

Google Calendar Setup Guide For Python

Beforehand

- Have Python 3.x installed
- Be signed into the desired google calendar account through your web browser
- Install the Google Client Library:

```
o pip install --upgrade google-api-python-client
```

Setting up credentials and allowing API access

- A. Go to <https://console.developers.google.com/start/api?id=calendar>
- B. Select "Create a project" from the scroll-down menu bar
- C. Click **Continue**
- D. Click **Go to credentials**
- E. On the "Add credentials to your project" page, click **cancel**
- F. Towards the middle-top of the page, select the tab, "OAuth consent screen"
- G. Here, enter an email address and a product name, for example, use the desired email address for the google calendar and a product name that optionally makes sense
- H. Click **save** after these two fields have been entered
- I. Click the credentials tab in the middle-top of the page if you're not already there, and click **Create credentials** and select "OAuth client ID"
- J. Checkmark the "Other" application type selection
- K. After, an input text box appears, titled 'Name'; Type "Google Calendar API Quickstart" into the field and click **Create**
- L. The OAuth client ID window should appear, just click ok
- M. To the right of the "Client ID" field, click **Download JSON**
- N. Move this file to the correct directory for the code (shown later) for use and rename it "client_secret.json"

Code

At this point, the json file should be in your working directory. Create a new python script titled, “quickstart.py”. This code and more can be found on our GitHub, in an easier to view format, link here:

https://github.com/bdaszkiewicz/info_display

Additionally, the code will be shown below here:

```
from
__future__
import
print_function

import httpplib2
import os

from apiclient import discovery
from oauth2client import client
from oauth2client import tools
from oauth2client.file import Storage

import datetime

try:
    import argparse
    flags =
    argparse.ArgumentParser(parents=[tools.argparser]).parse_args()
except ImportError:
    flags = None

# If modifying these scopes, delete your previously saved credentials
# at ~/.credentials/calendar-python-quickstart.json
SCOPES = 'https://www.googleapis.com/auth/calendar.readonly'
CLIENT_SECRET_FILE = 'client_secret.json'
APPLICATION_NAME = 'Google Calendar API Python Quickstart'
```

```
def get_credentials():
    """Gets valid user credentials from storage.
    If nothing has been stored, or if the stored credentials are invalid,
    the OAuth2 flow is completed to obtain the new credentials.
    Returns:
        Credentials, the obtained credential.
    """
    home_dir = os.path.expanduser('~')
    credential_dir = os.path.join(home_dir, '.credentials')
    if not os.path.exists(credential_dir):
        os.makedirs(credential_dir)
    credential_path = os.path.join(credential_dir,
                                   'calendar-python-quickstart.json')

    store = Storage(credential_path)
    credentials = store.get()
    if not credentials or credentials.invalid:
        flow = client.flow_from_clientsecrets(CLIENT_SECRET_FILE, SCOPES)
        flow.user_agent = APPLICATION_NAME
        if flags:
            credentials = tools.run_flow(flow, store, flags)
        else: # Needed only for compatibility with Python 2.6
            credentials = tools.run(flow, store)
        print('Storing credentials to ' + credential_path)
    return credentials

def main():
    """Shows basic usage of the Google Calendar API.
    Creates a Google Calendar API service object and outputs a list of the
    next
    10 events on the user's calendar.
    """
    credentials = get_credentials()
    http = credentials.authorize(httplib2.Http())
    service = discovery.build('calendar', 'v3', http=http)

    now = datetime.datetime.utcnow().isoformat() + 'Z' # 'Z' indicates UTC
    time
```

```
#print(datetime.datetime.today()) -- Check :)
tomorrow = datetime.datetime.today() + datetime.timedelta(days=1)
tomorrow = str(tomorrow)
tomorrow = tomorrow.replace(" ", "T") + 'Z'

#tomorrow = tomorrow.strftime('%Y-%m-%dT10:00:00Z') # now is
datetime.. not a string :(
print('Tomorrow: ')
print(tomorrow)
print('\nGetting the upcoming 10 events ... Getting events of: \n' +
now + '\n')
eventsResult = service.events().list(
    calendarId='primary', timeMin=now, maxResults=10,
singleEvents=True,
    #timeMax="2018-03-21T10:00:00-06:00",
    timeMax=tomorrow,
    orderBy='startTime').execute()
events = eventsResult.get('items', [])
#print(events, "events")

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

    print('\n', start, event['summary'])

    print('\n end:', end, '\n')

event_title = [event['summary']]
# event['summary'] is the name of the event #
print("event title:", event['summary'])

today_day = start[0:10]
print("today day:", today_day)
# today_day is the date for the current day #
```

```
time_start = start[11:19]
print("time start:", time_start)
    # event_start is the start time for the event #

time_end = end[11:19]
print("time end:", time_end)

try:
    color_ID = event['colorId']
    print("ColorId:", color_ID)
except:
    color_ID = 5
    print("ColorId:", color_ID)

#return(event['summary'],event_start)
    #clear this statement & implement into section where it's
called#

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

There are a few print statements to check values and such when coordinating specified dates to be shown.

For the final setup of this part, after having this code in your working directory and renaming it to “quickstart.py”, simply run the sample by issuing the command:

```
python quickstart.py
```

This will open a window to ask for allowance of google credentials. Once accepted, the following code will return values in a few test print statements. This code can be amended to return values in the code already or other values in the google python API.

References

<https://developers.google.com/calendar/quickstart/python>

<https://developers.google.com/calendar/v3/reference/events/list>

<https://developers.google.com/calendar/v3/reference/events>