Title: Personal Daily Meeting Viewer using Google Calendar API.

Use Case: This demo personal project was developed as a lightweight, customizable solution that allows users to view their calendar (e.g. Google Calendar) events for any selected day via a local web interface or could be hosted in AWS cloud.

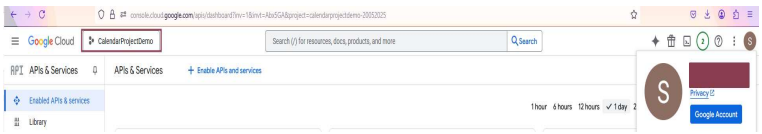
The solution could be applied to Outlook 365 and somethings in the code needs to be changed for Outlook POC. In my case I have used Google Calendar.

For example:
1. A manager can share a local dashboard with a PA to review meetings for specific dates.

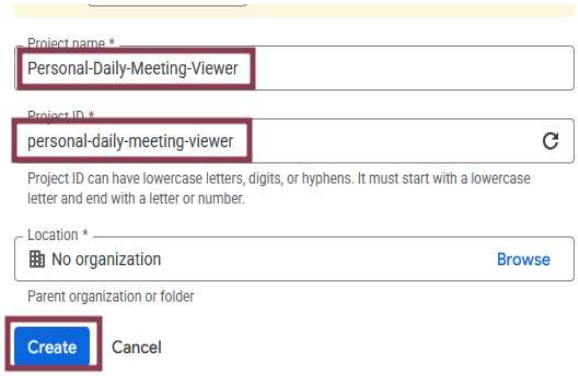
2. A remote engineer can check the daily meetings in a browser without opening Gmail.

Steps:

1. Enable the Calendar API
2. Navigate to the Link: https://console.developers.google.com/
3. Login with your personal Gmail account
4. You can create a new project or select an existing project
5. Would recommend a new project, you can delete after the proof of concept



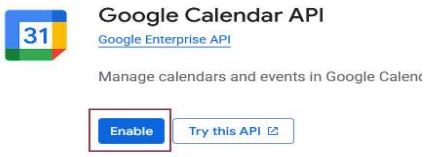
6. Provide a Project Name



7. Once the project has been created, make sure the project has been selected

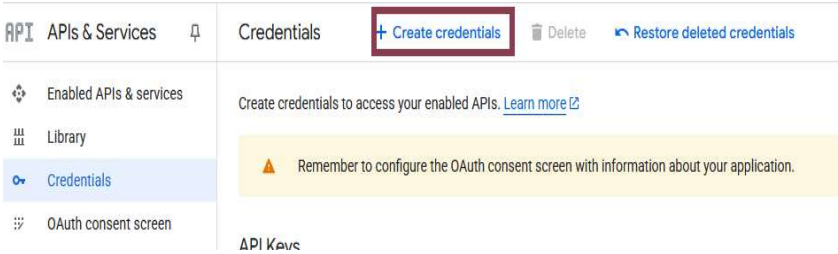
8. Now click on Library
9. Locate Google Calendar API

10. Enable the API



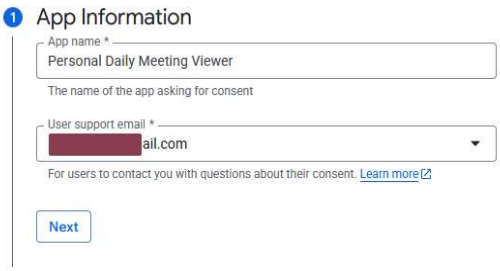
11. Now, click on create credentials

12. Select OAuth Client ID (This is personal project)

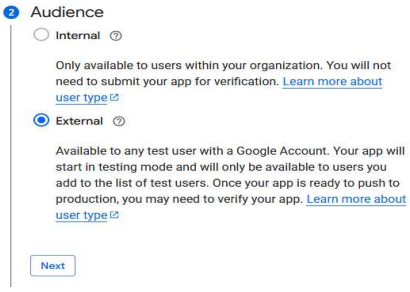


13. Click on Configure Consent Screen

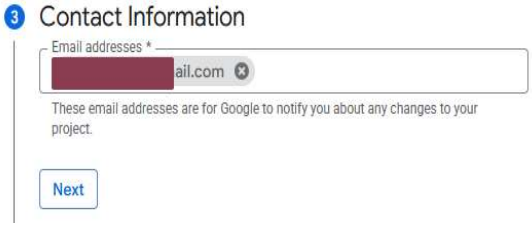
14. In the overview provide the following name
    a. App Information
    b. Your personal email address
    c. Click on Next



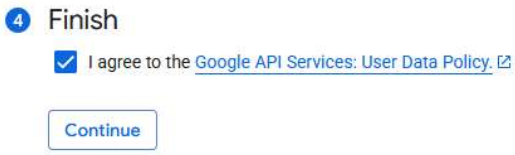
15. This is only test mode, and I will be the only user testing it. So, select external and click on Next



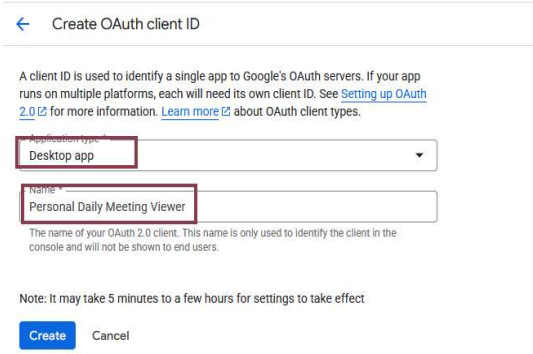
16. In the next screen, provide your personal Gmail address



17. Check the box and click on continue



18. Click on Create.
19. Now create OAuth Client
20. Select Desktop App
21. Provide a Name
22. Click on Create



23. Download the JSON in a secure area
24. Make a copy
25. Rename one to credentials.json to shorten the file name
26. Now comes the interesting path
27. Open jupyter notebook
28. The credentials json will be in the same path as your jupyter notebook
29. Install the required libraries
    a. !pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib
        i. Httplib2 will handle the HTTP requests
        ii. Oauthlig will handle the Oauth token
        iii. Python client is the API clinent library

30. Initially, the project demo was to integrate with WhatsApp and for the sandbox to get enabled will take 72 hours. This is a trial version. Will plan next week integration with whatsapp and adopted a different approach

31. The approach was to launch locally and also installed the following library

```
!pip install flask flask-cors
```

32. Run the python code to authenticate with Google Calendar API

33. Each portion is broken into smaller pieces just like projects to ensure key result is met
34. If you run into access issue or not authorized pop up message. Remember , while creating the project, selected external for testing purpose and not internal. This error is quite common.
    a. Go to the Google Cloud Console
    b. https://console.cloud.google.com/
    c. Under user type it is set to external and there is no need to publish it because it is a personal project
    d. In the Test Users Section – Add your Personal Gmail address. It will be same one you will use when executing the above python code
    e. Save the changes
    f. Restart the kernel and run the cell again
    g. Once successfully executed, output will be Google Calendar API connected successfully
35. Now fetch and display events using the API from today

```
import pytz
[redacted]
# Call the function
today_events = get_today_events(service)

Found 5 event(s) for today:

2025-05-20T07:00:00+05:30 - [redacted] Training
2025-05-20T10:00:00+05:30 - [redacted] Training
2025-05-20T13:00:00+05:30 - [redacted] Foundation
2025-05-20T14:00:00+05:30 - [redacted] Cloud Certification
2025-05-20T15:30:00+05:30 - [redacted] Assignments
```

36. The today's events are displayed, display the start time and title
37. As mentioned earlier, I am not using WhatsApp messages for the POC, creating a Html form to display the form

```
from flask import Flask, request, jsonify, render_template_string
from threading import Thread
from datetime import datetime, timedelta
import pytz
[redacted]
</html>
"""

@app.route('/')
def index():
```

38. The output will provide a URL. Copy and paste it in the browser.

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
```

39. Select the date and events are displayed