Google Workspace Synchronization Script

Overview

此程式將使用者聯絡方式從 `Google Workspace Directory` 複製到 `Google Contacts`中 專為希望使用 Google Workspace Directory 中的最新信息及時更新 Google 通訊錄的組織而設計。 在 `Google Workspace Directory` 中**添加或刪除成員**後，運行此程式會更新每一個使用的`Google Contacts` 資料 。

Requirements

- Python 3.7 or later
- Google Workspace Admin account with access to `Directory and Contact APIs`
- Google Cloud Platform (GCP) project with the `Admin SDK` and `People API` enabled
- Service Account with `domain-wide authority` in the GCP project

Set up

- google-auth, google-auth-httplib2, and google-auth-oauthlib for authentication
- google-api-python-client for interacting with Google APIs
- google-auth-oauthlib for OAuth 2.0 client-side flow

1. **Enable** the `Admin SDK` and `People API` in your GCP project.
2. Create a `Service Account` in your GCP project and generate a JSON key.
3. `Delegate domain-wide authority` to the Service Account.(The next chapter has teaching)
4. **Install the required dependencies**.

You can install these dependencies using pip:

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

or in project root folder use

```
pip install -r requirements.txt
```

Service Account Creation and Delegation

To create a service account and delegate it domain-wide authority:

1. In the GCP Console, go to the **IAM & Admin** > **Service accounts** page.
2. Click on Create **Service Account** at the top of the page.
3. In the Service account name field, enter a name. The console automatically fills in the `Service account ID` field based on this name.
4. In the Service account description field, enter a description.

5. Click **Create**.
6. The Service account permissions section appears. Click on **Continue**.
7. Click on **Done** to finish creating the service account.
8. Click on the newly created service account to view its details.
9. On the service account **KEYS** page, click on **Add Key, and select JSON**.

To delegate domain-wide authority to the service account:

1. Sign in to your Google Workspace admin account.
2. Go to the **Admin console**.
3. Go to **Security > Access and data control > API controls**.
4. In the **Domain-wide delegation** pane, select Manage Domain-Wide Delegation.
5. Click on Add new.
6. In the Client ID field, enter the **service account's client ID**.
7. In the OAuth Scopes field, enter the required OAuth scopes (https://www.googleapis.com/auth/admin.directory.user, https://www.googleapis.com/auth/contacts) Click on Authorize.

Run Script

在專案根目錄中執行

```
python app.py
```

Explanation of the code

- 導入所需的module並設置log記錄。該腳本會將logging（例如成功添加和刪除聯繫人）記錄到名為 `app.log` 的文件中。

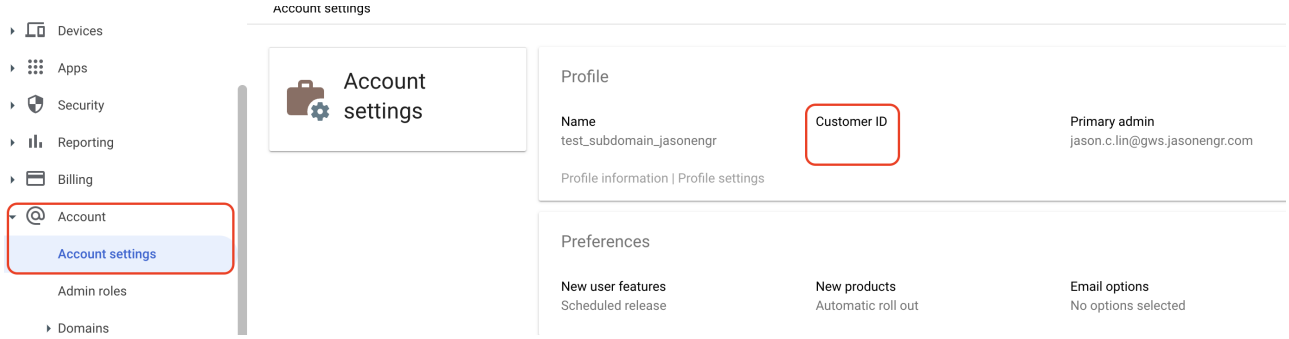
```
import logging
from google.oauth2 import service_account
from googleapiclient.discovery import build

# Configure logging
logging.basicConfig(filename='app.log', filemode='w', level=logging.INFO,
format='%(name)s - %(levelname)s - %(message)s')
```

- 從 JSON 密鑰檔案 loads `Service Account` credentials，
- 定義授權scopes來訪問Directory API 和Contacts API。 **P.S 以下的JSON檔案要放你的憑證**

```
# Path to your Service Account key file
key_file_path = 'YOUR_Service_Account_key .json'
# Load the Service Account credentials
creds = service_account.Credentials.from_service_account_file(
```

```
    key_file_path,scopes=
['https://www.googleapis.com/auth/admin.directory.user',
'https://www.googleapis.com/auth/contacts'])
```

- 利用Administrator建立了一個Service Object 來獲取 Google Workspace 網域中所有使用者的List。
  P.S `Administrator@domain`要放GWS管理員的帳號, `YOUR_CUSTOMER_ID`要放GWS Customer ID




```
# Create a service object for the admin user
admin_creds = creds.with_subject('Administrator@domain')  # Replace with
your Google Workspace admin user
service_admin = build('admin', 'directory_v1', credentials=admin_creds)

# Get all users
results =
service_admin.users().list(customer='YOUR_CUSTOMER_ID').execute()
users = results.get('users', [])
```

- 有了Users List，透過授權後可以得到查詢每個使用者帳戶裡各自的Google帳戶Contacts有哪些聯絡人，透過`for loop`一個個查看與更新

```
# For each user
for user in users:
    # Delegating authority to the service account to impersonate the
current user
    user_creds = creds.with_subject(user['primaryEmail'])
    service = build('people', 'v1', credentials=user_creds)

    # Get the user's contact list
    connections =
service.people().connections().list(resourceName='people/me',
personFields='names,emailAddresses,occupations,organizations,phoneNumbers'
).execute()
    contact_list = connections.get('connections', [])
```

- 獲取到當前帳戶Contacts中聯絡人的電子郵件地址，接著檢查 Google Workspace 網域中的每個用戶，避免將自己資訊加入到Contacts ，

```
    # Get the email addresses of all contacts
    contact_emails = [contact.get('emailAddresses', [{}])[0].get('value')
for contact in contact_list]

    # For each other user
    for contact in users:
        if contact['primaryEmail'] != user['primaryEmail']:
            # Prepare contact info
            contact_info = {
                'names': [{'givenName': contact['name']['fullName']}],
                'emailAddresses': [{'value': contact['primaryEmail']}],
            }
```

- 如果有title和orgUnitPath或聯絡電話也加入到Contacts中

```
            if 'organizations' in contact and contact['organizations']:
                if 'title' in contact['organizations'][0]:
                    contact_info['organizations'] = [{'title':
contact['organizations'][0]['title']}]
                if 'orgUnitPath' in contact:
                    if contact_info.get('organizations'):
                        contact_info['organizations'][0]['name'] =
contact['orgUnitPath']
                    else:
                        contact_info['organizations'] = [{'name':
contact['orgUnitPath']}]
            if 'phones' in contact and contact['phones']:
                contact_info['phoneNumbers'] = [{'value':
contact['phones'][0]['value']}]
```

- 避免更新資料重複加入一樣的聯絡人到Contacts中而做的檢查，確定沒有此人在Contacts中才加入

```
            # Check if the contact already exists in the user's contact
list
            if not any(c for c in contact_list if c.get('emailAddresses',
[{}])[0].get('value') == contact['primaryEmail']):
                # Add to user's contact list

service.people().createContact(body=contact_info).execute()
                # Log successful contact creation
                logging.info(f"Added contact: {contact['primaryEmail']} to
{user['primaryEmail']}")
```

- 更新資料時，發現Directory成員以離職不在清單上時，將每個帳戶的Contacts中剔除此人

```
    # For each contact email
    for contact_email in contact_emails:
        # If the contact email does not exist in the Directory
        if not any(u for u in users if u['primaryEmail'] ==
contact_email):
            # Find the contact in the contact list
            contact = next((c for c in contact_list if
c.get('emailAddresses', [{}])[0].get('value') == contact_email), None)
            if contact:
                # Delete the contact

service.people().deleteContact(resourceName=contact['resourceName']).execu
te()
                 # Log successful contact deletion
                logging.info(f"Deleted contact: {contact_email} from
{user['primaryEmail']}")
```