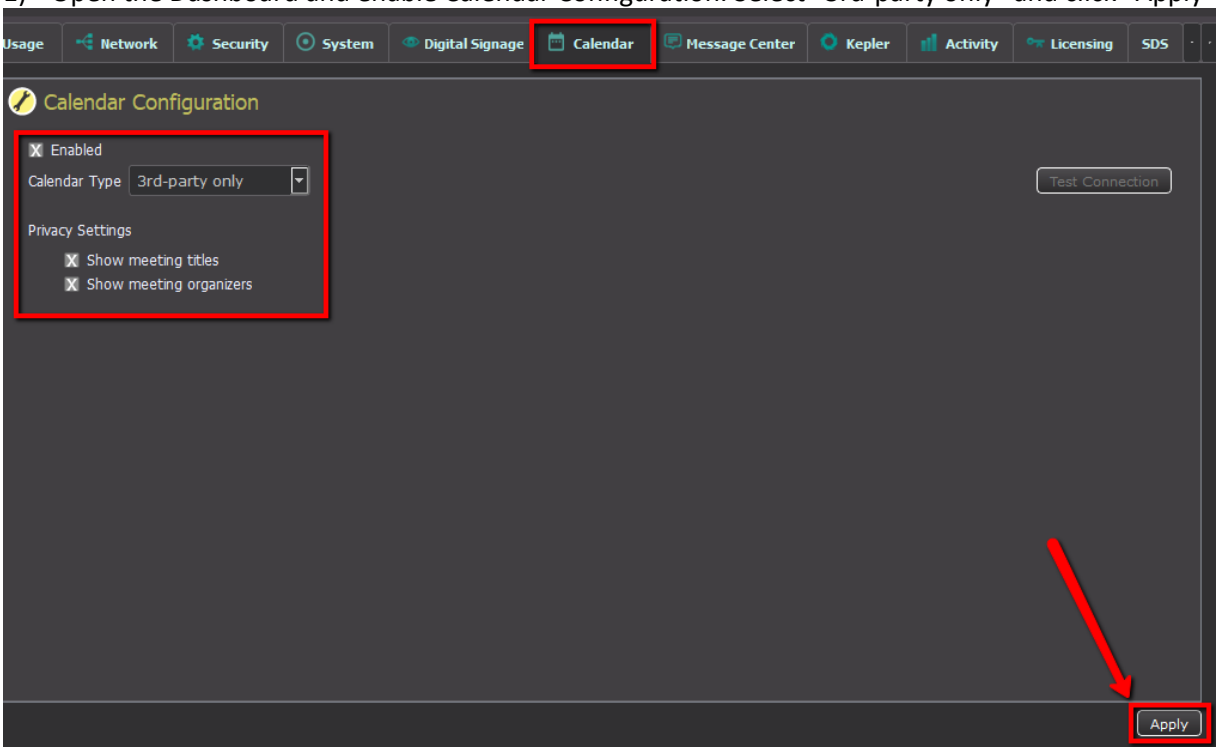
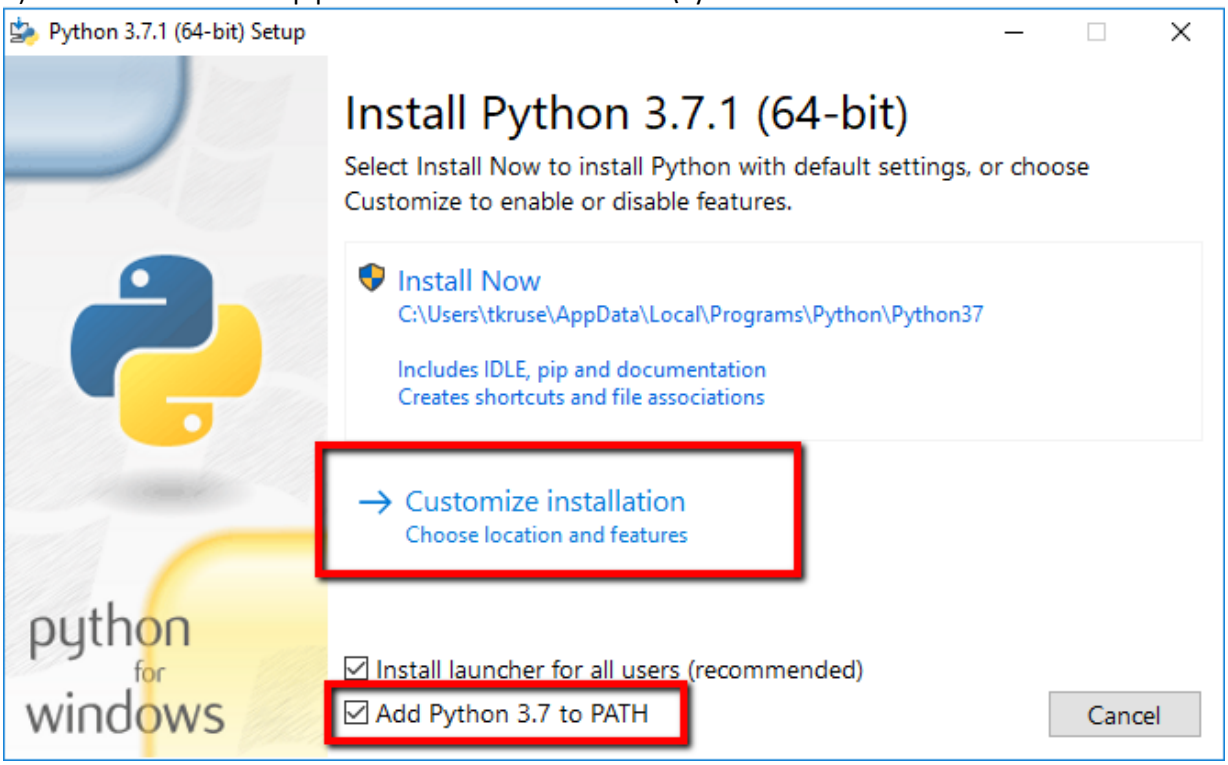
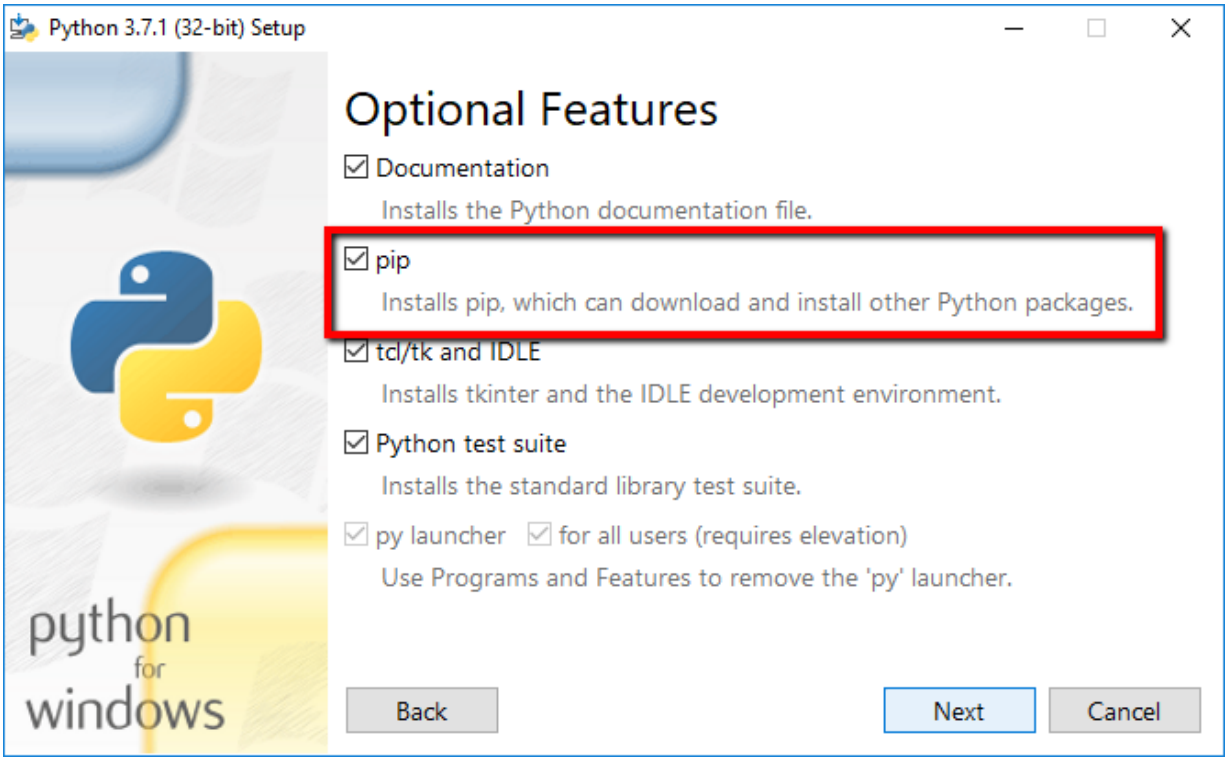
1) Open the Dashboard and enable Calendar Configuration. Select “3rd-party only” and click “Apply”

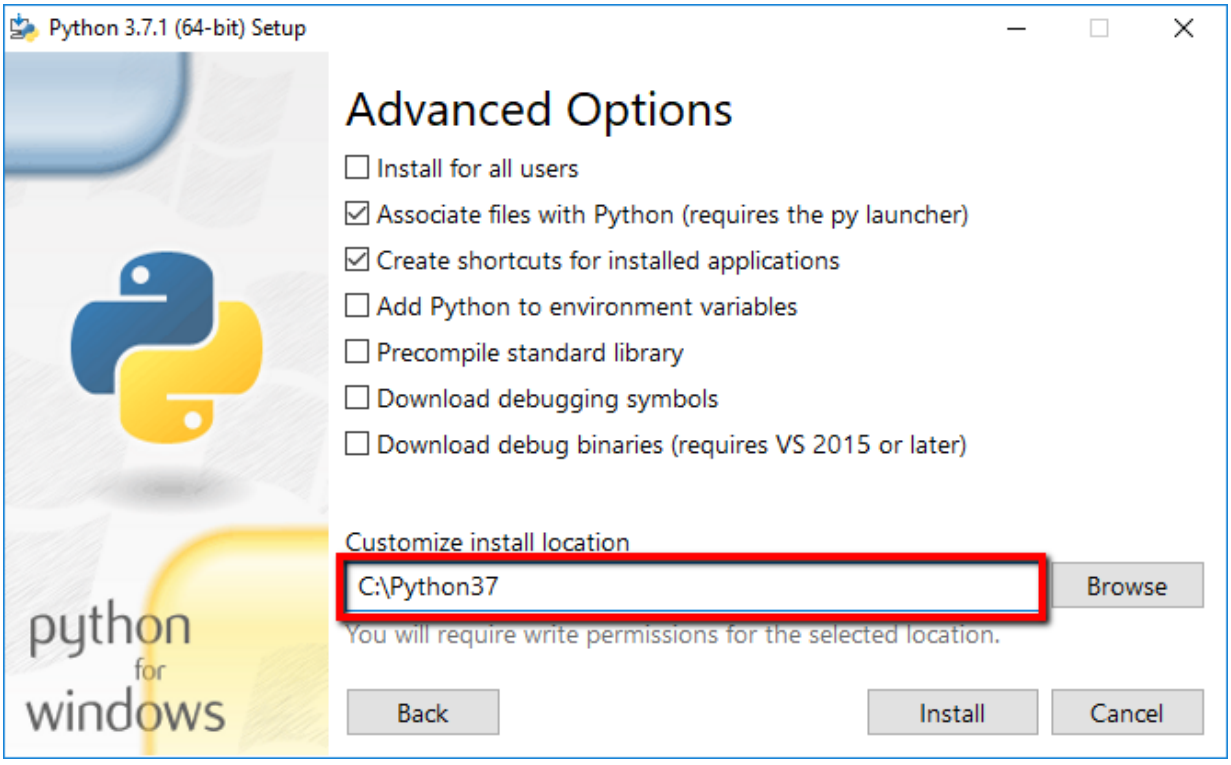


2) Download Python 3.7 from:
https://www.python.org/downloads/

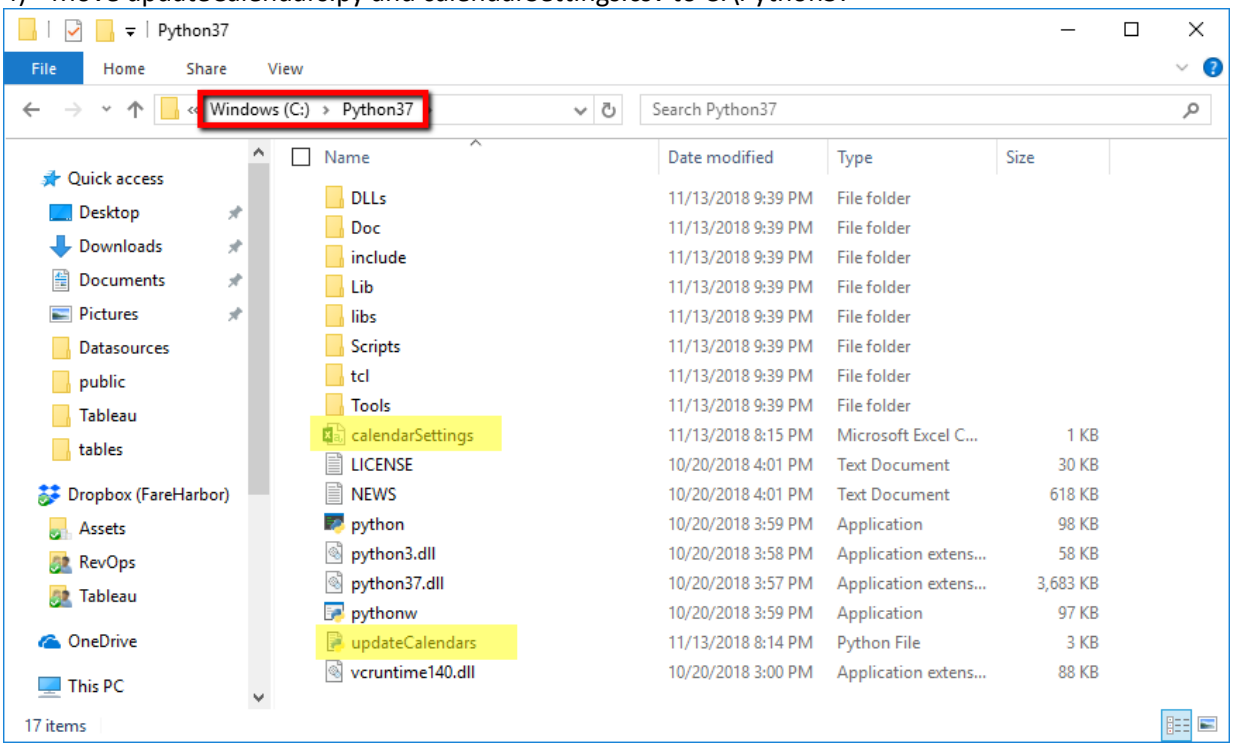
3) Make sure to install pip and set install location to C:\Python37




Python 3.7.1 (32-bit) Setup
Optional Features
Documentation
Installs the Python documentation file.
pip
Installs pip, which can download and install other Python packages.
tcl/tk and IDLE
Installs tkinter and the IDLE development environment.
Python test suite
Installs the standard library test suite.
py launcher
for all users (requires elevation)
Use Programs and Features to remove the 'py' launcher.
python
for
windows
Back
Next
Cancel


Python 3.7.1 (64-bit) Setup
Advanced Options
Install for all users
Associate files with Python (requires the py launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2015 or later)
Customize install location
C:\Python37
Browse
You will require write permissions for the selected location.
python
for
windows
Back
Install
Cancel

4) Move updateCalendars.py and calendarSettings.csv to C:\Python37



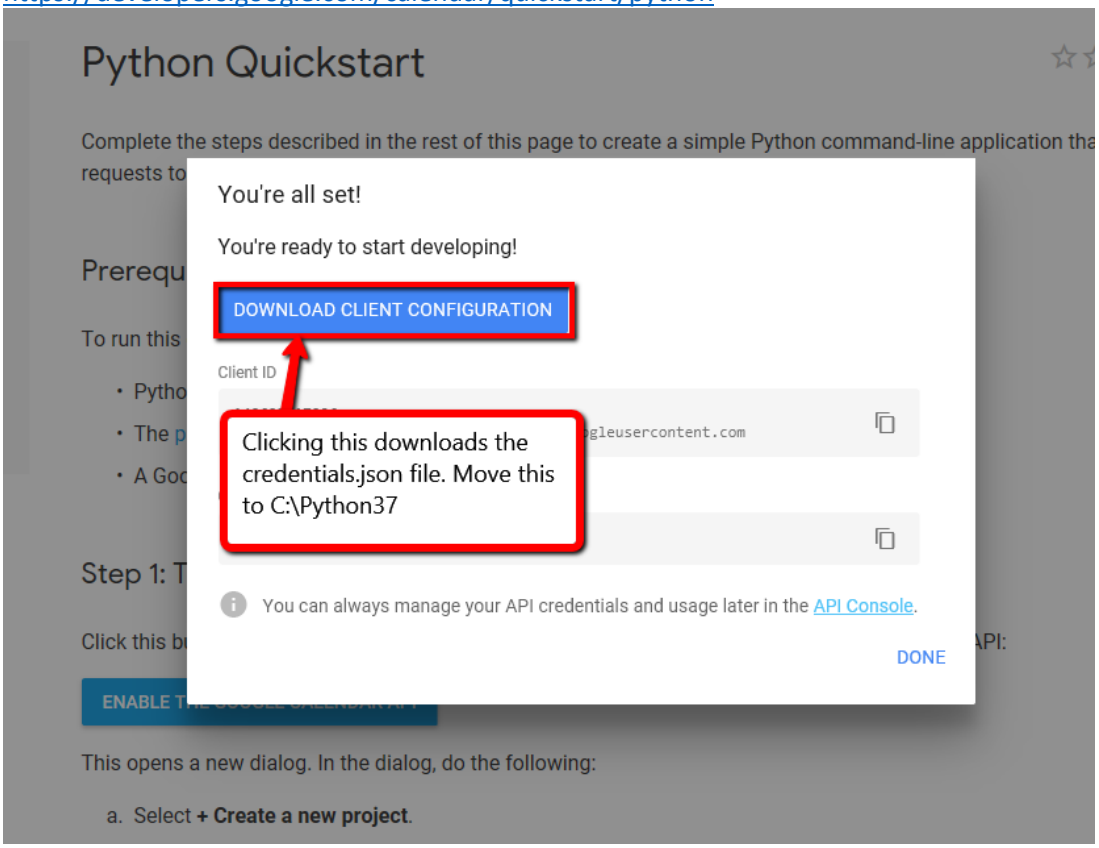
5) Open command prompt and change directory to C:\Python37:
>>cd C:\Python37

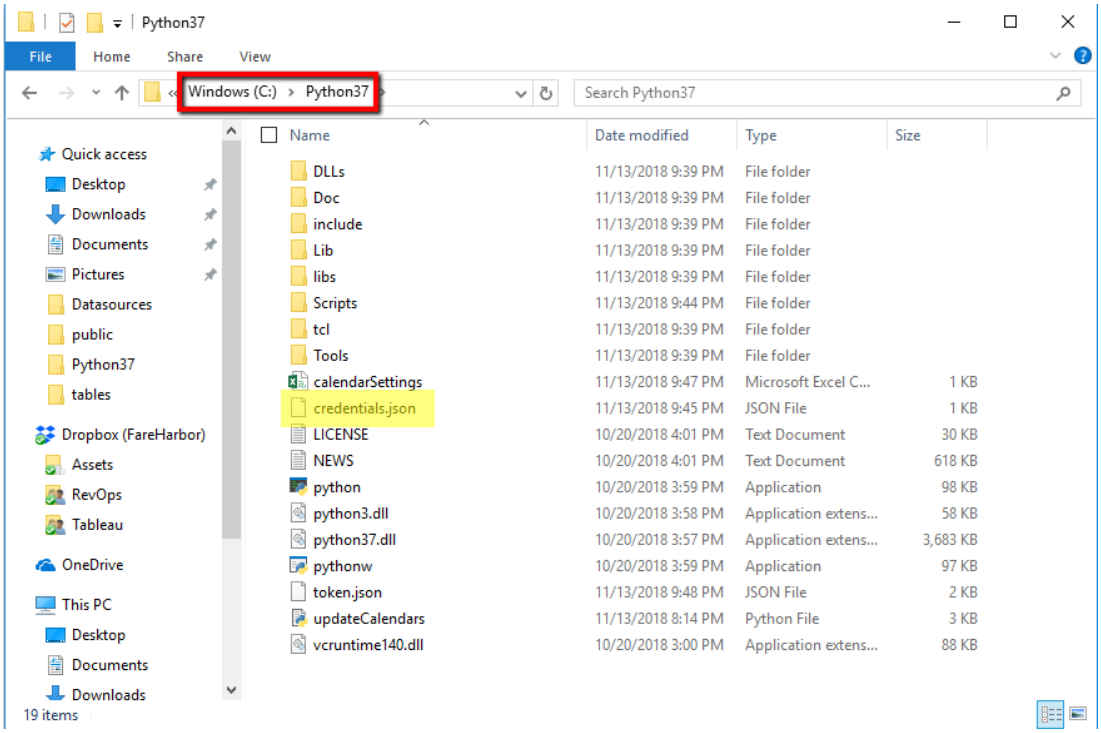
6) Upgrade pip (if applicable)
C:\Python37>python -m pip install --upgrade pip

7) Install the requests package:
C:\Python37>python -m pip install requests

8) Install the Google Client Library
C:\Python37>python -m pip install --upgrade google-api-python-client

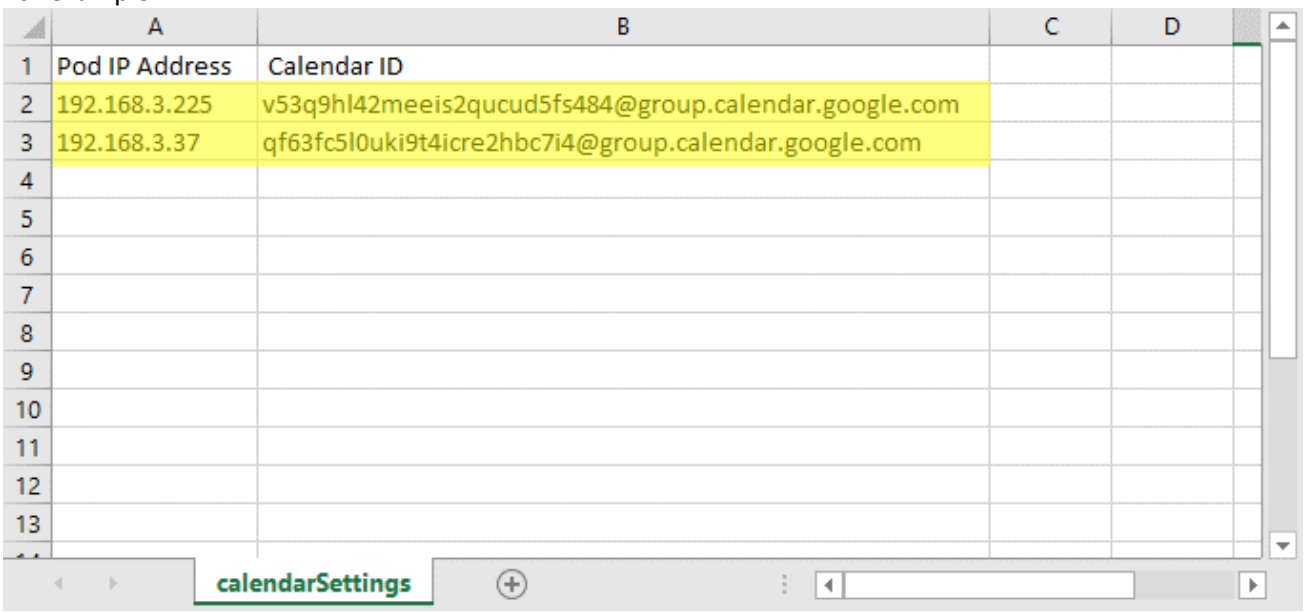
9) Open following page (in Chrome for best results) and complete step 1:
https://developers.google.com/calendar/quickstart/python





10) Install the Google client library:
C:\Python37>python -m pip install --upgrade oauth2client

11) Update calendarSettings.csv with Pod IP addresses and corresponding calendar IDs.
For example:



	A	B	C	D
1	Pod IP Address	Calendar ID		
2	192.168.3.225	v53q9hl42meeis2qucud5fs484@group.calendar.google.com		
3	192.168.3.37	qf63fc5l0uki9t4icre2hbc7i4@group.calendar.google.com		
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				

calendarSettings

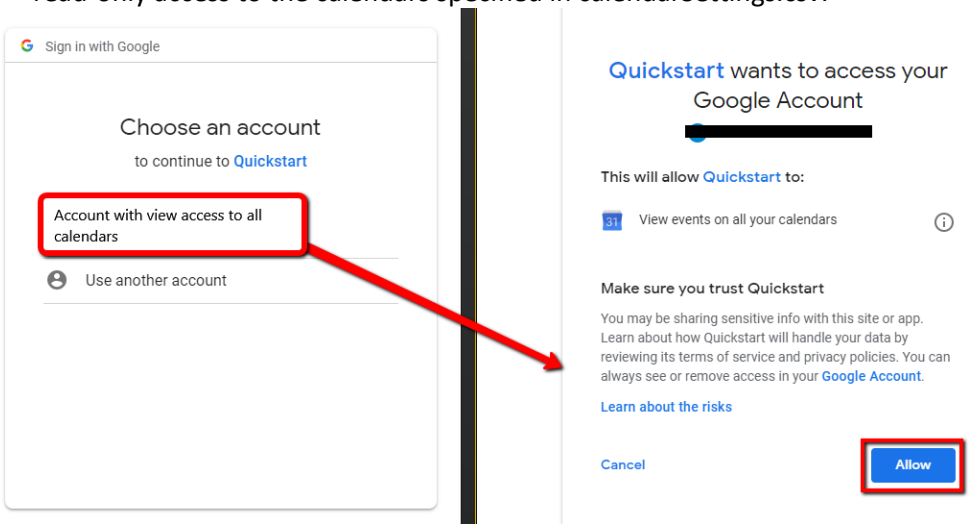
12) Save and close CSV file – if file is left open, script won't be able to access it.

13) Open the updateCalendars.py script in an editor. If there is an admin password on the Pod(s), set ADMIN_PASSWORD_USED to "True" and update the ADMIN_PASSWORD value (must be in quotes). Otherwise, set ADMIN_PASSWORD_USED to "False". Be sure to save the script.

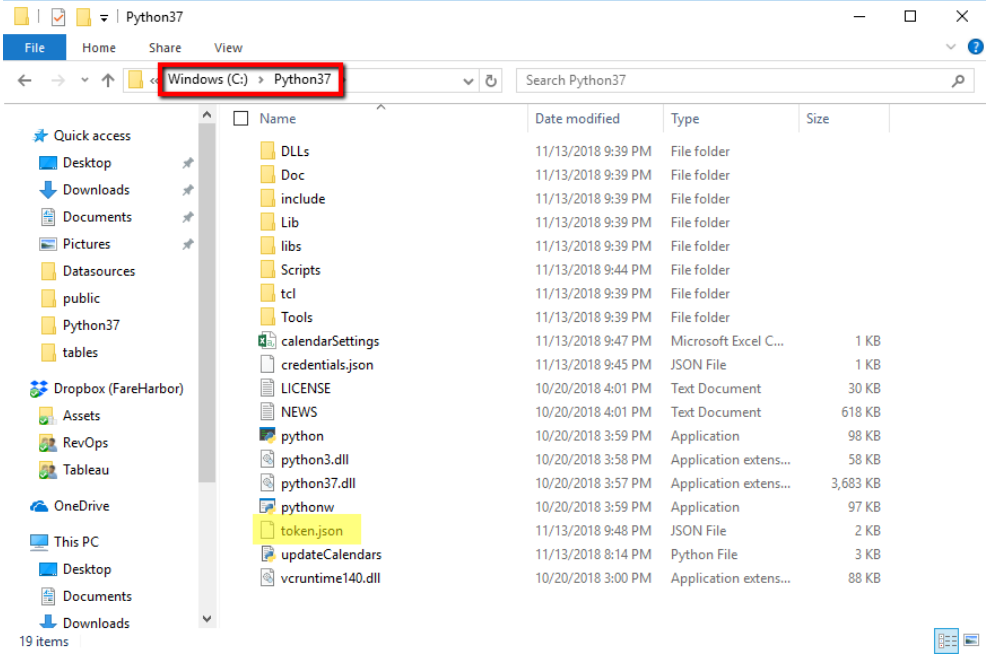
```
13  # Import the modules needed to run the script.
14  from __future__ import print_function
15  import datetime
16  import time
17  import sys
18  import requests
19  import random
20  import csv
21  import getpass
22  from googleapiclient.discovery import build
23  from httplib2 import Http
24  from oauth2client import file, client, tools
25  from random import choice
26  from calendar import timegm
27  import hashlib
28  true=1
29  false=0
30
31  #Script assumes admin password data is uniform across all Pods
32  ADMIN_PASSWORD_USED = True
33  ADMIN_PASSWORD='INSERT PASSWORD HERE' #Content doesn't matter if ADMIN_PASSWORD_USED is false
34
```

14) Run the script once with:
C:\Python37>python updateCalendars.py

15) Initially, a browser will open and you'll be prompted to select an account. Log into an account with read-only access to the calendars specified in calendarSettings.csv.



When authentication completes, token.json will be added to your working directory:



To continually check the calendar for new events, you can create a batch file with the following lines:
cd C:\Python37
python updateCalendars.py

Then, use the Task Scheduler to run the .bat as frequently as needed.