Setting Up Google as Your Mail Service Provider

WHMCS can send emails without any additional configuration using **PHP mail()**. While this works in most cases, other mail providers may give you a better experience and access to additional features.

In addition to the existing SMTP support, WHMCS 8.0 added support for Mailgun, SendGrid, SparkPost, and Google® OAuth with SMTP. WHMCS 8.6 and later also include Microsoft® services. For more information, see:

- Configuring WHMCS to use SMTP for WHMCS 8+
- Configuring WHMCS to use Other Mail Providers
- Setting Up Microsoft As Your Mail Service Provider
- Mail Provider Integrations

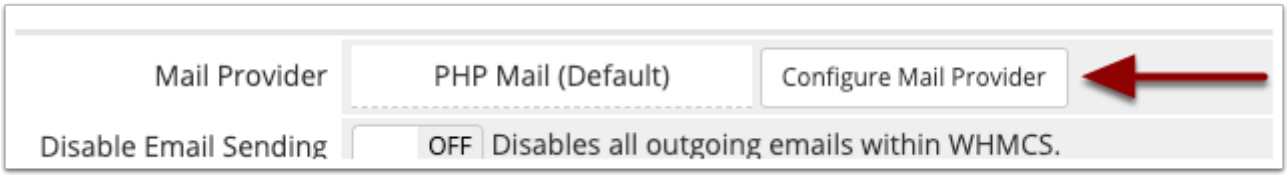
Set up Google in WHMCS

To configure **Google**, you will need to create an app in the Google Cloud console and configure the mail provider in WHMCS.

To configure the mail provider:

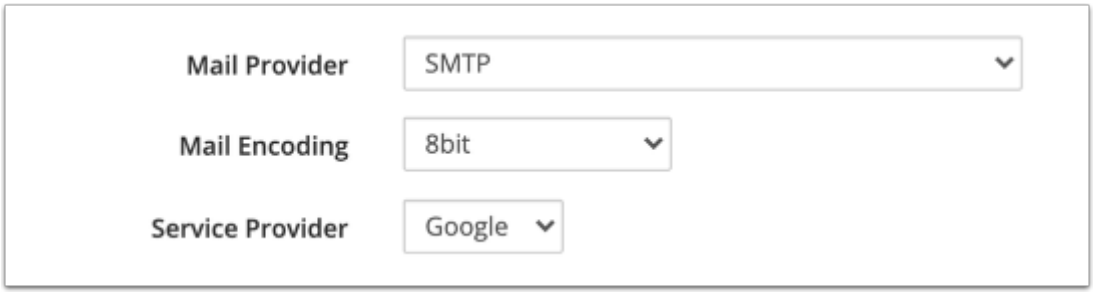
1. In the Admin Area, go to the **Mail** tab at **Configuration > System Settings > General Settings**.

2. Click **Configure Mail Provider**.



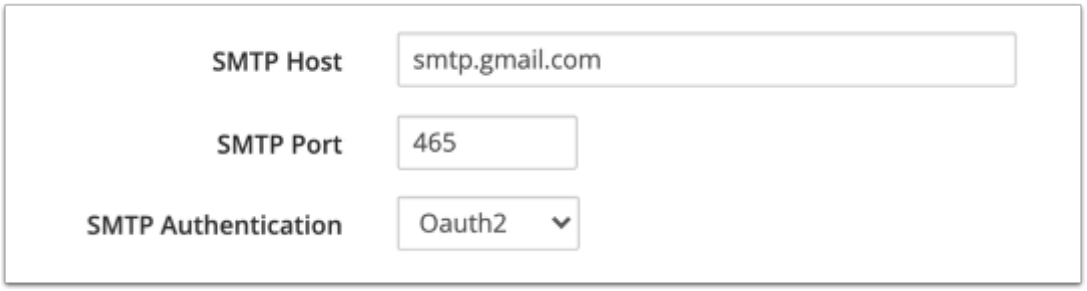
3. Select *SMTP* as your **Mail Provider** and choose a **Mail Encoding**.

4. Select *Google* as your **Service Provider**.



5. Enter `smtp.gmail.com` for the **SMTP Host** and `465` for the **SMTP Port**.

6. Select *Oauth2* for **SMTP Authentication**.

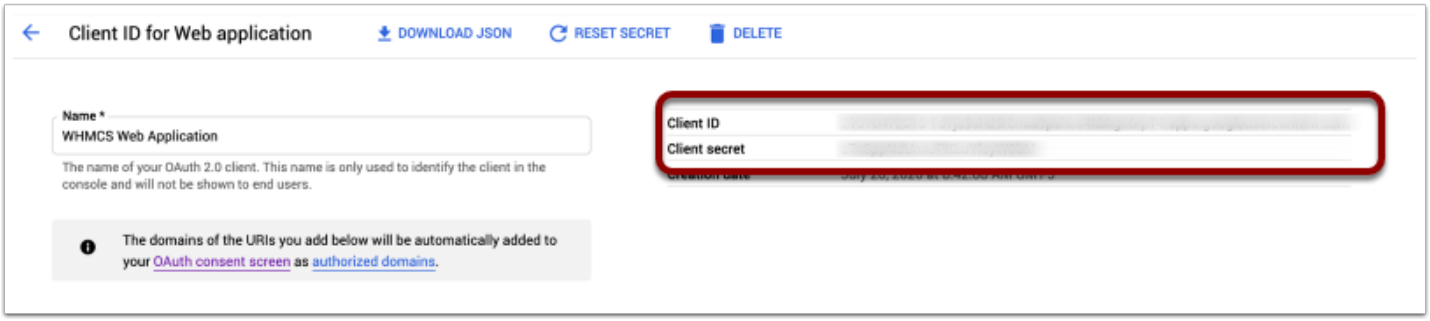


7. For **SMTP Username**, enter the Gmail™ address that you will be using in your application.

> For steps to generate the Client ID and Client Secret in Google, see the [Create your Google Application] section below.

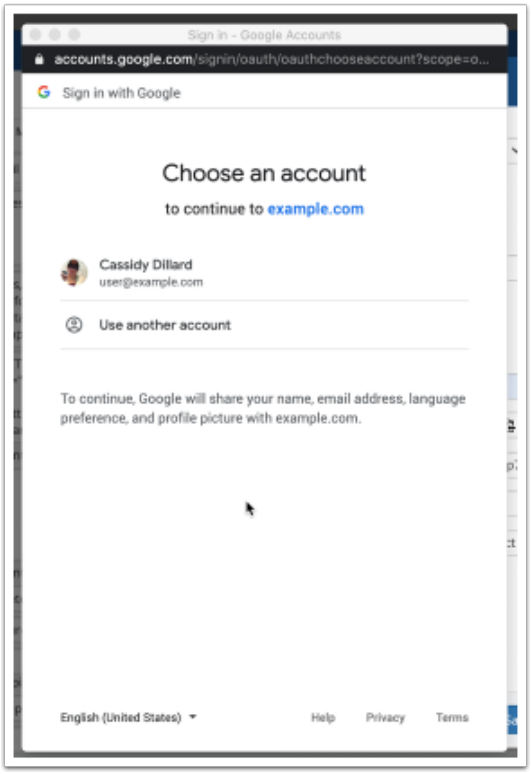
8. Copy-and-paste the **Client ID** and **Client Secret** from the Google Cloud Console into the appropriate boxes in the confirmation message.

9. You can also edit them by going to **Credentials** and clicking the edit icon for the appropriate **OAuth 2.0 Client IDs** row:



10. Next to **Connection Token**, click **Connect**. (**Connection Token** will be empty.)

11. Under **Choose an account**, select the account that you used to create the app.



12. Follow the prompts to approve access for your account.

> If you see a *This app isn't verified.* error, click **Advanced** and then click **Go to** at the bottom of the window.

The system will automatically enter a token in **Connection Token**.

13. To ensure that your configuration works, click **Test Configuration**.

14. Click **Save**. The system will test your configuration again when you save.

Create your Google application

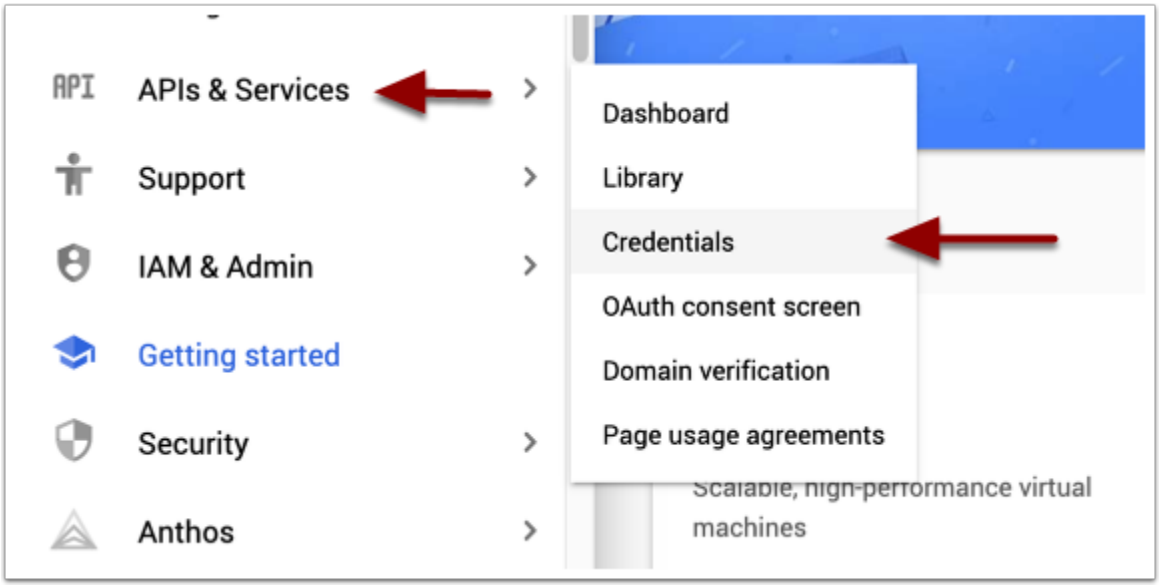
To use Google as your service provider, you will need to create an app and then create an associated client ID. This will let you connect to Google via WHMCS.

If you have already used Google Cloud's console, you may not need to perform some of these steps, or interfaces may not look like the screenshots below.

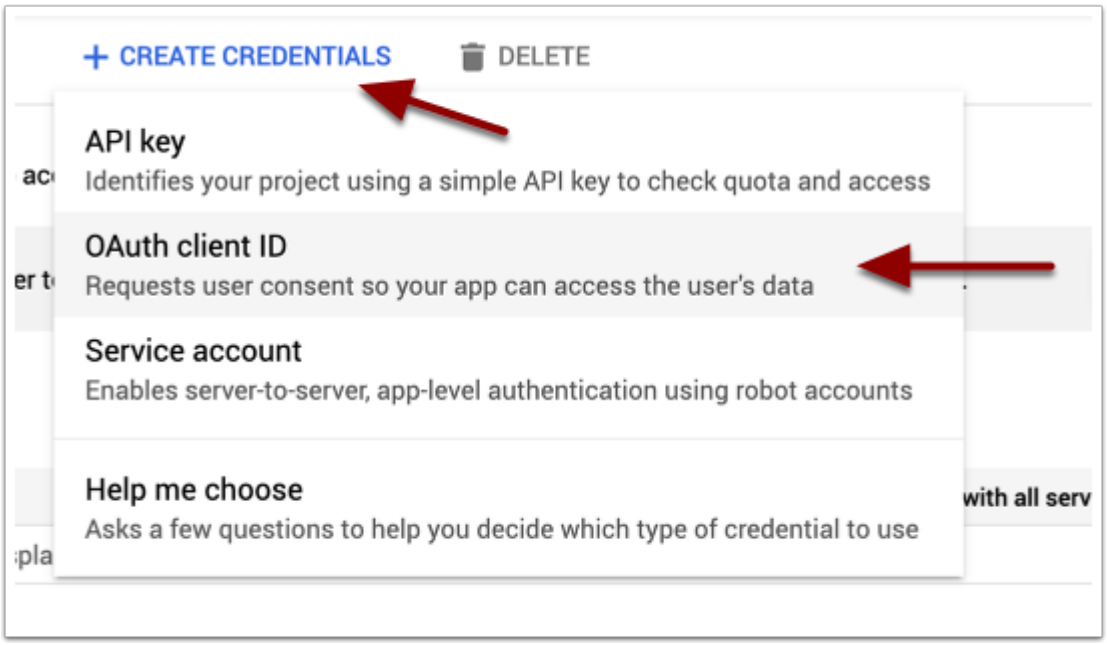
First, create the app:

1. Log in to the Google Cloud console. If you haven't before, select your country and agree to Google's *Terms of Service*.

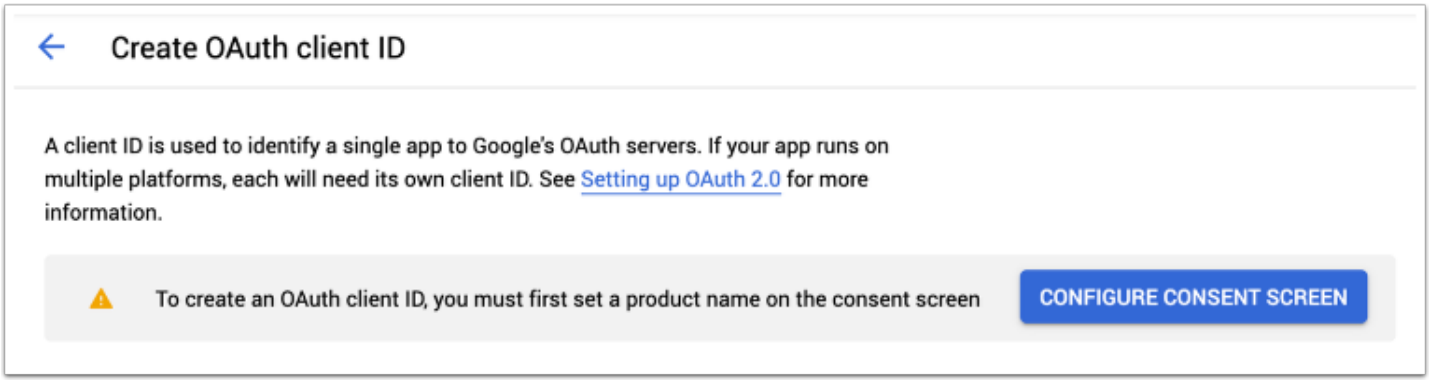
2. Go to **APIs and Services > Credentials**.



3. Click **Create Credentials** and select **OAuth client ID**.

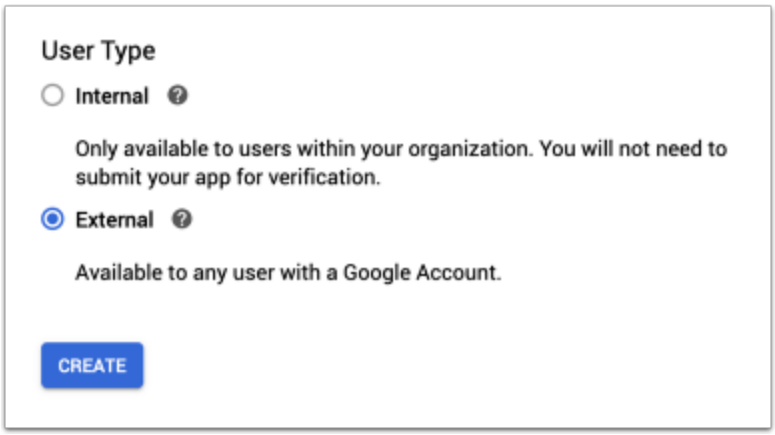


4. Click **Configure Consent Screen**.



5. Select *External* for **User Type**.

Selecting this allows anyone to use the generated client ID after a verification process. However, when creating a client ID that is only for WHMCS, you do not need verification.

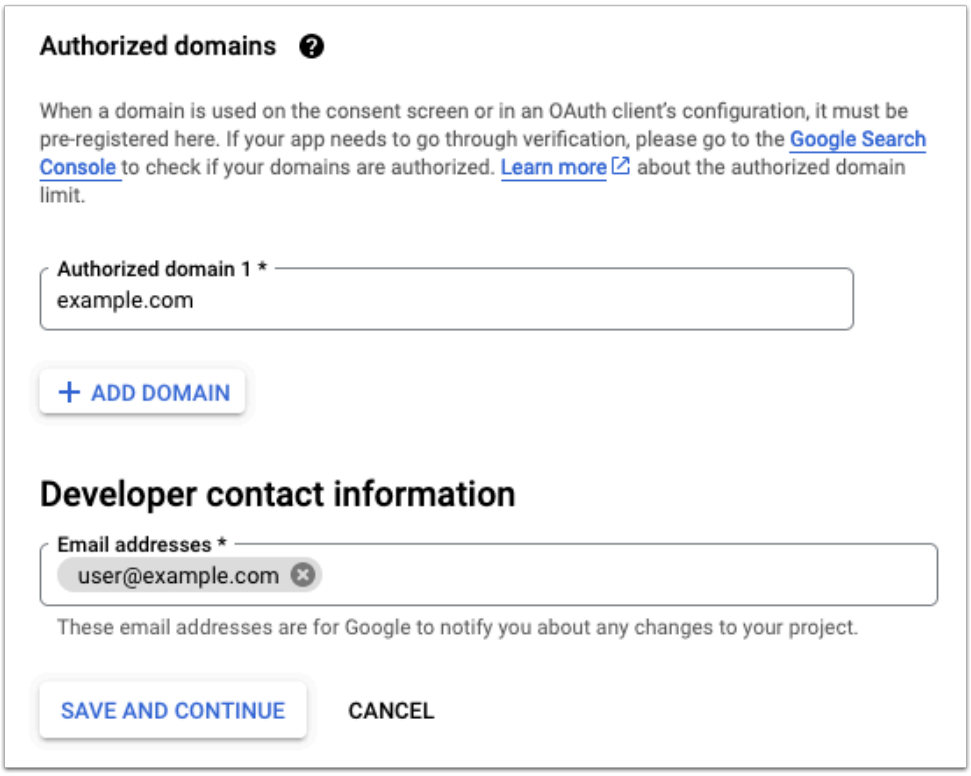


6. Click **Create**.

7. Enter a new **App name**.

8. Select a **User support email** address.

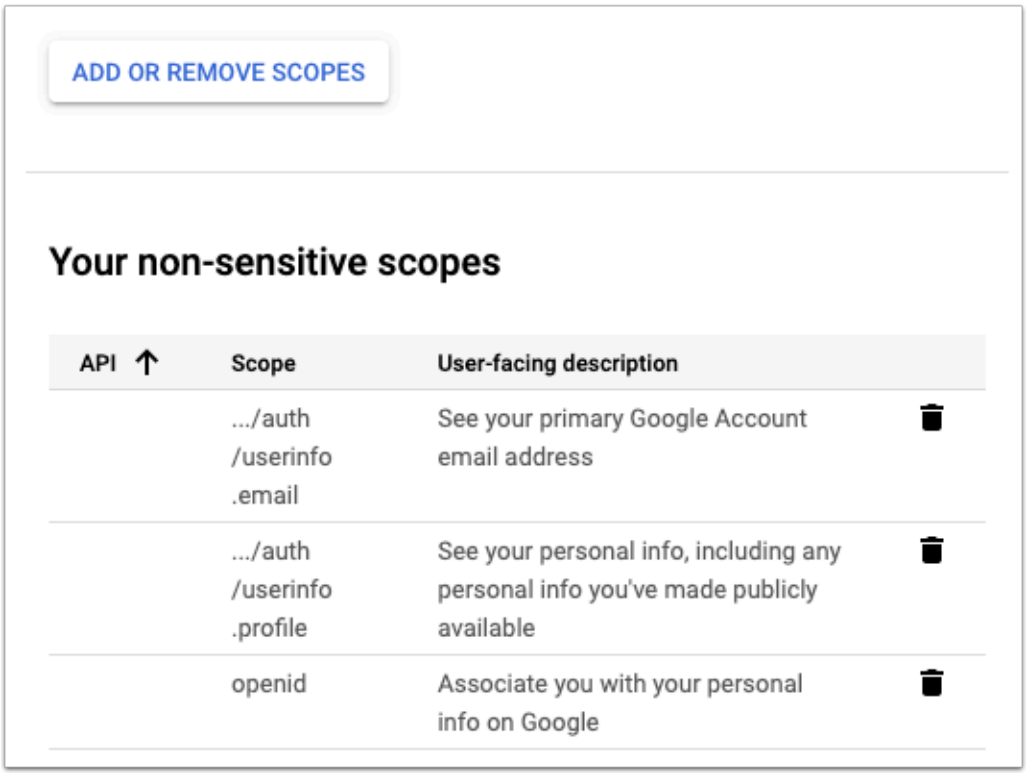
9. Click **Add Domain** and enter the domain for your WHMCS installation.



10. Click **Save and Continue**.

11. Click **Add or Remove Scopes** and add the following scopes:

- *userinfo.email*
- *userinfo.profile*
- *openid*



12. Click **Save and Continue**.

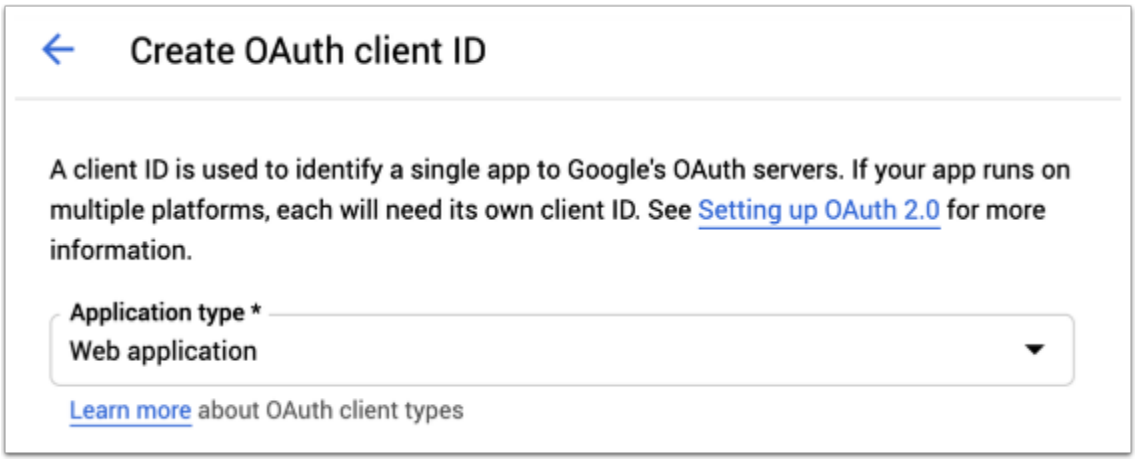
13. For **Test users**, click **Save and Continue** without making any changes.

After you create the app, you can create the client ID:

1. In the left sidebar, click **Credentials**.

2. Click **Create Credentials** and select **OAuth client ID** again.

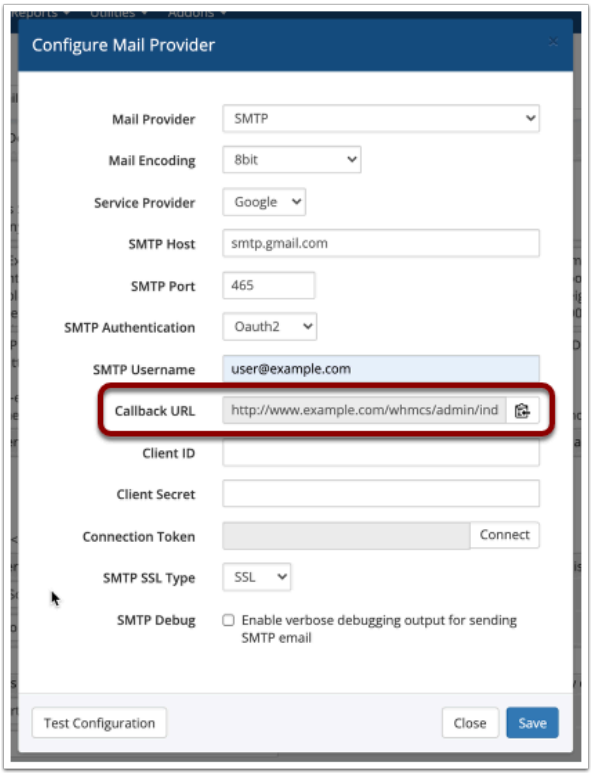
3. For the **Application Type**, select **Web Application**.

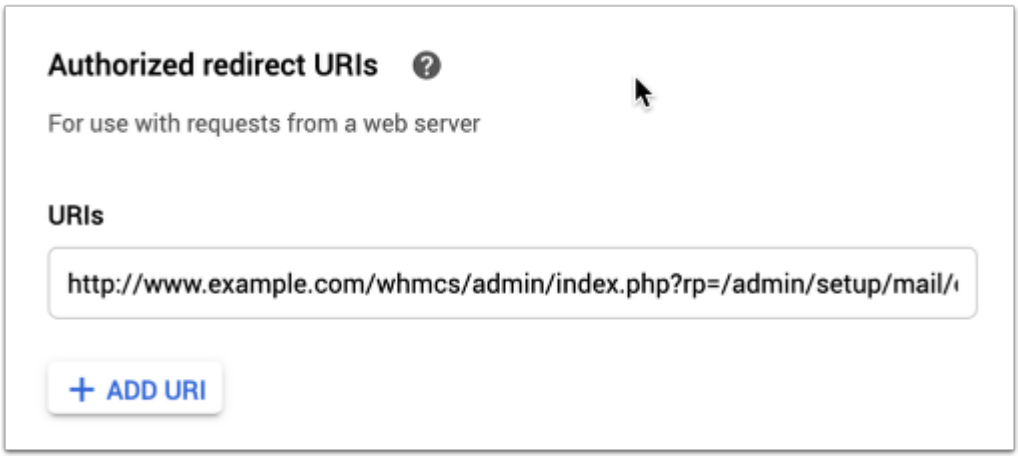


4. Enter a name for your application.

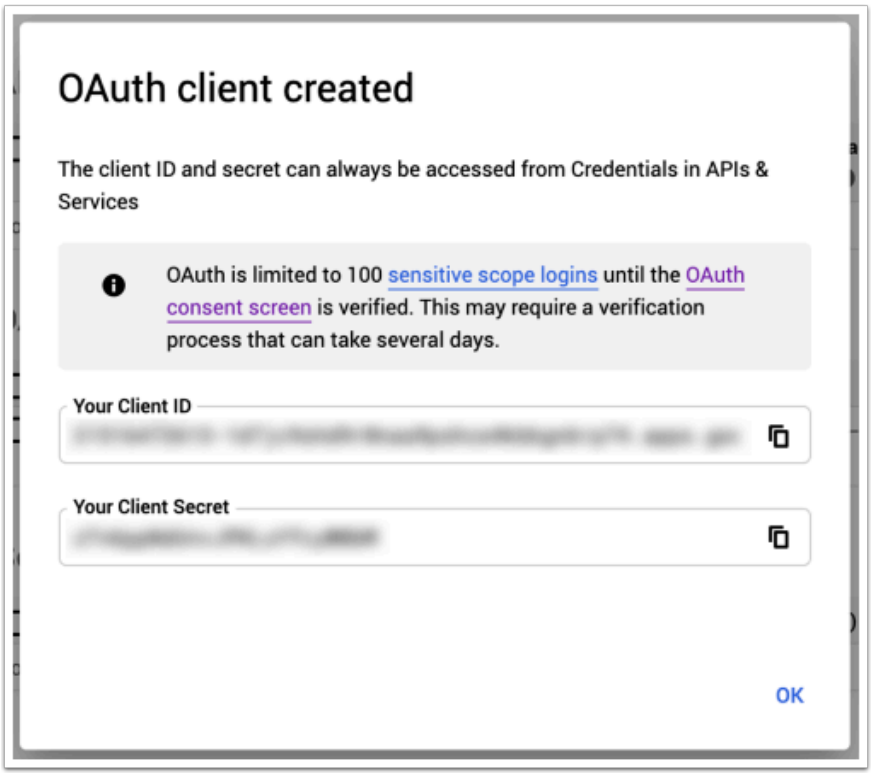
5. Under **Authorized redirect URIs**, click **Add URI**.

6. Enter the **Callback URL** that displays in WHMCS.





7. Click **Create**. A confirmation message will appear, with the **Client ID** and **Client Secret** to use in the steps above.



You can now continue configuring SMTP with OAuth 2.0 (see the section above).