

How to Automate Google Calendar with Python using the Calendar API

By Karen Collective Published On 24 July 2019 , Updated On 24 July 2019

[Google calendar](#)

[Tech](#)

HOW TO AUTOMATE GOOGLE CALENDAR WITH PYTHON

KARENAPP.IO



Google Calendar

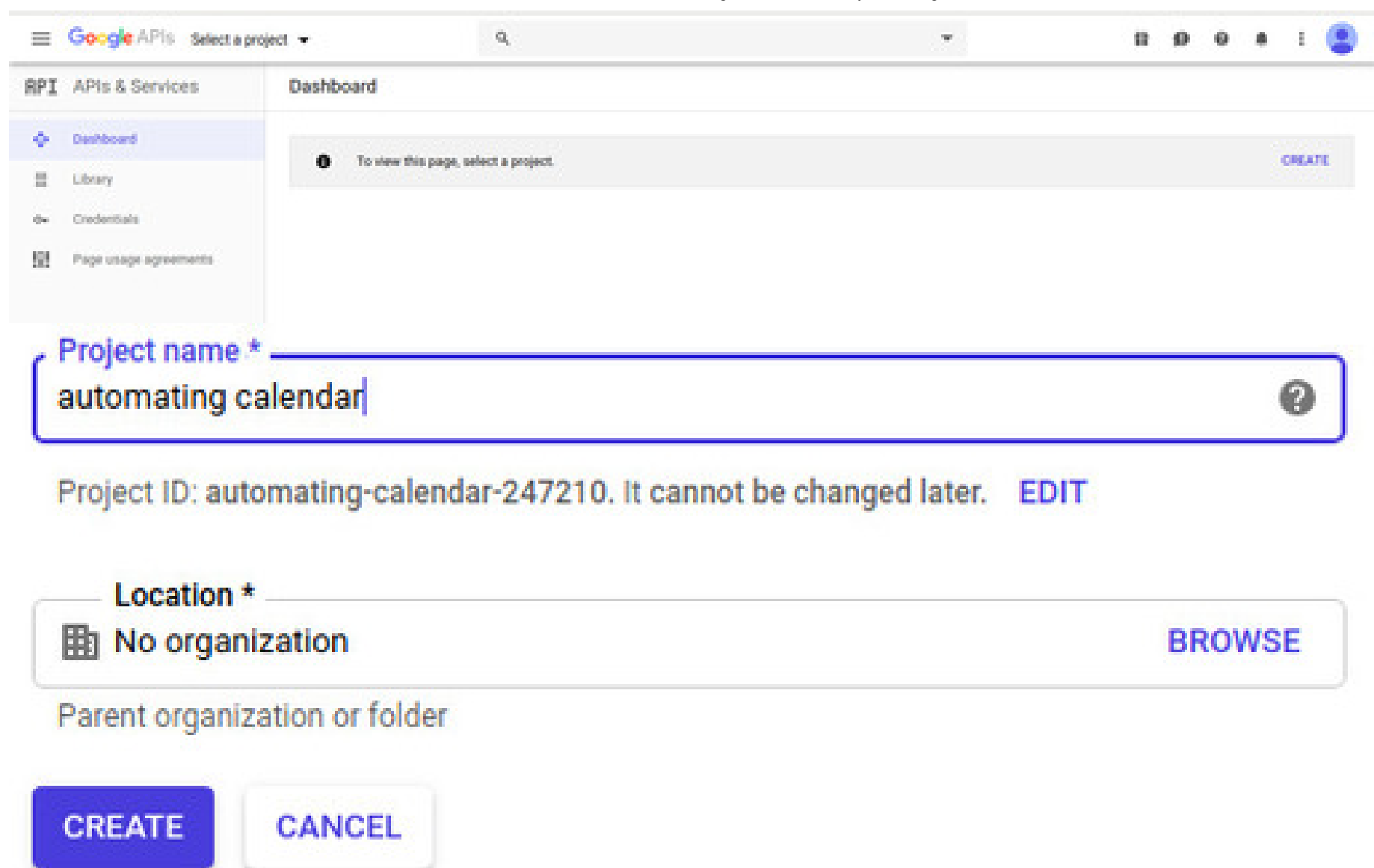


Google Calendar offers one of the most convenient and popular ways to manage schedule, events, meetings, or even plan out your holiday events. A few things you think will be straightforward end up being complicated for customised need. The Google Calendar API lets you automate your calendar, for those custom needs.

In this article we cover these aspects in detail, and provide a sample code using which you can try things out. We assume the basic knowledge of Python and working with APIs.

1. How to Create a project in Google Developer Console?

To kick things off let's head [Google Developers Console] (<https://console.developers.google.com/>) and create a new project. We name the project as Automating Calendar.



Google APIs Select a project

APIs & Services

Dashboard

To view this page, select a project. [CREATE](#)

Project name *
automating calendar ?

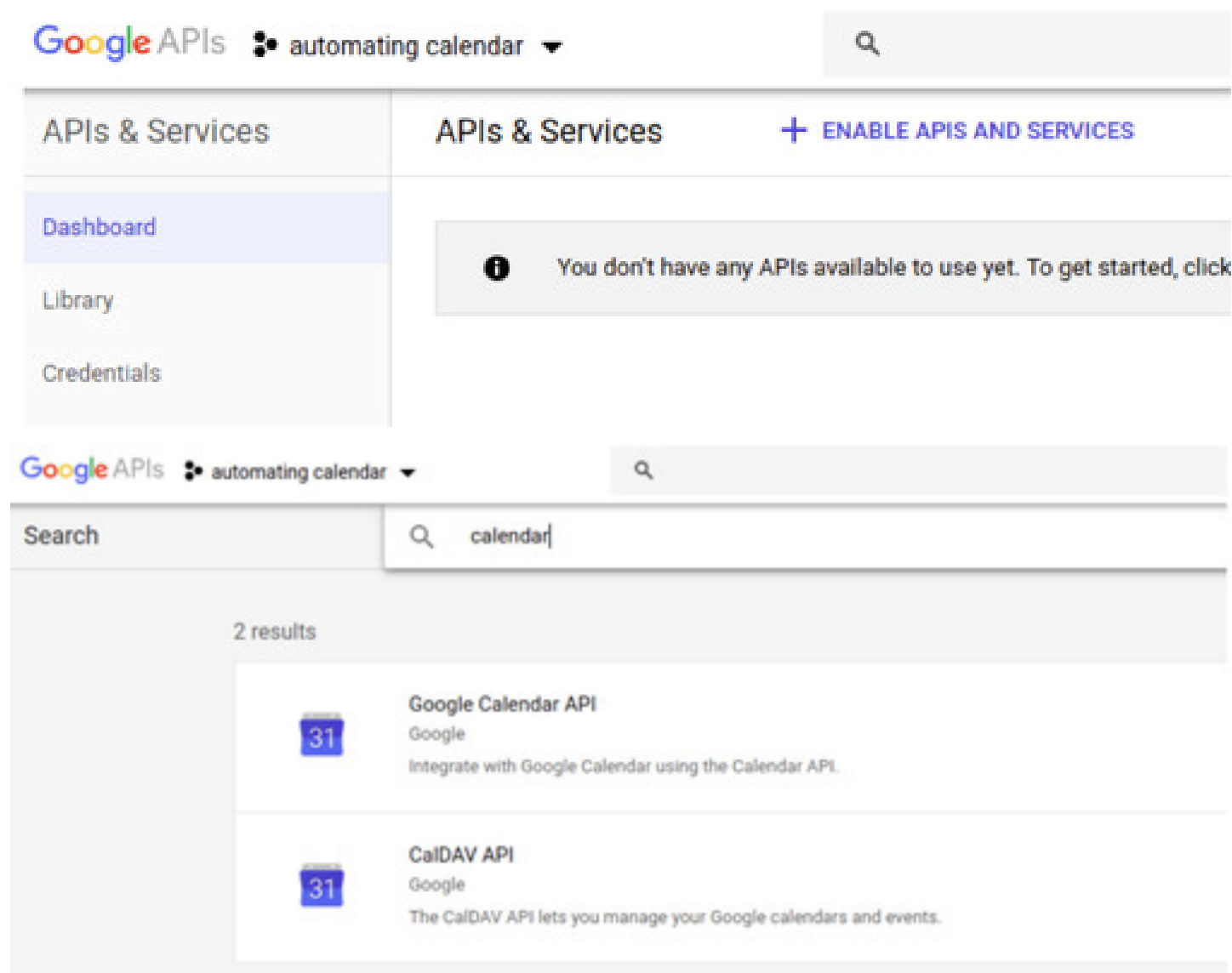
Project ID: automating-calendar-247210. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Then go to the dashboard and search for Calendar API in the search bar in order to enable the Calendar API for your project.



Google APIs automating calendar

APIs & Services [+ ENABLE APIS AND SERVICES](#)

Dashboard

Library

Credentials

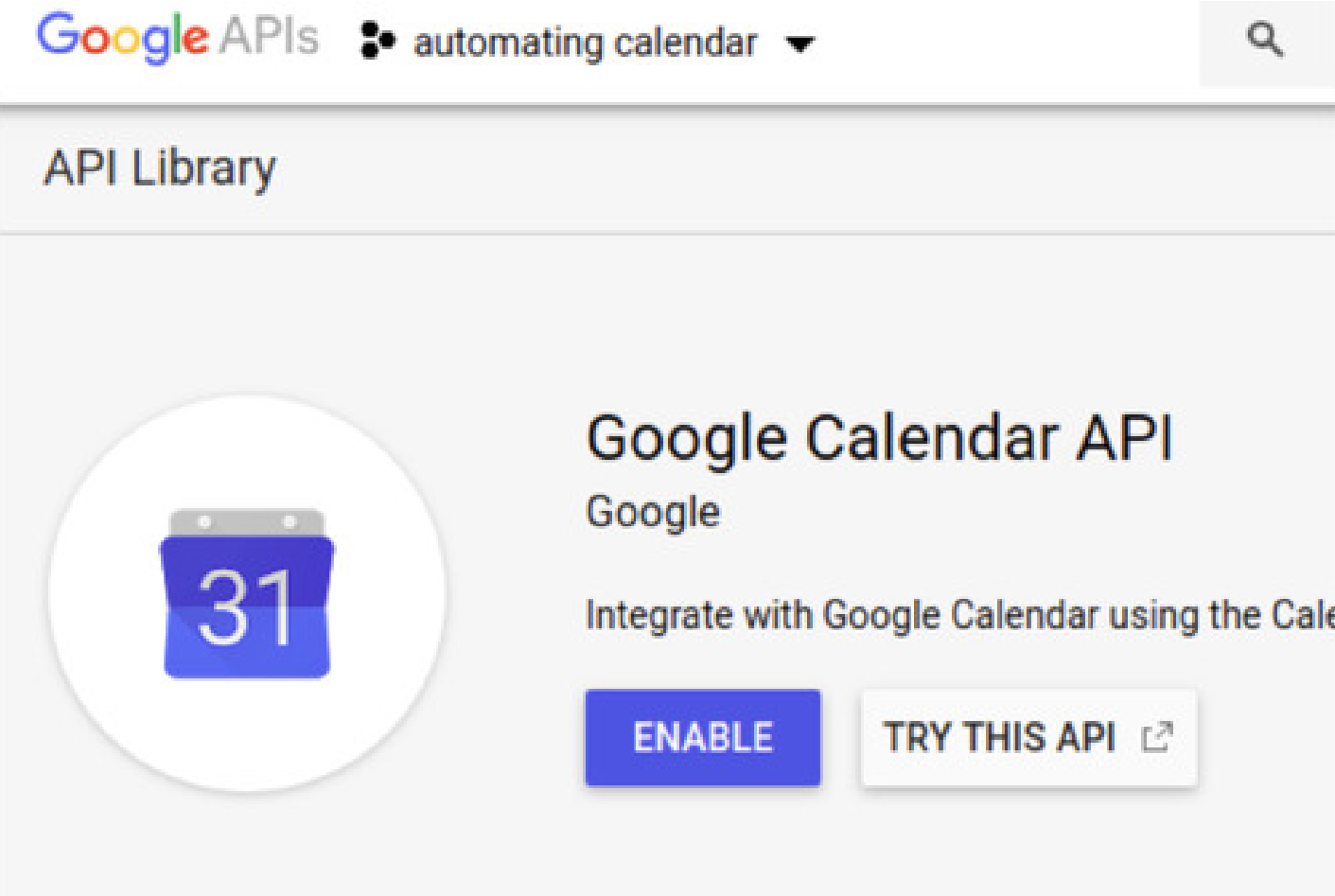
You don't have any APIs available to use yet. To get started, click

Search calendar

2 results

Google Calendar API
Google
Integrate with Google Calendar using the Calendar API.

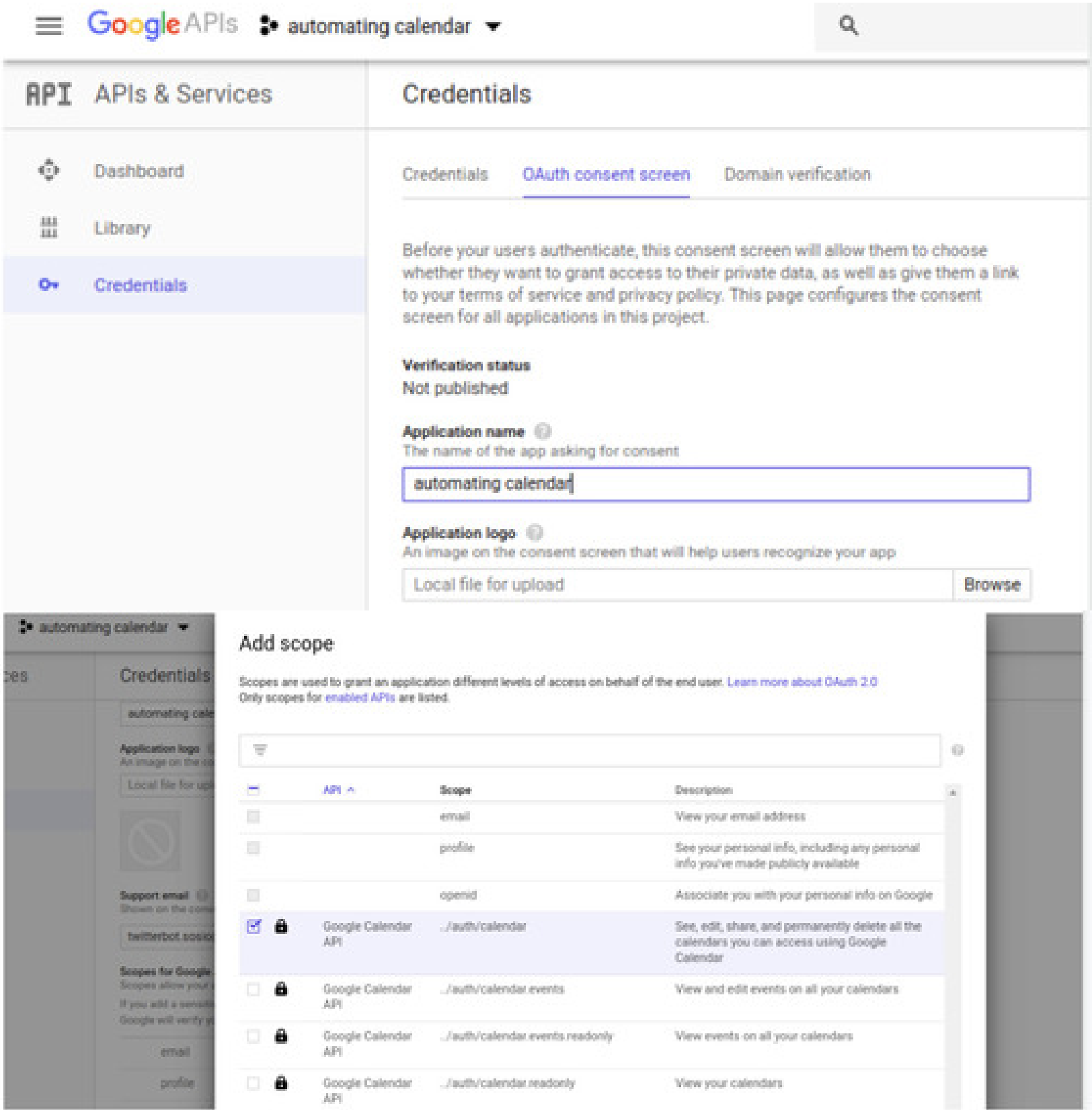
CalDAV API
Google
The CalDAV API lets you manage your Google calendars and events.



The first step is done. We need to now, move over to generate credentials which will be used.

2. How to generate application credentials for Google Calendar API?

The first thing to do is to go to the *OAuth Consent Screen* tab, provide it an application name, pick all the necessary scopes that you would like to obtain permissions for. For instance, if you need only read permissions or edit, specify the checkboxes as per your need.



Once the application scope and consent screen is setup, we proceed to generating the credentials. On the credentials tab, click the Create credentials, and then select the OAuth client ID, specify the major platform where you will be using the applicatin, if you are not sure, select the **Other** box. This generates the **Client ID** and **Client Secret Keys** but we want them as in a **JSON** file so download and save the file. We will need this in next step.

example.com

Type in the domain and press Enter to add it

Application Homepage link

Shown on the consent screen. Must be hosted on an Authorized Domain.

https:// or http://

Application Privacy Policy link

Shown on the consent screen. Must be hosted on an Authorized Domain.

https:// or http://

Application Terms of Service link (Optional)

Shown on the consent screen. Must be hosted on an Authorized Domain.

https:// or http://

Save

Submit for verification

Cancel

Google APIs

automating calendar

API

APIs & Services

Credentials

Dashboard

Library

Credentials

Credentials

OAuth consent screen

Domain verification

Create credentials

Delete

API key

Identifies your project using a simple API key to check quota and access

OAuth client ID

Requests user consent so your app can access the user's data

Service account key

Enables server-to-server, app-level authentication using robot accounts

Help me choose

Asks a few questions to help you decide which type of credential to use

Google APIs

automating calendar

Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more

Application type

Web application

Android [Learn more](#)

Chrome App [Learn more](#)

iOS [Learn more](#)

Other

Name

calendar automation

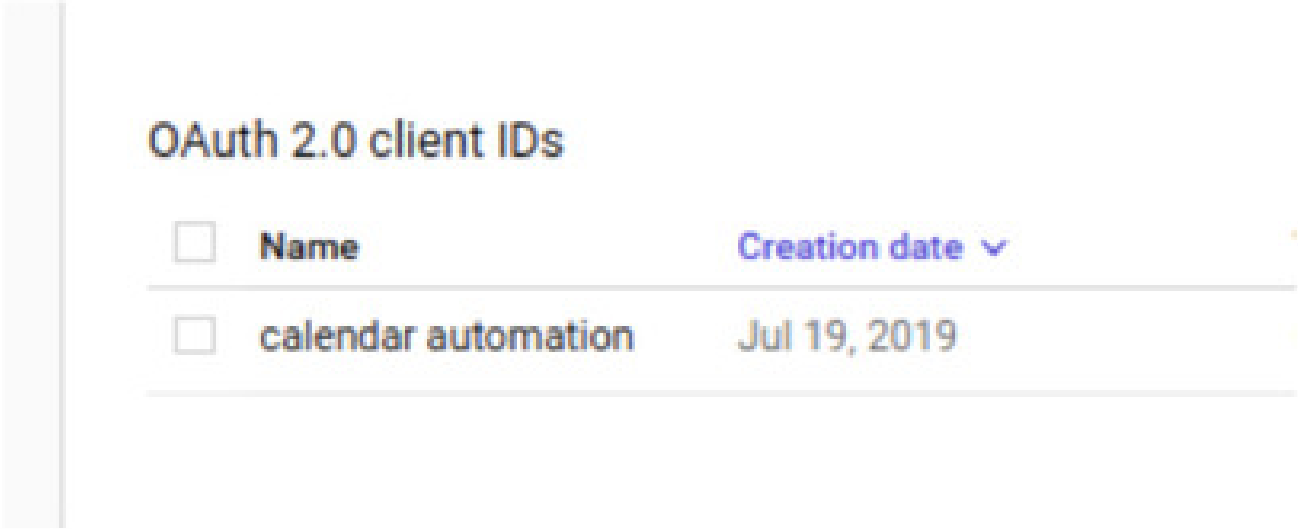
Create

Cancel

https://karenapp.io/articles/how-to-automate-google-calendar-with-python-using-the-calendar-api/

4/11

Find the downloaded file and rename it as `credentials.json`, save it preferably in the same directory, where you will start writing your code.



3. How to start using Google OAuth SDK for Python?

To connect to the Google Calendar API, google has released SDK, we will use `pip` to install the necessary library

```
pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib
```

Set up OAuth, let us create a file named `cal_setup.py` with the code below:

```
import datetime
import pickle
import os.path
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request

# If modifying these scopes, delete the file token.pickle.
SCOPES = ['https://www.googleapis.com/auth/calendar']

CREDENTIALS_FILE = 'path_to_file/credentials.json'

def get_calendar_service():
    creds = None
    # The file token.pickle stores the user's access and refresh tokens, and is
    # created automatically when the authorization flow completes for the first
    # time.
    if os.path.exists('token.pickle'):
        with open('token.pickle', 'rb') as token:
            creds = pickle.load(token)
    # If there are no (valid) credentials available, let the user log in.
    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow = InstalledAppFlow.from_client_secrets_file(
                CREDENTIALS_FILE, SCOPES)
            creds = flow.run_local_server(port=0)

    # Save the credentials for the next run
    with open('token.pickle', 'wb') as token:
        pickle.dump(creds, token)

    service = build('calendar', 'v3', credentials=creds)
    return service
```

4. How to Fetch list of all Calendars from a User Account using Calendar API

Let us create a file with name `list_calendars.py` in the same directory as that of `cal_setup.py` with the snippet below:

```

from cal_setup import get_calendar_service

def main():
    service = get_calendar_service()
    # Call the Calendar API
    print('Getting list of calendars')
    calendars_result = service.calendarList().list().execute()

    calendars = calendars_result.get('items', [])

    if not calendars:
        print('No calendars found.')
    for calendar in calendars:
        summary = calendar['summary']
        id = calendar['id']
        primary = "Primary" if calendar.get('primary') else ""
        print("%s\t%s\t%s" % (summary, id, primary))

if __name__ == '__main__':
    main()

```

The above code is calling the Calendar API with the credentials that have already been setup, it will fetch all the calendars.

When the above code is run for the first time, a google auth link will be opened in default browser seeking permission to access the calendar, after the consent is provided a token is saved which can be used for making requests in future to the Google Calendar API.

Once the code has run, in the terminal you will see on the console the names of all the calendars that you have created.

5. How to List all the Google Calendar Events

To list all the Calendar events, let us create a file `list_events.py` in the same directory as `cal_setup.py` with the snippet as mentioned below

```

import datetime
from cal_setup import get_calendar_service

def main():
    service = get_calendar_service()
    # Call the Calendar API
    now = datetime.datetime.utcnow().isoformat() + 'Z' # 'Z' indicates UTC time
    print('Getting List o 10 events')
    events_result = service.events().list(
        calendarId='primary', timeMin=now,
        maxResults=10, singleEvents=True,
        orderBy='startTime').execute()
    events = events_result.get('items', [])

    if not events:
        print('No upcoming events found.')
    for event in events:
        start = event['start'].get('dateTime', event['start'].get('date'))
        print(start, event['summary'])

if __name__ == '__main__':
    main()

```

The snippet above lists all the events of `primary` calendar. In case you want any other calendar's events to be listed, provide a different calendar id.

The console will reflect all the events that have been published.

6. How to Create an event using the Google Calendar API?

To create an event in a calendar, create a file `create_event.py` in the same directory where `cal_setup.py` resides, paste the snippet below :

```
from datetime import datetime, timedelta
from cal_setup import get_calendar_service

def main():
    # creates one hour event tomorrow 10 AM IST
    service = get_calendar_service()

    d = datetime.now().date()
    tomorrow = datetime(d.year, d.month, d.day, 10)+timedelta(days=1)
    start = tomorrow.isoformat()
    end = (tomorrow + timedelta(hours=1)).isoformat()

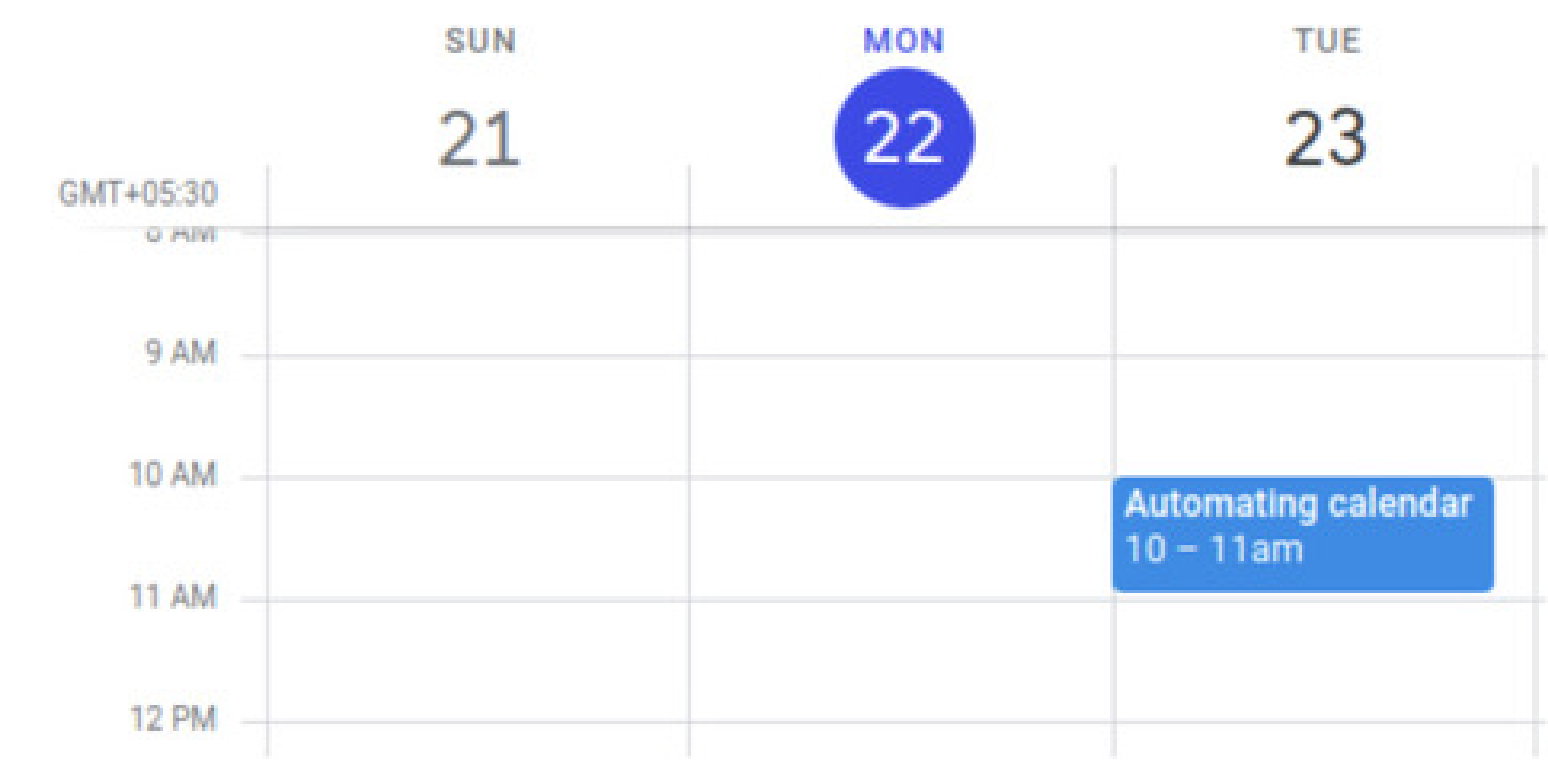
    event_result = service.events().insert(calendarId='primary',
        body={
            "summary": 'Automating calendar',
            "description": 'This is a tutorial example of automating google calendar with python',
            "start": {"dateTime": start, "timeZone": 'Asia/Kolkata'},
            "end": {"dateTime": end, "timeZone": 'Asia/Kolkata'},
        }
    ).execute()

    print("created event")
    print("id: ", event_result['id'])
    print("summary: ", event_result['summary'])
    print("starts at: ", event_result['start']['dateTime'])
    print("ends at: ", event_result['end']['dateTime'])

if __name__ == '__main__':
    main()
```

The above code creates an event in the primary calendar at 10 AM the next day in IST timezone. The console reflects the output. Also save the **event ID** for later use.

The same can be verified in the Google Calendar as well :



There are other options that can be passed, for details, please refer the [Google Calendar API Documentation] (https://developers.google.com/resources/api-libraries/documentation/calendar/v3/python/latest/calendar_v3.events.html#insert).

7. How to Update an Event in Google Calendar using Python?

To update an event in your Google Calendar, create a file **update_event.py** in the same directory as that of **cal_setup.py** and paste the snippet from below :

```
from datetime import datetime, timedelta
from cal_setup import get_calendar_service

def main():
    # update the event to tomorrow 9 AM IST
    service = get_calendar_service()

    d = datetime.now().date()
    tomorrow = datetime(d.year, d.month, d.day, 9)+timedelta(days=1)
    start = tomorrow.isoformat()
    end = (tomorrow + timedelta(hours=2)).isoformat()

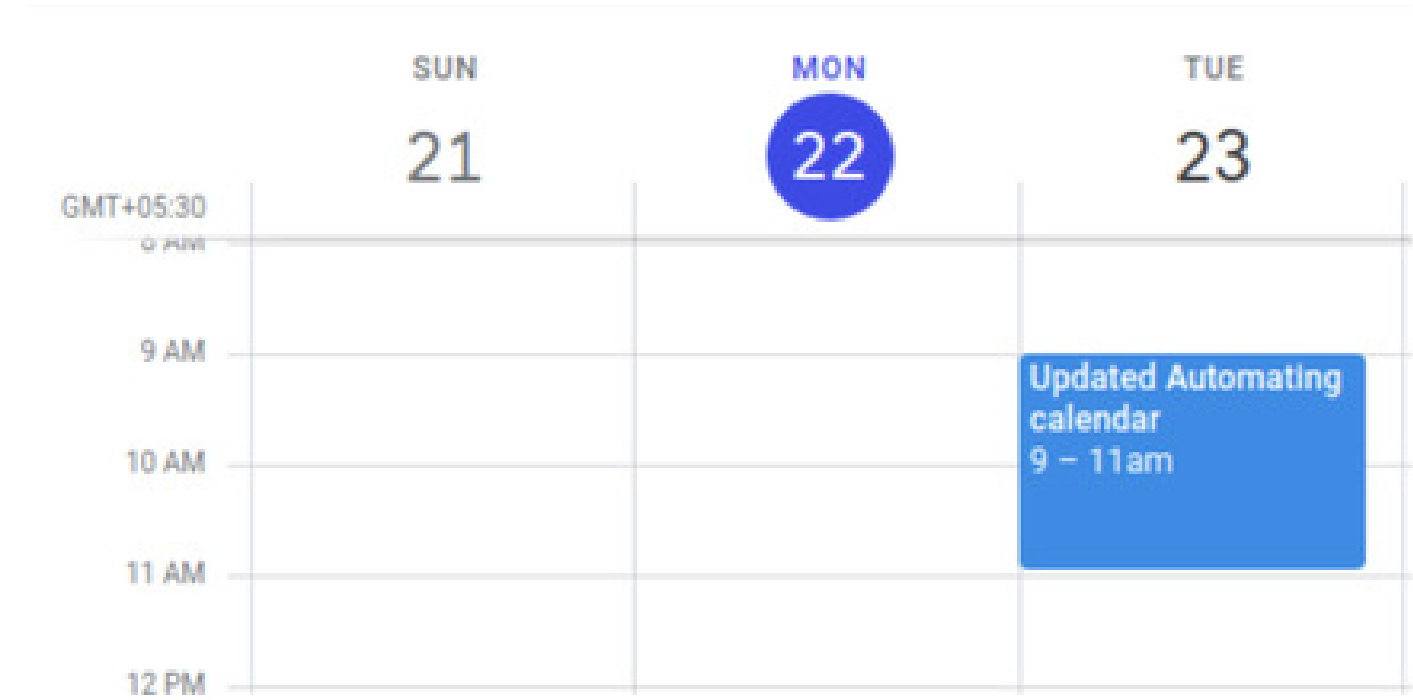
    event_result = service.events().update(
        calendarId='primary',
        eventId='<place your event ID here>',
        body={
            "summary": 'Updated Automating calendar',
            "description": 'This is a tutorial example of automating google calendar with python, updated time.',
            "start": {"dateTime": start, "timeZone": 'Asia/Kolkata'},
            "end": {"dateTime": end, "timeZone": 'Asia/Kolkata'},
        },
    ).execute()

    print("updated event")
    print("id: ", event_result['id'])
    print("summary: ", event_result['summary'])
    print("starts at: ", event_result['start']['dateTime'])
    print("ends at: ", event_result['end']['dateTime'])

if __name__ == '__main__':
    main()
```

The snippet above updates the event created in the primary calendar, from its original schedule of 10 AM to 9 AM, it also updates the summary and description. The output console reflects the changes.

The same can be verified in Google Calendar:



8. How to Delete and Event in Google Calendar using Python

To delete a scheduled event in the Google Calendar, create a file `delete_event.py` in the same directory, where all other files are living, and paste the snippet below.


```
## Script to delete events from google calendar

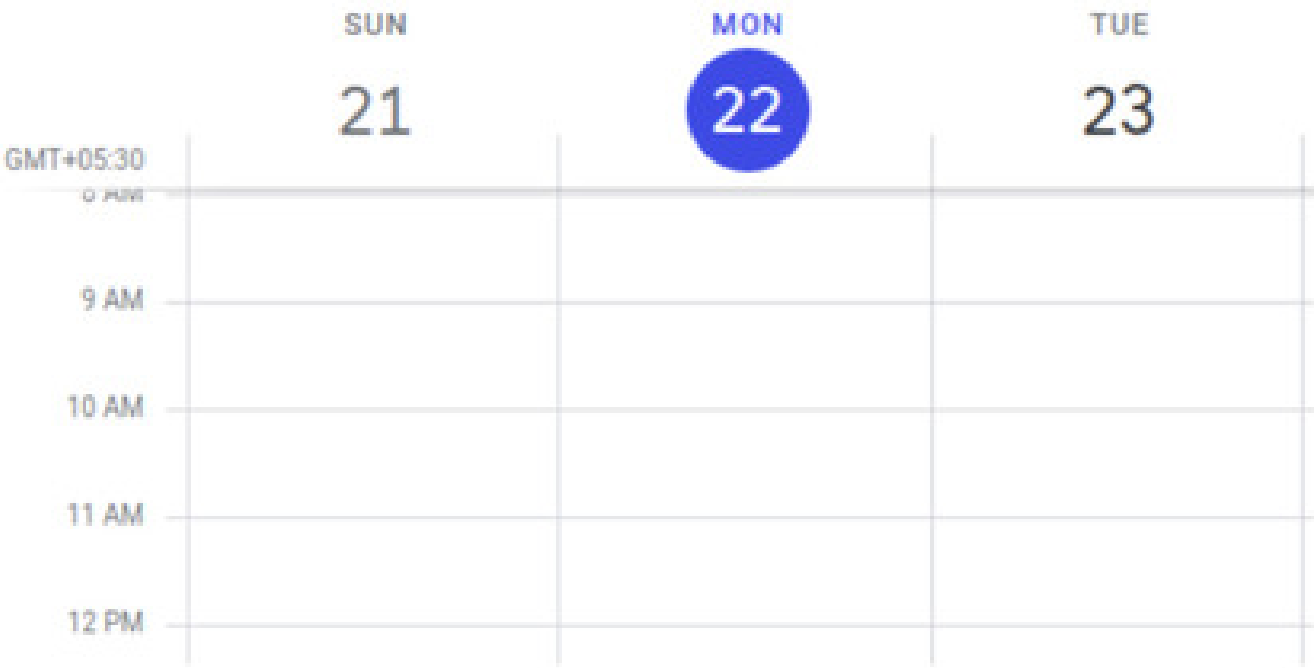
from cal_setup import get_calendar_service

def main():
    # Delete the event
    service = get_calendar_service()
    try:
        service.events().delete(
            calendarId='primary',
            eventId='<place your event ID here>',
        ).execute()
    except googleapiclient.errors.HttpError:
        print("Failed to delete event")

    print("Event deleted")

if __name__ == '__main__':
    main()
```

The above snippet deletes an event from the primary calendar. On checking our Google Calendar, the same event seems to have been deleted.



I hope the snippet was useful, the codebase can be found on [our Github](#). Let us know, what else would you like us to write about in the comment section.

Related article

- [How to Sync Facebook Events to Google Calendar](#)
- [15 Best Appointment Scheduling Softwares for Small Business in 2019](#)
- [Lets Play With Google Calendar](#)

Table of Contents

- [1. How to Create a project in Google Developer Console?](#)
- [2. How to generate application credentials for Google Calendar API?](#)
- [3. How to start using Google OAuth SDK for Python?](#)
- [4. How to Fetch list of all Calendars from a User Account using Calendar API](#)
- [5. How to List all the Google Calendar Events](#)
- [6. How to Create an event using the Google Calendar API?](#)
- [7. How to Update an Event in Google Calendar using Python?](#)
- [8. How to Delete and Event in Google Calendar using Python](#)

Recent Articles

[Zoom Integration](#) August 11, 2020 [Latest Trends in Appointment Scheduling Software](#) July 4, 2020

[How to Plan a Summer Camp](#) June 26, 2020 [9 Strategies for the Flow State of Mind](#) June 18, 2020

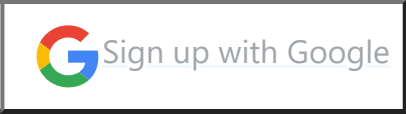
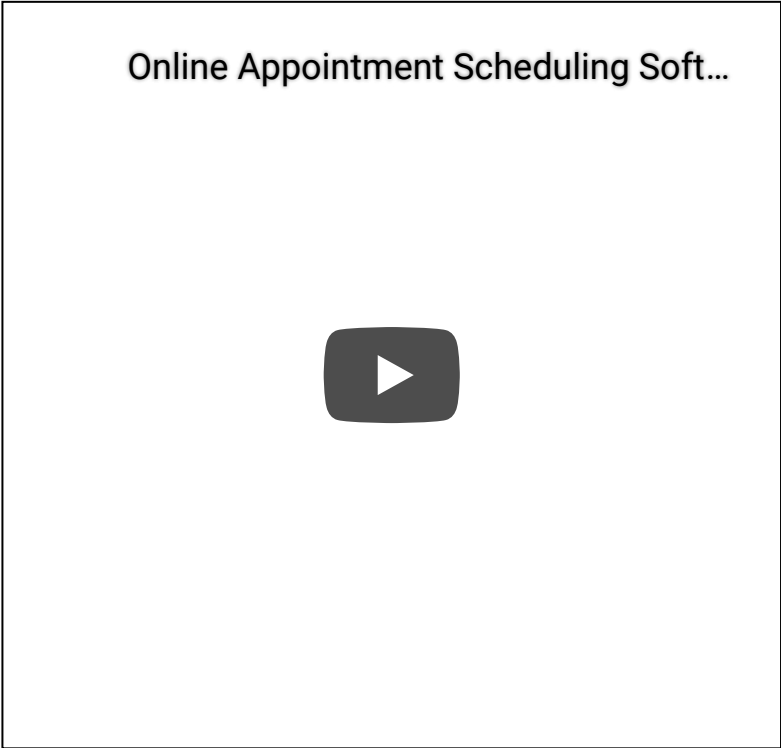
[5 Tips to Simplify Appointment Scheduling Software](#) June 10, 2020

[Calendly vs ScheduleOnce Feature and Pricing Comparison : 2020](#) May 20, 2020

[Calendly vs Setmore Feature and Pricing Comparison : 2020](#) May 12, 2020

Popular Tags

Acuity scheduling, Alternatives, Appointments, Appointy, Booking, Books, Calendar, Calendly, Calendly alternatives, Calendly vs, Camping, Comparisons, Customer relationship, Doodle, Education institutions, Facebook event, Google calendar, Integration, Integrations, Leisure, Management, Non profit, Online appointment scheduling, Online appointments, Online scheduling, Plan, Plugins, Productivity, Recruiters, Scheduleonce, Scheduling, Setmore, Shopify, Small business, Tech, Trends, Tutorial, Wordpress, Youcanbookme, Zoom,



Related Articles

[Lets Play With Google Calendar](#) April 29, 2019

[How to Sync Facebook Events to Google Calendar](#) June 29, 2019

Karen	Products	From our Blog	Comparisons
Home	Interview Scheduling Software	Connect Karen with 1500+ Apps	Calendly Alternative
How It Works	Yoga Studio Software	using Zapier	Doodle Alternative
Benefits	Volunteer Scheduling Software	Calendly vs Doodle Comparison	Scheduleonce Alternative
Features	Chrome Extension	and review	Setmore Alternative
Pricing	Appointment Scheduling	21 Wonderful Benefits of Online	BookLikeABoss Alternative
FAQ	Wordpress Plugin	Appointment	Rezdy Alternative
Contact		SchedulingSoftware	
Blog		25 Questions to ask before you	
Sign Up with Google			

chose Online Appointment

Scheduling Software