

Profile Manager and PTT Pro

Workcloud Communication



ZEBRA

Google Workspace Integration Guide

2024/04/24

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2023 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: zebra.com/linkoslegal.

COPYRIGHTS: zebra.com/copyright.

PATENTS: ip.zebra.com.

WARRANTY: zebra.com/warranty.

END USER LICENSE AGREEMENT: zebra.com/eula.

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

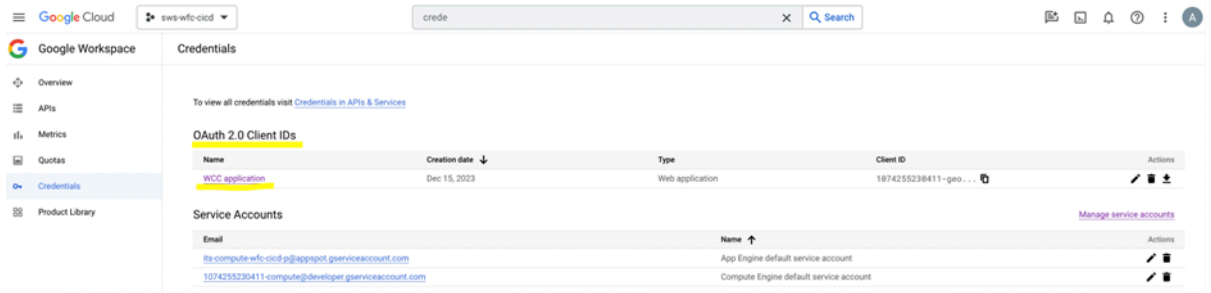
In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Configuring the GCP

The user must have created a Google space account.

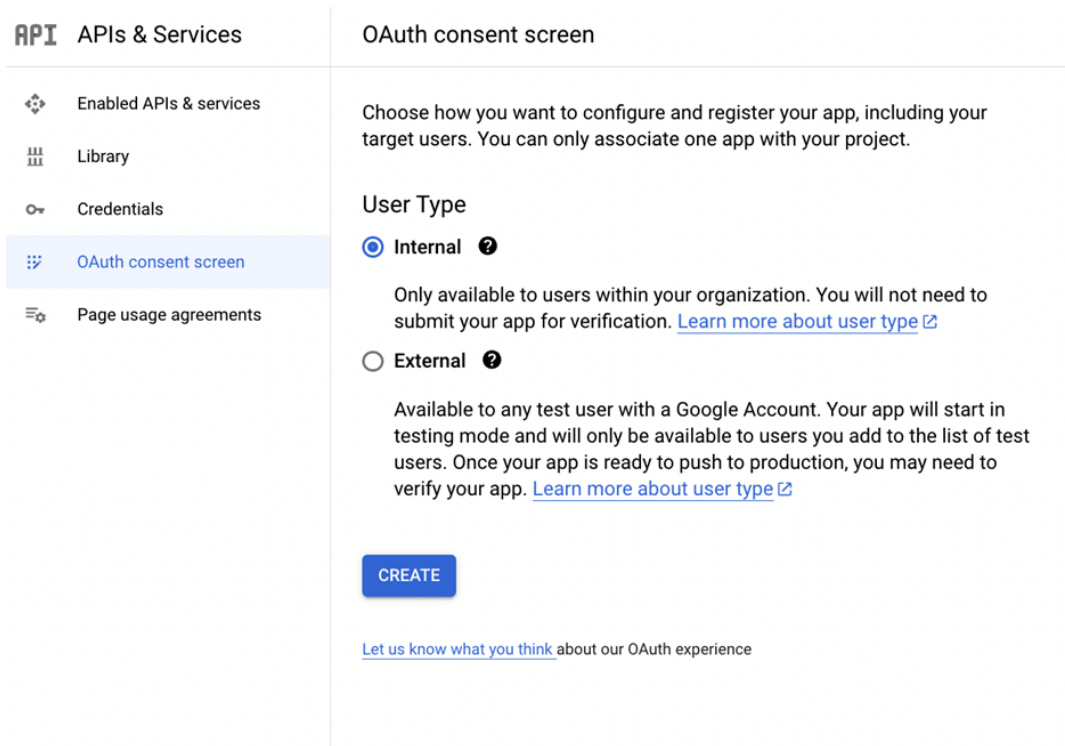
1. Sign in to the **Google Space** account.

2. Navigate to the **Credentials** page on GCP and select the **OAuth 2.0 Client IDs**.



If OAuth 2.0 Client IDs do not exist, follow the given instructions to create one. Skip this step if it already exists.

- Select the **OAuth Consent** screen from the **APIs & Services** navigation menu.
- For **User Type**, select **Internal** and click **Create**.



a) Define the **App name**, example: WCC PTTPro

The screenshot shows the 'Edit app registration' page in the Google Cloud console. The left sidebar contains navigation options: 'APIs & Services', 'Enabled APIs & services', 'Library', 'Credentials', 'OAuth consent screen' (highlighted), and 'Page usage agreements'. The main content area is titled 'Edit app registration' and has three steps: '1 OAuth consent screen', '2 Scopes', and '3 Summary'. The 'OAuth consent screen' step is active.

App information
 This shows in the consent screen, and helps end users know who you are and contact you

App name *
 WCC PTTPro
The name of the app asking for consent

User support email *
 [Redacted]
For users to contact you with questions about their consent. [Learn more](#)

App logo
 This is your logo. It helps people recognize your app and is displayed on the OAuth consent screen.
 After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of "Testing". [Learn more](#)

Logo file to upload [BROWSE](#)
Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

App domain
 To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page
Provide users a link to your home page

Application privacy policy link
Provide users a link to your public privacy policy

Application terms of service link
Provide users a link to your public terms of service

Authorized domains ⓘ
 When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

[+ ADD DOMAIN](#)

Developer contact information

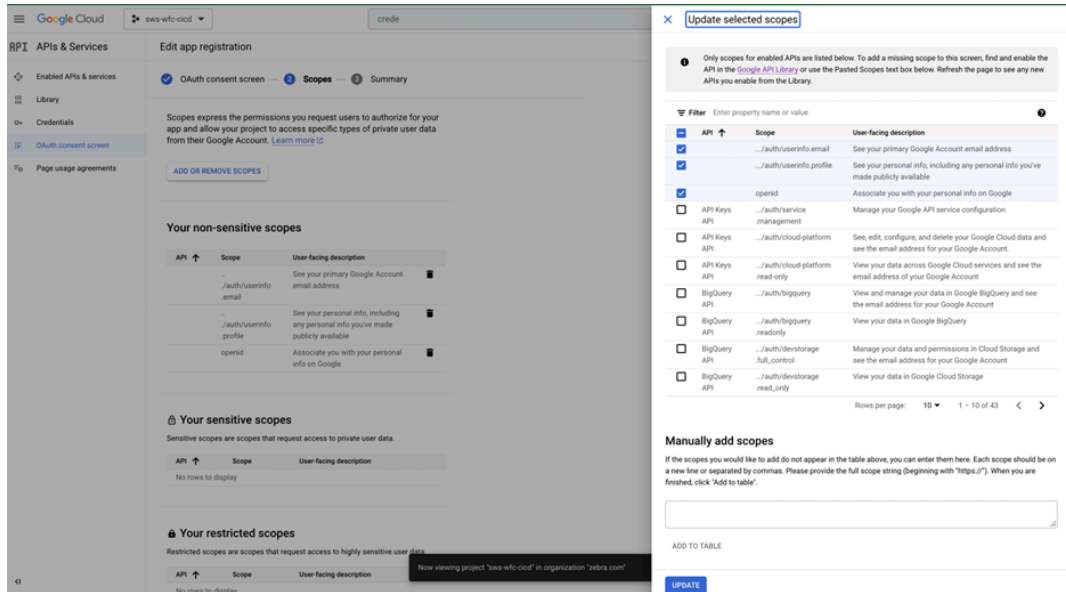
Email addresses *
 [Redacted]
These email addresses are for Google to notify you about any changes to your project.

[SAVE AND CONTINUE](#) [CANCEL](#)

3. Select an administrator account from the **User support email** drop down,
4. For **Developer contact information**, enter the administrator email address or any service account in the **Email address** field.
5. All other settings are optional and can be altered if required.
6. Select **Save and Continue**.

7. Select **Add or Remove Scopes** and enable the following scopes from the given:

- API
- .../auth/userinfo.email
- .../auth/userinfo.profile
- openid

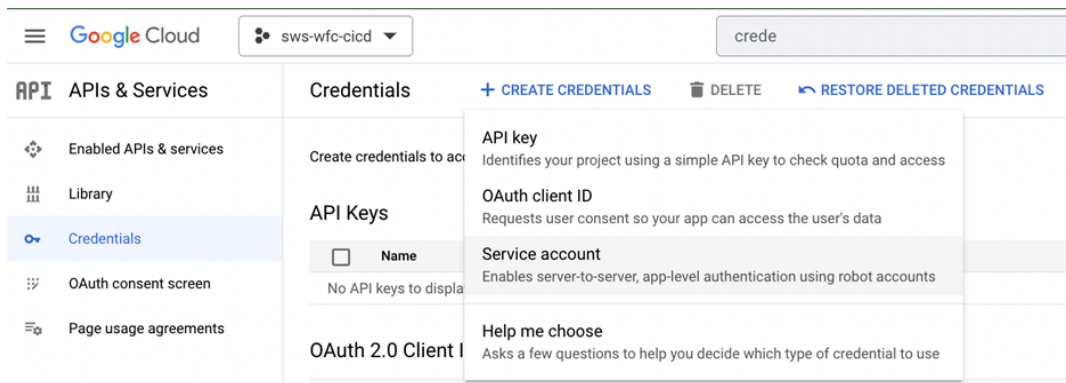


8. Click **Save and Continue**.

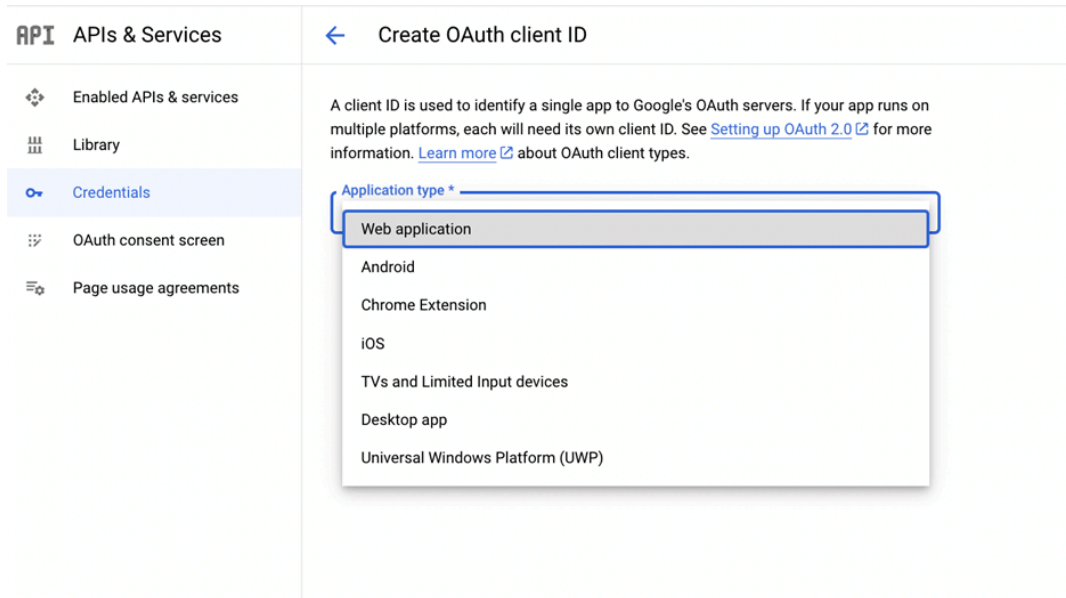
9. Review all the configurations and select **Back to Dashboard**.

10. Select the **Credentials** option from the **APIs & Services** navigation menu and click **Create Credentials**.

11. Select **OAuth client ID**.



12. For **Application type**, select **Web Application** from drop down.



13. Enter the application name, for example; the WCC application. Add <https://localhost> under **Authorized Redirect URIs** and click **Create**.

Google Cloud sws-wfc-cicd crede

API APIs & Services

- Enabled APIs & services
- Library
- Credentials**
- OAuth consent screen
- Page usage agreements

Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type *
Web application

Name *
Zebra WCC

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins

For use with requests from a browser

+ ADD URI

Authorized redirect URIs

For use with requests from a web server

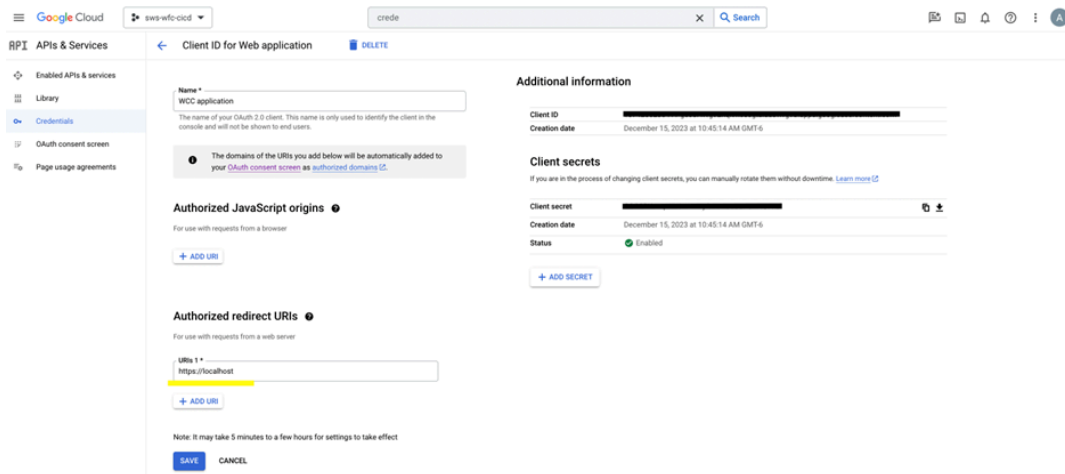
URIs 1 *
<https://localhost>

+ ADD URI

Note: It may take 5 minutes to a few hours for settings to take effect

CREATE CANCEL

14. Add the following under **Authorized redirect URIs**, <https://localhost>



15. Enable the following scopes:

- API
- .../auth/userinfo.email
- .../auth/userinfo.profile
- openid

16. Note down the **ClientID** and **Client Secret** for future use.

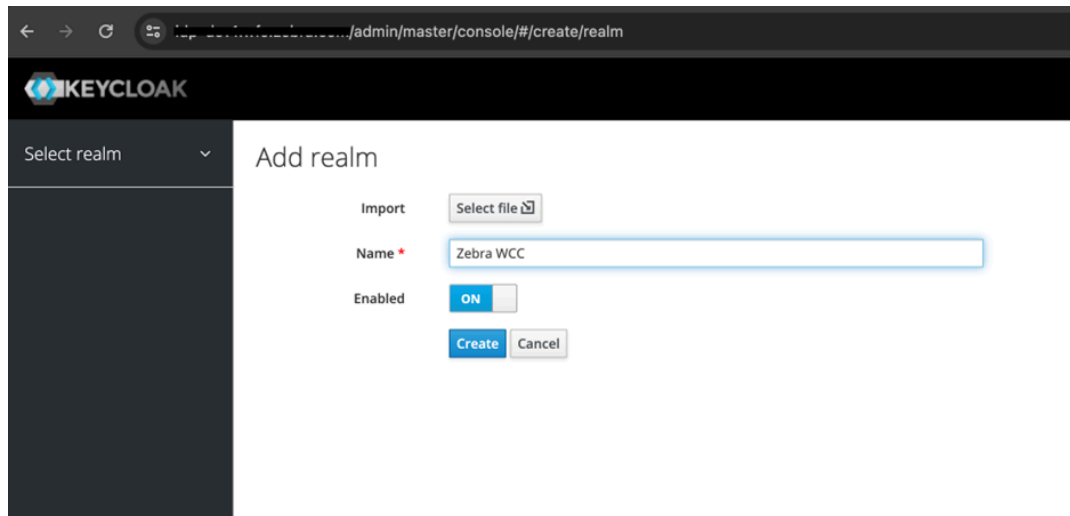
The screenshot shows the Google Cloud console interface for configuring a Client ID for Web application. The left sidebar contains the navigation menu with 'APIs & Services' selected, and 'Credentials' highlighted. The main content area is titled 'Client ID for Web application' and includes a 'DELETE' button. The configuration fields are as follows:

- Name ***: WCC application
- Authorized JavaScript origins**: For use with requests from a browser. Includes a '+ ADD URI' button.
- Authorized redirect URIs**: For use with requests from a web server. Includes two input fields: 'URIs 1 *' with 'https://localhost' and 'URIs 2 *' with 'https://[redacted]/realms/GoogleWorkspace-Test1/broker/gooq'. A '+ ADD URI' button is located below the second field.

A note at the bottom states: 'Note: It may take 5 minutes to a few hours for settings to take effect'. At the very bottom, there are 'SAVE' and 'CANCEL' buttons.

Configuring Google Space via ACS and IDP for Identifying Brokering

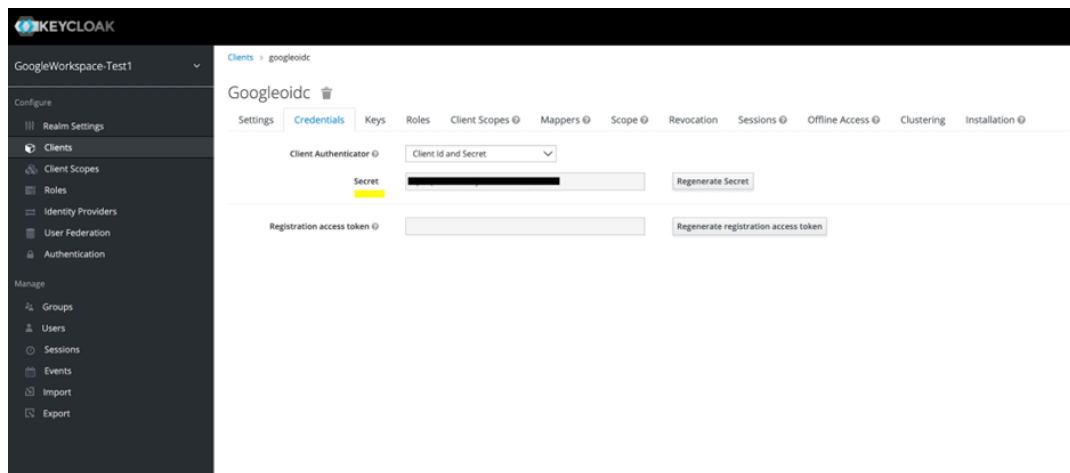
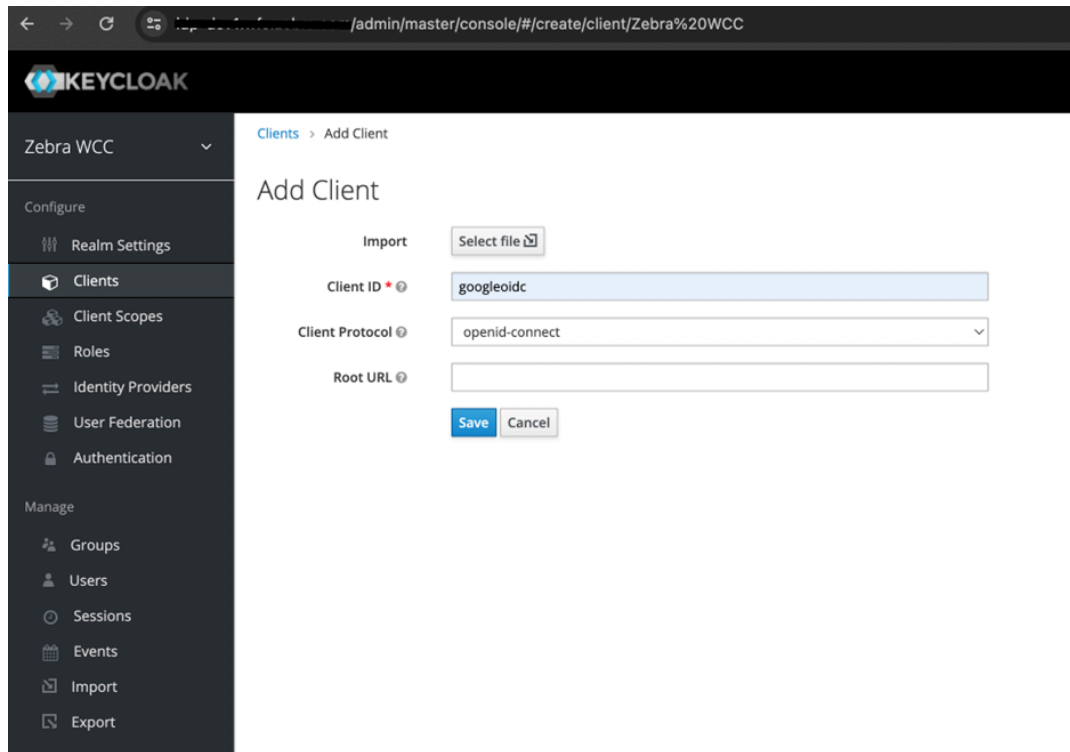
1. Add a New Realm.



The screenshot shows the Keycloak administration console interface for adding a new realm. The browser address bar indicates the URL is `.../admin/master/console/#/create/realm`. The page title is "Add realm". On the left, there is a sidebar with "Select realm" and a dropdown arrow. The main content area contains the following fields and controls:

- Import:** A button labeled "Select file" with a file icon.
- Name:** A text input field containing "Zebra WCC".
- Enabled:** A toggle switch currently set to "ON".
- Buttons:** "Create" and "Cancel" buttons at the bottom.











2. Create a client.



3. Add the mappers for upn **Name** and **Token Claim Name** , both must point to **username** in the Property field.

[Clients](#) > [googleoidc](#) > [Mappers](#) > upn

Upn 

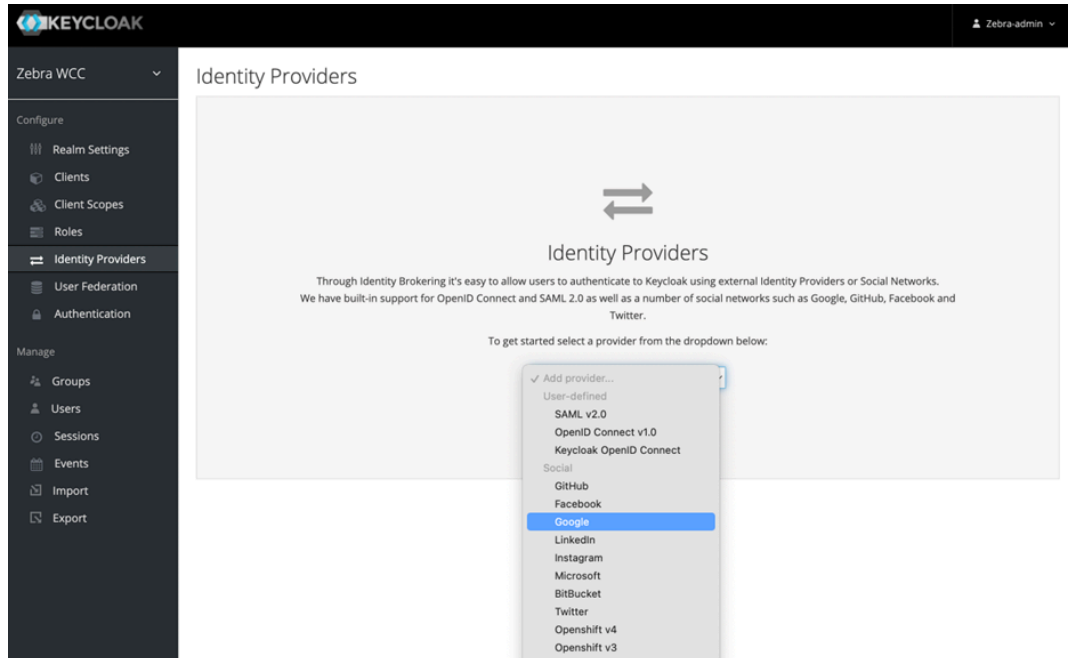
Protocol 	<input type="text" value="openid-connect"/>
ID	<input type="text" value="2869c40b-b16a-4ef5-8b5c-cf244707d93d"/>
Name 	<input type="text" value="upn"/>
Mapper Type 	<input type="text" value="User Property"/>
Property 	<input type="text" value="username"/>
Token Claim Name 	<input type="text" value="upn"/>
Claim JSON Type 	<input type="text" value="String"/> 
Add to ID token 	<input checked="" type="checkbox"/>
Add to access token 	<input checked="" type="checkbox"/>
Add to userinfo 	<input checked="" type="checkbox"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Clients > googleoidc > Mappers > unique_name

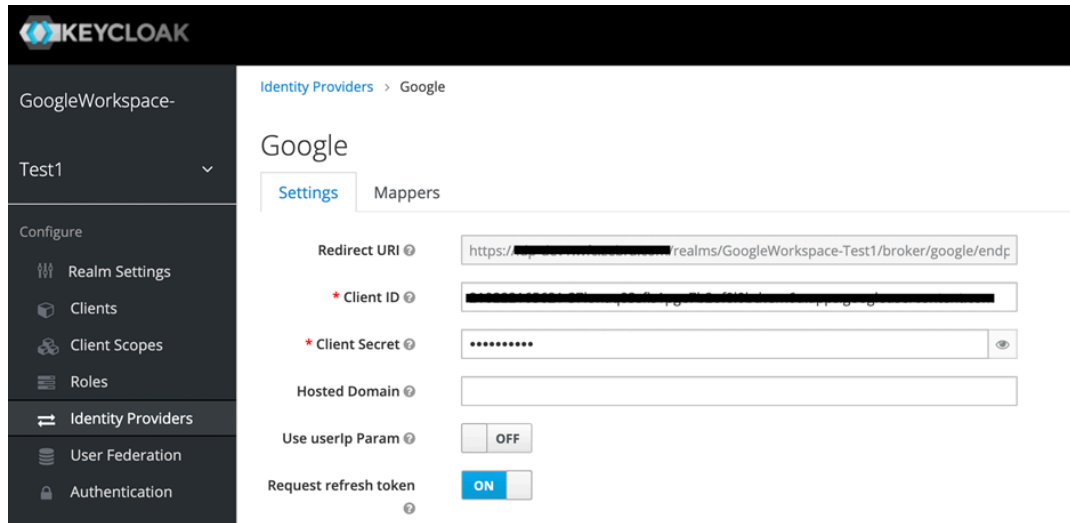
Unique_name

Protocol ?	<input type="text" value="openid-connect"/>
ID	<input type="text" value="509c48cf-58f2-4639-bc24-ad254cd19b4e"/>
Name ?	<input type="text" value="unique_name"/>
Mapper Type ?	<input type="text" value="User Property"/>
Property ?	<input type="text" value="username"/>
Token Claim Name ?	<input type="text" value="unique_name"/>
Claim JSON Type ?	<input type="text" value="String"/>
Add to ID token ?	<input checked="" type="checkbox"/>
Add to access token ?	<input checked="" type="checkbox"/>
Add to userinfo ?	<input checked="" type="checkbox"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

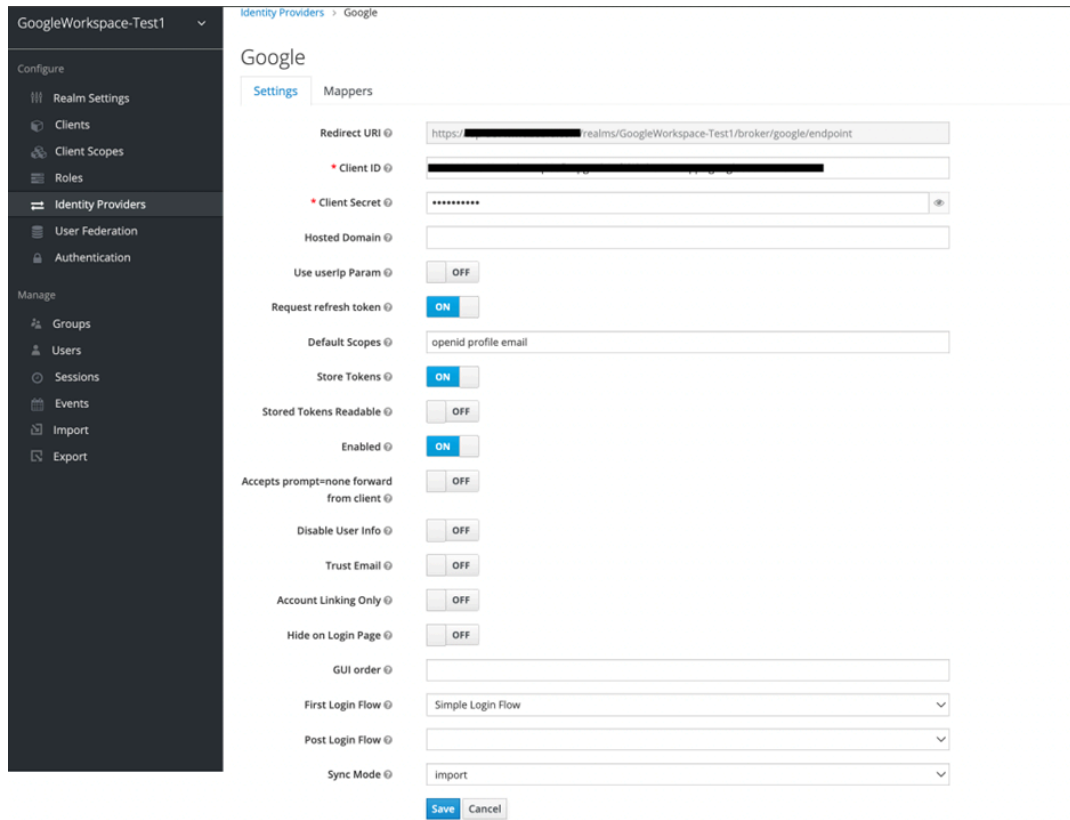
4. From the **Identity Provider** menu, select the **Google** from drop down.



5. Add configuration copied from step1, the **Google Client ID** and **Client Secret** in the **Identity Providers** menu.



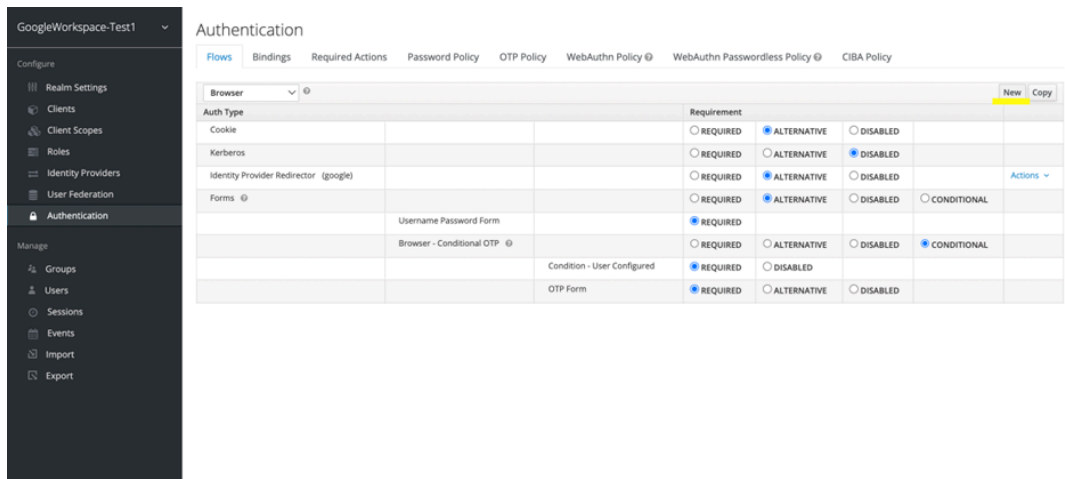
6. For Default scopes, use **openid profile email**.
7. Enable the toggle for the **Request refresh token**, and **Store Tokens** option.



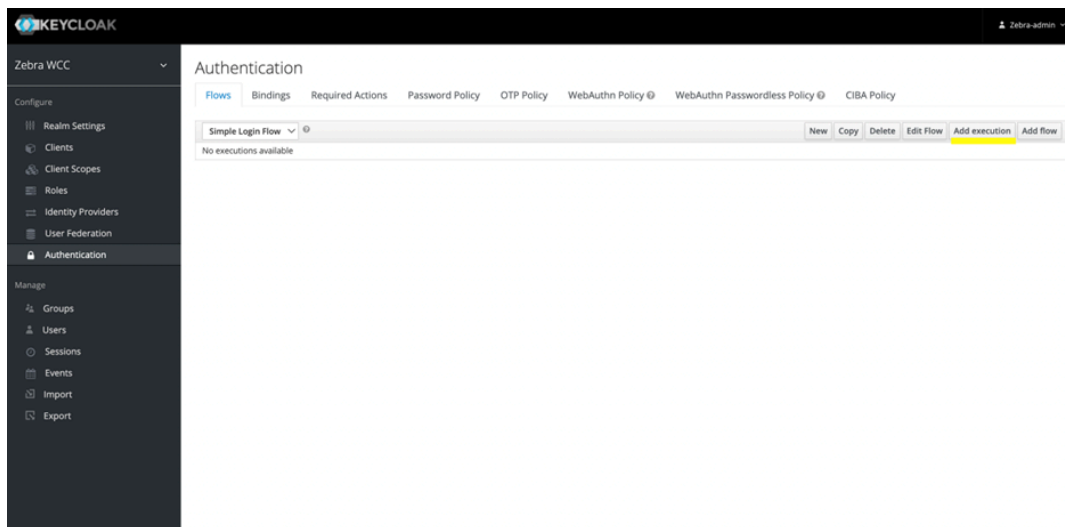
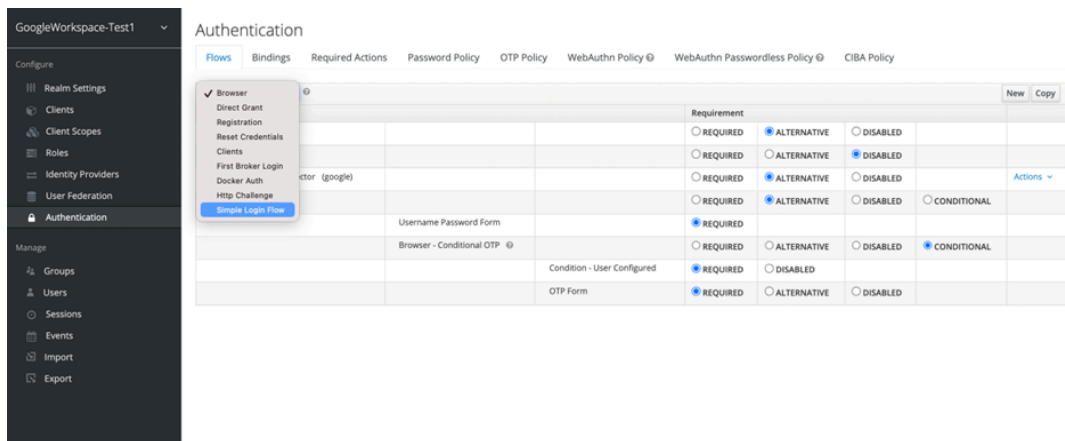
8. For the **First Login Flow** field, select the **Simple Login Flow** from drop down.

Creating a Simple Login Flow

1. On the Admin page, open **Authentication** > **Flows** tab and click **New** button.

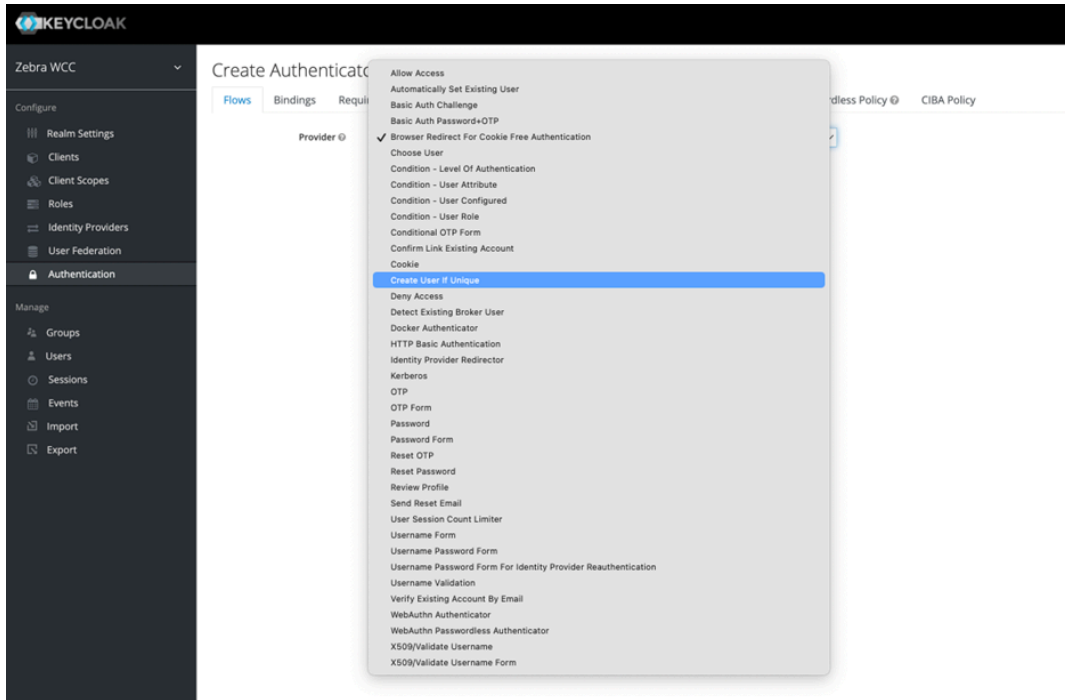


2. Enter the **Simple Login Flow** name in the **Alias** field.



3. Click **Save**.

4. Select the **Simple Login Flow** and Add execution.
5. Select **Create User If Unique** from the list and save.



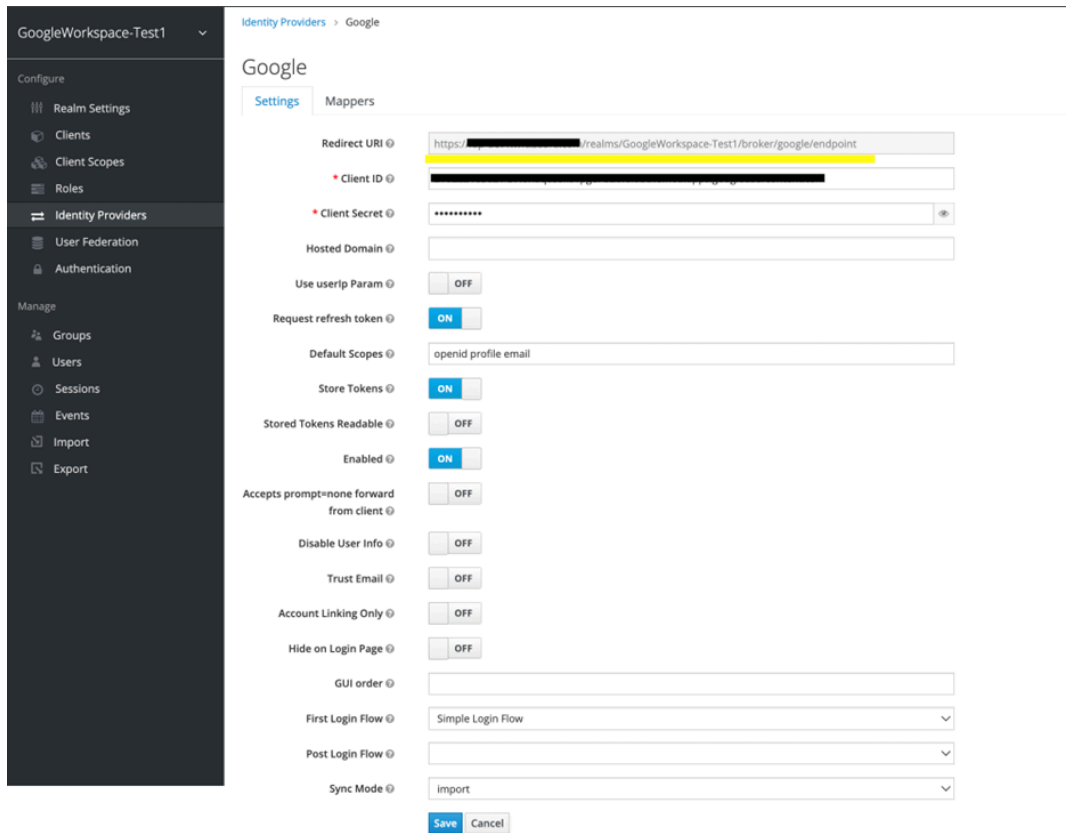
6. Configure the execution. Select **ALTERNATIVE Requirement**.

7. Now, you can use the flow in the Identity Provider configuration.

The screenshot displays the configuration interface for the Google identity provider. On the left is a dark sidebar with navigation options: 'Configure' (Realm Settings, Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication) and 'Manage' (Groups, Users, Sessions, Events, Import, Export). The main content area is titled 'Identity Providers > Google' and 'Google'. It features a 'Settings' tab and a 'Mappers' section. The configuration fields include: Redirect URI (https://.../realms/GoogleWorkspace-Test1/broker/google/endpoint), Client ID (redacted), Client Secret (masked with dots), Hosted Domain (empty), Use userip Param (OFF), Request refresh token (ON), Default Scopes (openid profile email), Store Tokens (ON), Stored Tokens Readable (OFF), Enabled (ON), Accepts prompt=none forward from client (OFF), Disable User Info (OFF), Trust Email (OFF), Account Linking Only (OFF), Hide on Login Page (OFF), GUI order (empty), First Login Flow (Simple Login Flow, highlighted in yellow), Post Login Flow (empty), and Sync Mode (import). 'Save' and 'Cancel' buttons are at the bottom.

8. Copy the **Redirect URI** from IDP configuration.

This needs to be added to the GCP OAuth Client ID application under the Authorized Redirect URIs section.



Google Cloud sws-wfc-cicd crede

API APIs & Services Client ID for Web application DELETE

- Enabled APIs & services
- Library
- Credentials**
- OAuth consent screen
- Page usage agreements

Name *
WCC application
The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins
For use with requests from a browser
+ ADD URI

Authorized redirect URIs
For use with requests from a web server

URIs 1 *
https://localhost

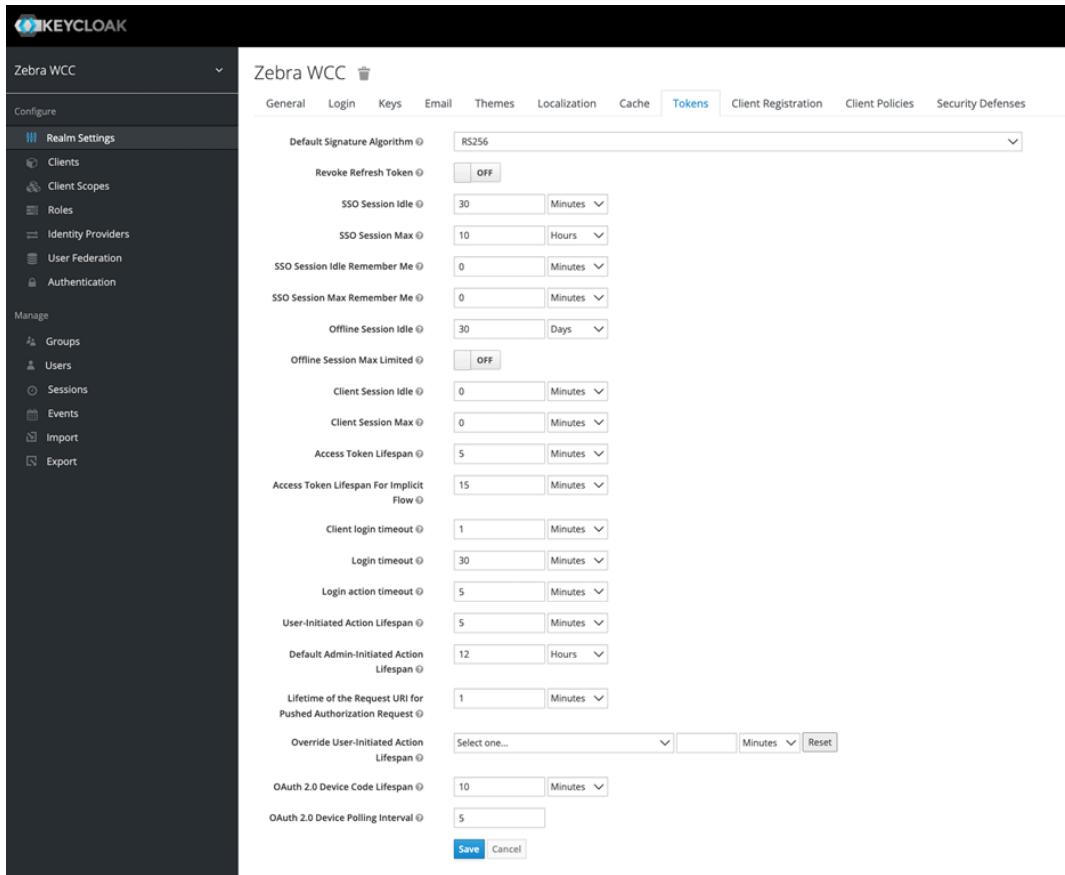
URIs 2 *
https://[redacted]/realms/GoogleWorkspace-Test1/broker/gooq

+ ADD URI

Note: It may take 5 minutes to a few hours for settings to take effect

SAVE CANCEL

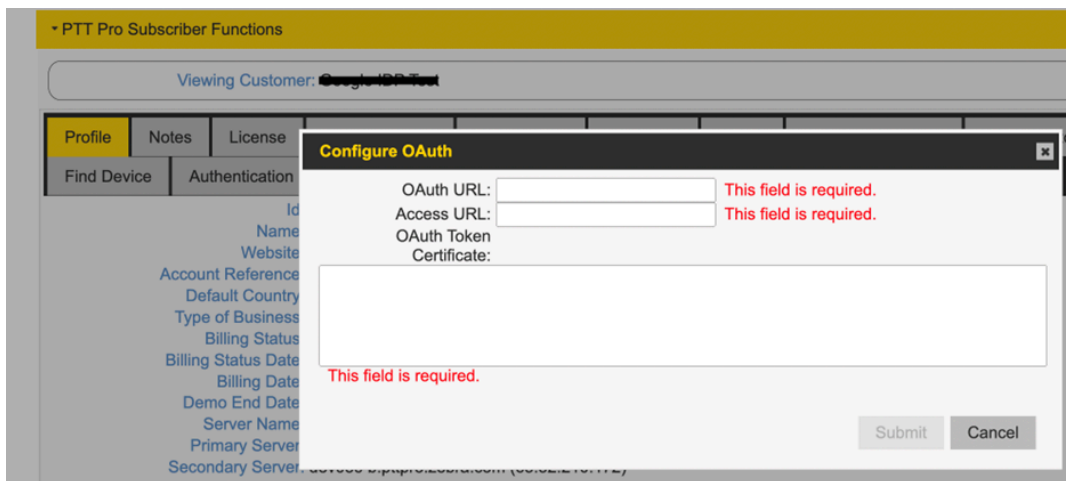
9. Select **Realm Settings > Tokens**, update the **SSO Session Idle**, **SSO Session Max**, **Access Token Lifespan**, and **Access Token Lifespan for Implicit Flow** accordingly based on the customer's requirements.



10. Click **Save** to save the configuration.

Configuring OAuth via PTT Pro Customer Settings

1. Add the **OAuth URL**, **Access URL**, and **OAuth Token Certificate** on the PTT Pro customer settings.

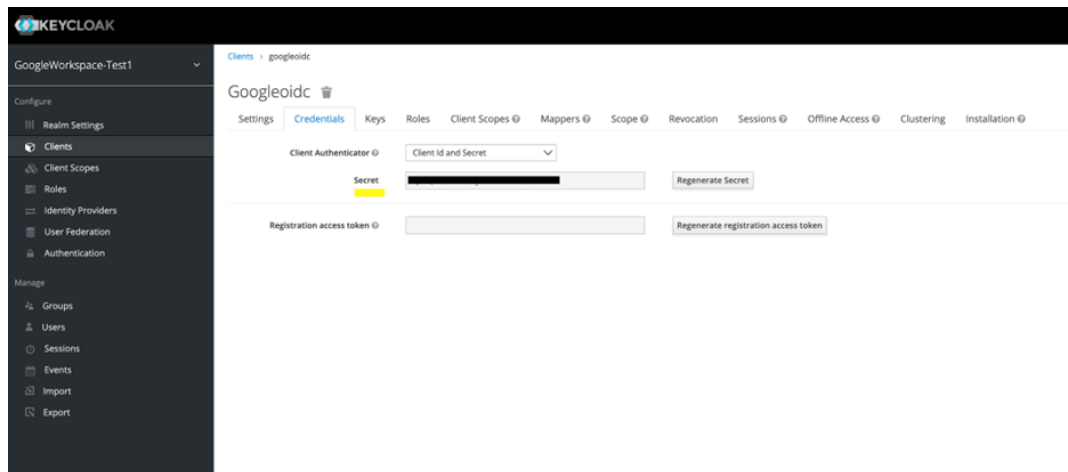


2. The **OAuth URL**, and the **Access URL** can be obtained from IDP's well-known URL, for example:
`https://<DNS>/realms/<Realm-name>/well-known/openid-configuration`
3. Update the **DNS** and **Realm-name** accordingly.
4. Certificate can be obtained from **Realm settings > Keys > RS256 > Certificate**.
5. To enable **Google IDP**, log in a standalone **PTT Pro Client** (without PFM), the following JSON configuration needs to be pushed to the PTT Pro Client before initiating the login.

```
{
  "oAuthClientID": "googleoidc",
  "oAuthClientSecret": "<<CHANGETHIS>>",
  "oAuthBasicHeader": false,
  "customUserAgentString": "Zebra Android/PTTPro"
}
```

Configuring Using the PFM Tenant Configuration

1. Copy the **OAuth URL** and the **Access URL** from the IDP well-known URL, for example:
`https://<DNS>/realms/<Realm-name>/well-known/openid-configuration`
2. Update the **DNS** and the **Realm-name** accordingly.
3. Copy the **Client Secret** from IDP.



4. For **Client ID** enter `googleoidc`.
5. For **Token username** enter `unique_name`.

6. Configure the **OAuth Details** of the PFM tenant.


OAuth Details:

Host Url


Authentication Path

Token Path

Client ID

Client Secret Key
 

Token Username

Client Authentication *
 

7. To enable **Google IDP**, log in the Profile Client application, we must push the custom user agent string to the device and start the intent. Sample Configuration Key.json file:

```
{  
  "customUserAgentString": "Zebra/PFC"  
}
```

