”.



Search for “Calendar”, choose “Google Calendar API” and enable it.



After about 10 seconds, the API should be enabled. However, to delete and insert events, Google also require authorization from the user. Therefore, the user has to create a credential.

31

APIs & Services

Google Calendar API

Overview

Metrics

Quotas

Credentials

Credentials

+ CREATE CREDENTIAL

DELETE

Credentials compatible with this API

To view all credentials or create new credentials visit [Credentials in APIs & Services](#)

Follow the below figure when creating a credential

Credentials

Add credentials to your project

1 Find out what kind of credentials you need

We'll help you set up the correct credentials

If you wish you can skip this step and create an [API key](#), [client ID](#), or [service account](#)

Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google Calendar API

Where will you be calling the API from?

Credentials can be restricted using details of the context from which they're called. Some credentials are unsafe to use in certain contexts.

Web browser (Javascript)

What data will you be accessing?

Different credentials are required to authorize access depending on the type of data that you request.

User data

Access data belonging to a Google user, with their permission

Application data

Access data belonging to your own application

What credentials do I need?

Credentials

Add credentials to your project

Find out what kind of credentials you need

Calling Google Calendar API from a web browser

2 Create an OAuth 2.0 client ID

Name

CalendarMod

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://\*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://www.example.com

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

http://localhost:8080

https://www.example.com/oauth2callback

Create OAuth client ID

Fill in the fields with the values:

- Name: CalendarMod
- Authorized redirect URIs: http://localhost:8080/

## Credentials

### Add credentials to your project

- ✓ Find out what kind of credentials you need  
Calling Google Calendar API from a web browser

- ✓ Create an OAuth 2.0 client ID  
Created OAuth client 'CalendarMod'

#### 3 Set up the OAuth 2.0 consent screen

Email address ?

nnminh171298@gmail.com

Product name shown to users ?

CalendarMod

✓ More customization options

Continue

## Credentials

### Add credentials to your project

- ✓ Find out what kind of credentials you need  
Calling Google Calendar API from a web browser

- ✓ Create an OAuth 2.0 client ID  
Created OAuth client 'CalendarMod'

- ✓ Set up the OAuth 2.0 consent screen

#### 4 Download credentials

client ID 352123584720-k2pfc7egjuedt5l5bvvnk5jqk06nklg9.apps.googleusercontent.com

Download this credential information in JSON format. This is always available for you on the credentials page.

Download I'll do this later

Done Cancel

At the final step, choose “Download” and then choose “Done”. The file containing your client ID should be now downloaded under the name “client\_id.json”.

Rename the file to “client\_id\_CalendarMod.json” and put it in the same folder with the executable file. The program is now ready to run.

### Using the program

Run the executable file. The program should display like the below figure.

CalendarMod

Group code  ☐ Calendar modified

User Google authorization ☐ Authorized

Clear primary calendar ☐ Cleared

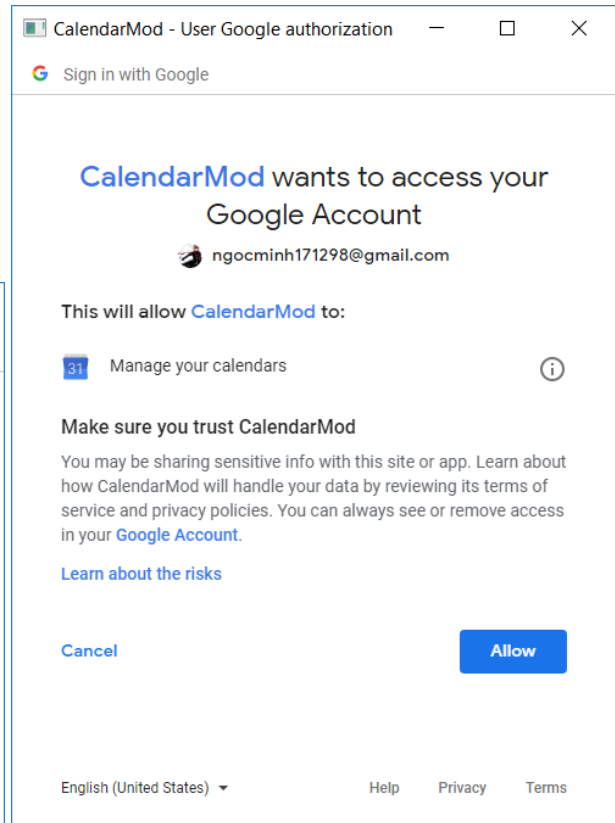
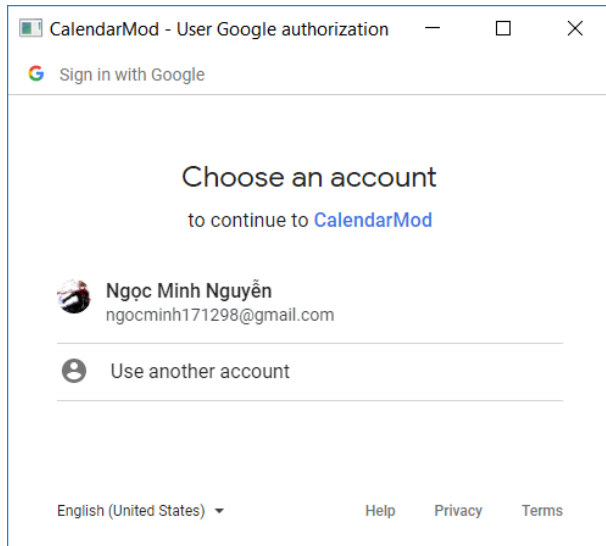
Insert events progress  0% ☐ All inserted

Revoke token ☐ Revoked

Start

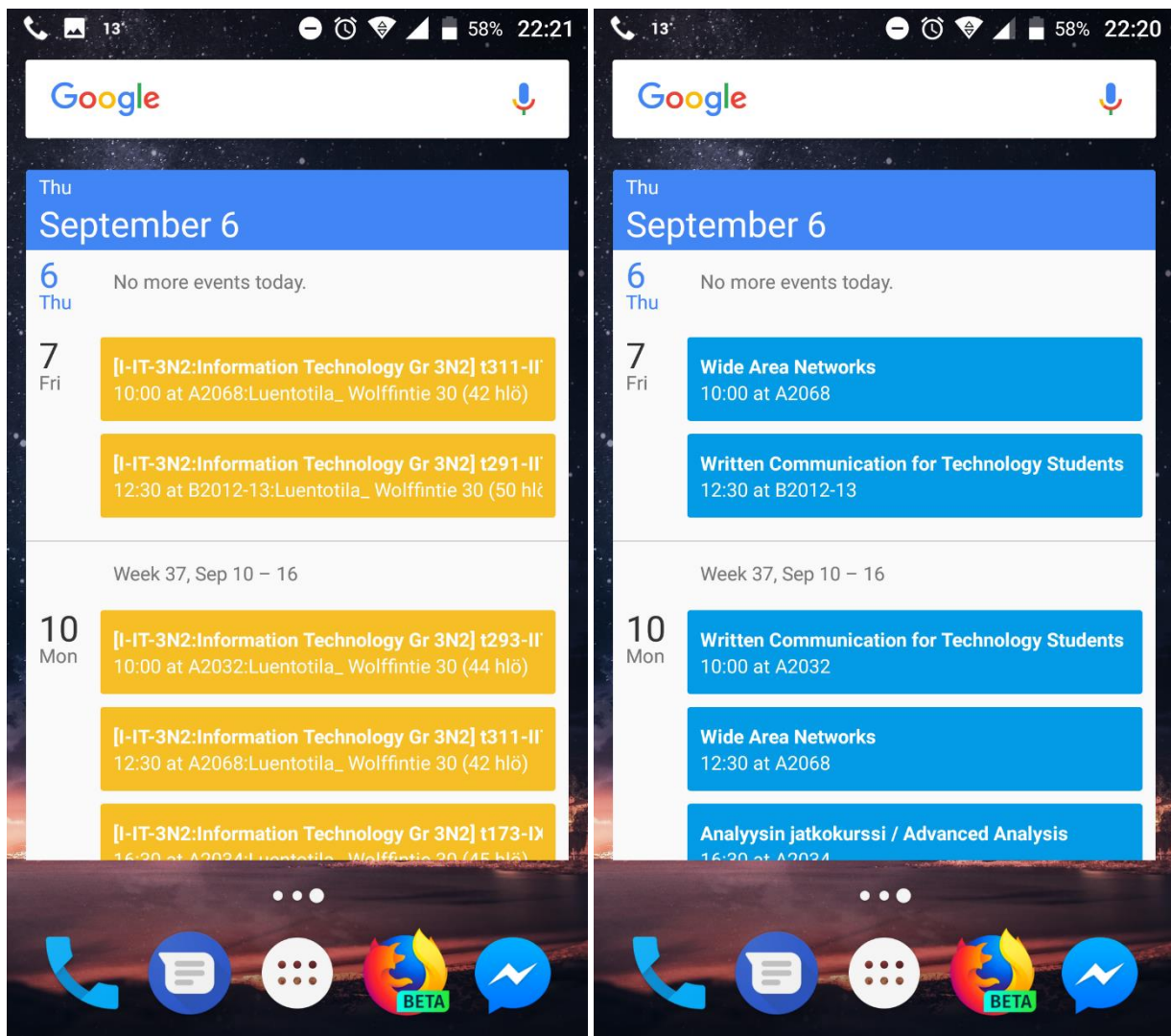
The user should enter the desired group code **IN PRINT** and click “Start”.

After modifying the calendar. Another window will show up asking for the user authorization.



After that, the other tasks are completely automated. The user just has to wait until all the checkboxes are checked and the job is done.

## Result



The left figure is how the calendar looks when added manually using the link given by VAMK without any modification

The right figure is how the calendar looks when added automatically using the program.