

# G Suite Provisioning Configuration Guide

## Introduction

SecureAuth IdP can be configured for directory - appliance - G Suite (formerly Google Apps) user profile provisioning, including password synchronization and other IdM tools. There are various ways to enable the features in SecureAuth IdP, and G Suite APIs must be configured appropriately.

Use this guide to enable user profile provisioning via G Suite APIs.

## Prerequisites

1. Procure G Suite and access to the Developers Console and Admin Console
2. Create a new SecureAuth IdP realm for provisioning G Suite
3. Obtain a directory **Service Account** with **read and write** access for SecureAuth IdP
4. Set up an Active Directory field to which SecureAuth IdP can map a **Profile Property**

For other data stores, the field mapping must be configured through the directory

## G Suite API Configuration Steps

### Create Project

The screenshot shows the Google Developers Console interface. In the top navigation bar, the 'CREATE PROJECT' button is highlighted with a red box. Below it, the 'Projects' section is visible, showing a table with columns for Project name, Project ID, Organization, Requests, Errors, and Charges. A modal dialog titled 'New Project' is open, with the 'Project name' field highlighted by a red box and containing the text 'Google Provisioning'. The 'Organization' dropdown is also visible. At the bottom of the dialog, there are 'CANCEL' and 'CREATE' buttons, with the 'CREATE' button highlighted by a red box.

1. Log into the [Google Developers' Console](#) , and navigate to **IAM & Admin > Projects** from the three bars menu
2. Select **Create Project**

These steps can also be completed by opening the **Projects** dropdown menu at the top, and selecting **Create project**

3. Provide a **Project Name**, and select an **Organization** if the project is not already being created within one
4. Click **Create**

## Enable Admin SDK

The screenshot shows the 'Admin SDK' page in the Google Cloud console. At the top, there is a navigation bar with a back arrow and the text 'Admin SDK'. To the right of this text is a red-bordered button with a play icon and the word 'ENABLE' in blue capital letters. Below the navigation bar, the page content includes a section titled 'About this API' with a 'Documentation' link and a 'Try this API in APIs Explorer' link. The main text describes the Admin SDK's purpose. There are two diagrams: one for 'Using credentials with this API' showing a flow from 'Your app' to 'User consent' to 'User data', and another for 'Server-to-server interaction' showing a flow from 'Your service' to 'Authorization' to 'Google service'.

5. In the **Libraries** section, search for **Admin SDK**, and select the option

6. On the **Admin SDK** page, click **Enable**

## Create Service Account

The screenshot shows the 'API Manager' 'Credentials' section in the Google Cloud console. On the left, there is a sidebar with a three-bar menu icon and the text 'Library'. The 'Credentials' option is highlighted with a red box. The main content area shows the 'Credentials' section with a 'Create credentials' button. Below this button, there are four options: 'API key', 'OAuth client ID', 'Service account key', and 'Help me choose'. The 'Service account key' option is highlighted with a red box. The text for 'Service account key' reads: 'Enables server-to-server, app-level authentication using robot accounts'.

7. On the **API Manager** page (accessible via the three bars menu), navigate to the **Credentials** section, and click **Create Credentials**

8. Select **Service Account Key**

## Create Service Account Key



### Create service account key

#### Service account

New service account

Service account name ?

service-account

Role ?

Owner

#### Service account ID

service-account @iam.gserviceaccount.c

#### Key type

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

JSON

Recommended

P12

For backward compatibility with code using the P12 format

Create

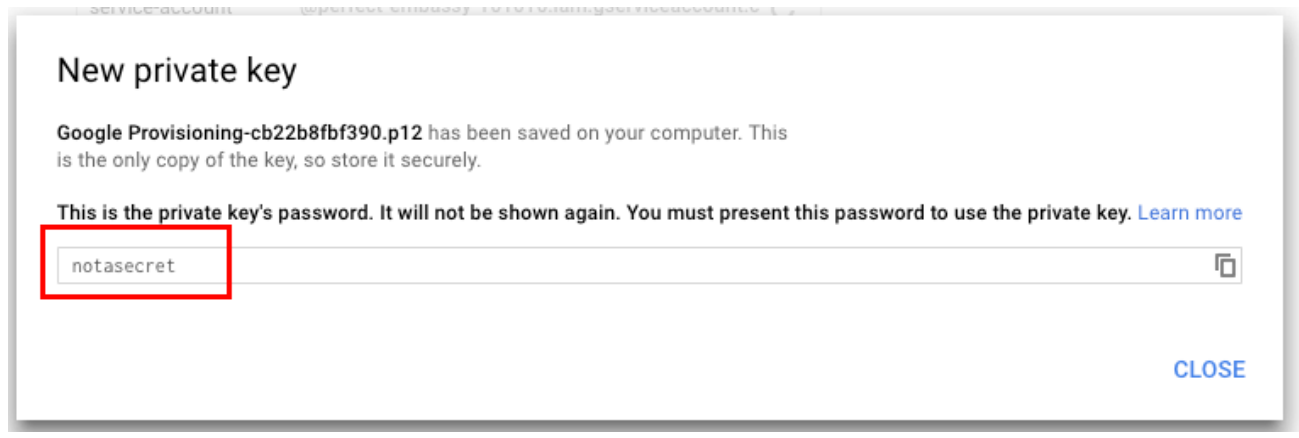
Cancel

9. Select **New service account** from the **Service Account** dropdown, and provide a **Service Account Name**

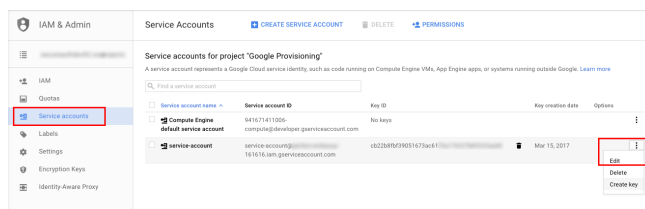
10. Select **Project > Owner** from the **Role** dropdown

11. Select **P12**, and click **Create**

12. Save the p12 file that downloads, which is uploaded to the SecureAuth appliance (see steps below), note the **Private Key Password**, and click **Close**



## Manage Service Account



13. On the **Credentials** page, click **Manage Service Accounts**

14. Click the three dots on the newly-created service account, and select **Edit**

## Edit Service Account

### Edit service account

Service account name [?](#)

**Enable G Suite Domain-wide Delegation**  
Grants a client access to all users' data on a G Suite domain without manual authorization on their part. [Learn more](#)

**i** To change settings for G Suite domain, product name for the OAuth consent screen must be configured. Assign the product name below or configure the OAuth consent screen.

Product name for the consent screen

[CANCEL](#) [SAVE](#) [CONFIGURE CONSENT SCREEN](#)

15. Check **Enable G Suite Domain-wide Delegation** and provide a **Product name for the consent screen**

16. Click **Save**

Click **Configure Consent Screen** to set additional (optional) preferences for the consent page; or access the configuration at **API Manager > Credentials > OAuth Consent Screen**

Service accounts for project "Google Provisioning"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more](#)

Find a service account

Service account name	Service account ID	Key ID	Key creation date	Options
Compute Engine default service account	941671411006-compute@developer.gserviceaccount.com	No keys		
service-account	service-account@iam.gserviceaccount.com	db2268f9951673ad179a17...	Mar 15, 2017	<a href="#">View Client ID</a>

Back on the **Service Accounts** page, a new **DwD** section appears for the service account

17. Click **View Client ID**

## Credentials

Credentials

[←](#) [Download JSON](#) [Delete](#)

Client ID for Service account client

**i** Service account clients are created when **domain-wide delegation** is enabled on a service account. [Manage service accounts](#)

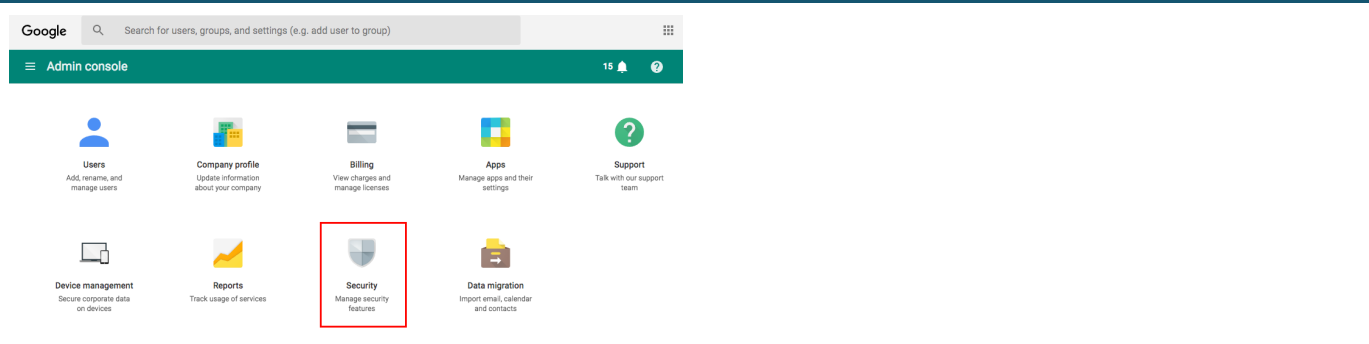
Client ID	1166612500697452
Service account	service-account@iam.gserviceaccount.com
Creation date	Mar 15, 2017, 10:27:40 AM

Name

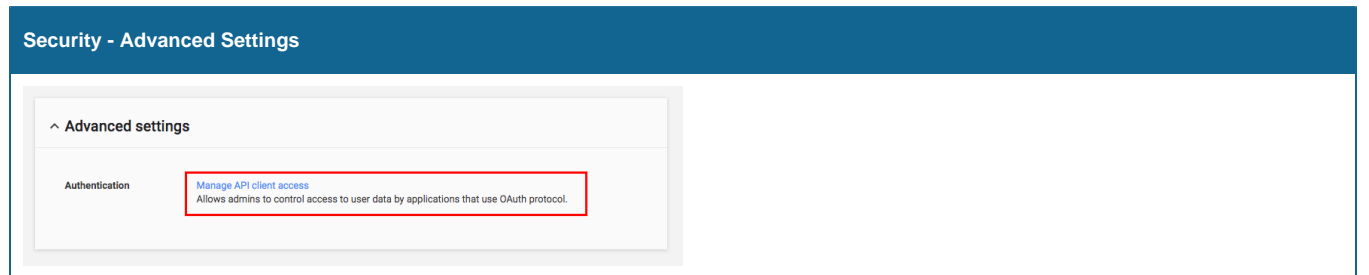
[Save](#) [Cancel](#)

18. Note the **Client ID**, which is used in the G Suite Administrative Configuration Steps (below), and the **Service Account** email address, which is used in the SecureAuth IdP Configuration Steps (below)

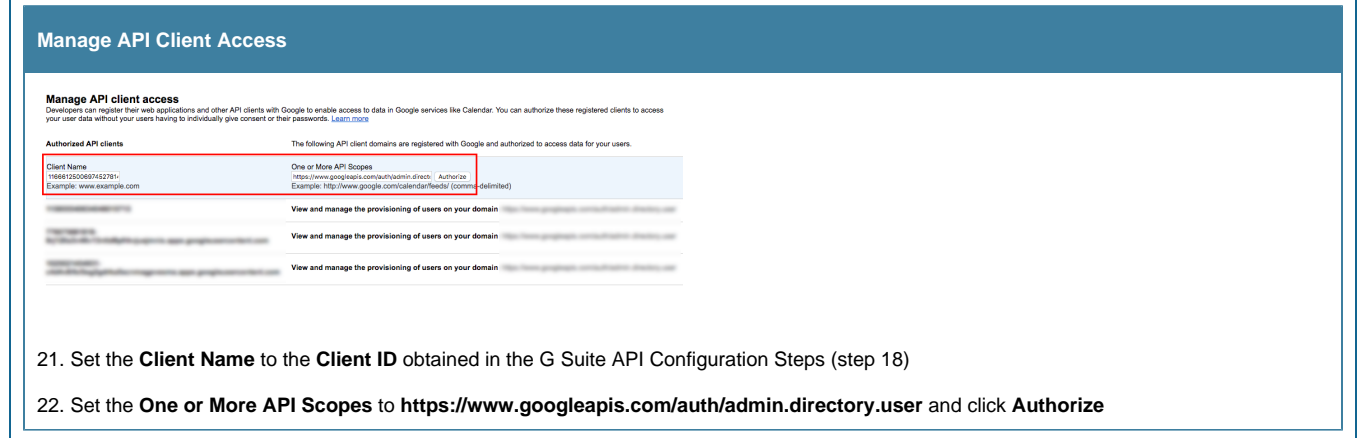
## G Suite Administration Configuration Steps



19. Log into the **G Suite Administrative Console** and select **Security**



20. Under **Advanced Settings**, select **Manage API Client Access**



## SecureAuth IdP Configuration Steps

The following SecureAuth IdP steps are required to enable G Suite provisioning functions from SecureAuth IdP / enterprise directory to G Suite  
See the **Related Documentation** links below to view the configuration steps for specific provisioning features

### Post Authentication

#### Google Apps Functions

1. Leave the **Google Apps Domain Name** field blank
2. Set the **Admin Email** to the G Suite Administrative email account
3. Set the **Service Email** to the **Service Account** email address obtained from the G Suite API Configuration Steps above (step 18)
4. Click **Choose File** and select the **p12 File** obtained in the G Suite API Configuration Steps above (step 12)
5. Set the **P12 Password** to the **Private Key Password** obtained in the G Suite API Configuration Steps above (step 12)
6. Select **Enabled** from the **Create User** dropdown if SecureAuth IdP is to automatically create the G Suite user account (if it does not already exist)
7. Select **Enabled** from the **Sync Password** dropdown if SecureAuth IdP is to conduct a one-way synchronization of the user's directory password to G Suite

To synchronize on specific dates versus every time the password changes, map a directory field to the **Ext. Sync Pwd Date** property in the **Data** tab

If no field is mapped, then the password synchronizes every time

G Suite requires passwords with a minimum of 8 characters

8. Select **Enabled** from the **Mail Forwarding** dropdown if another email address will receive messages; select **Disabled** to disable the feature; or select **Not Set** if SecureAuth IdP is to not be included in this feature
9. Select the **Profile Field** that contains the user's **Forwarding Email Address** (applicable only if **Enabled** is selected in step 8)

Click **Save** once the configuration has been completed and before leaving the **Post Authentication** page to avoid losing changes

## Related Documentation

- [Directory Password Synchronization with G Suite Configuration Guide](#)
- [Forgot Password Configuration Guide](#)
- [G Suite \(IdP-initiated\) Integration Guide](#)
- [G Suite \(SP-initiated\) Integration Guide](#)
- [iOS G Suite Provision Configuration Guide](#)
- [Reset Password Configuration Guide](#)