OpenID | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/plugins/auth-systems/open-id.html> |
| Date de scraping | 2025-04-08T15:55:16.433201+00:00 |

## OpenID ​

OpenID is a layer on top of the OAuth 2.0 protocol that allows you to authenticate users and get their basic profile from the OAuth provider of your choice, including but not limited to Google, Slack, LINE, etc.  
How you configure the OpenID plugin in WeWeb will depend on the OAuth provider you are working with. In this article, we will demonstrate using Google OAuth.

## Provider setup ​

Before you can configure the WeWeb plugin, you'll need the following information from your OAuth provider:  
the domain WeWeb should send the request to via OpenID a client ID a client secret a scope (i.e. what information you want to get) a response type  
How you get this information will depend on your OAuth provider.  
To allow users to signup or login with their Google account, you would need to:  
go to the Google Cloud Console , and create a new project  
  
Once you've created a project, you'll be able to configure the provider's consent screen and get credentials so WeWeb can make secure requests to the provider via the OpenID plugin.

### Images

<https://docs.weweb.io/assets/open-id1.DvrD9HAJ.png>

## OAuth consent screen ​

The OAuth consent screen is the screen the user will see when they choose to connect with the provider.  
For example, the Google consent screen looks something like this:  
  
To configure the Google OAuth consent screen, you'll need to go to APIs & Services > OAuth consent screen :  
  
At that stage, the provider will ask for some information such as:  
Branding – For example, what is the name of your application and its logo.  
Contacts – For example, a developer and/or support email address.  
User type – Whether your app is reserved for people in your company ( internal ) or open to people outside your organization ( external )  
Authorized domains – A list of domain names that are authorized to make a request for autentication to the provider. For example,  
weweb.io so it works when you're testing inside the WeWeb Editor, weweb-preview.io to test the auth flow on a published app without a custom domain, and your own custom domain to ensure it works properly on your live app  
Scopes – To define what users will be giving access to (e.g. just their basic info or also their calendar).  
Test users – A list of test users who can test the app while it's under development.

### Images

<https://docs.weweb.io/assets/open-id2.CVWGD_-d.png>

<https://docs.weweb.io/assets/open-id3.BhUCq5FR.png>

## API keys & credentials ​

In order for the provider to authorize the call you make through the OpenID plugin in WeWeb, you'll need create OAuth credentials in the provider's interface and copy them in WeWeb when you configure the plugin (but more on that later).  
In Google, you can do this by going to APIs & Services > Credentials > Create credentials > OAuth client ID :  
  
At that stage, the provider will ask for information such as:  
Name – You will be invited to name the credentials. It doesn't really matter what you name it but it's helpful if it's descriptive enough that you know where you're using them. For example, you could name them: OpenID via WeWeb  
Authorized JavaScript origins – A list of all the domains your web-app is hosted on. For example:  
the WeWeb editor: https://editor.weweb.io your app in staging: https://4132ed05-ba19-4dc2-9867-ba4d4f9a9d76-staging.weweb-preview.io your app in production: https://www.your-custom-domain.com  
Authorized redirect URIs – A list of URIs where users can be redirected after being authenticated. For example:  
the page in the WeWeb editor: https://editor.weweb.io/4132ed05-ba19-4dc2-9867-ba4d4f9a9d76/86d66b45-bedd-47e9-a2ae-c1af5a1b2f5d the page in the app in staging: https://4132ed05-ba19-4dc2-9867-ba4d4f9a9d76-staging.weweb-preview.io/post-login/ the page in production: https://www.your-custom-domain.com/post-login/  
These links should correspond to the page where users are redirected after being authenticated through the OpenID plugin in all the environments of your app (e.g. the WeWeb editor, in staging, in production).  
It should match the URIs of the page you defined in the OpenID plugin in WeWeb:

### Images

<https://docs.weweb.io/assets/open-id4.BNPc3t-C.gif>

<https://docs.weweb.io/assets/open-id5.Q_nLEczf.png>

<https://docs.weweb.io/assets/open-id6.y98VdBBU.png>

## Plugin configuration ​

In WeWeb, you'll find the OpenID plugin in the Plugins > Authentication menu:  
  
When you add the plugin, you'll be invited to provide information that will allow WeWeb to make calls on behalf of your app to your OAuth provider:  
the domain of your OAuth provider the OAuth 2.0 credentials created for your app (i.e. the client ID and client secret ) the scope values you want OpenID to return from the User Info endpoint, e.g. profile, email, address, or phone. Learn more about different scope values in OpenID's user docs the response type . Learn more about different OAuth flows in this video by Oracle and through OpenID's user docs  
In the example below, you can see:  
we are working with Google as a provider, and are asking for the user email and profile of the user

### Images

<https://docs.weweb.io/assets/open-id7.DbmHLn54.png>

<https://docs.weweb.io/assets/open-id8.DrUQW-R1.png>

## Redirections ​

Once you have told the OpenID plugin how to communicate with your OAuth provider, you will need to define pages where users are redirected:  
after they sign-in when they are not signed-in  
  
In the example above, we named those pages Post login and Login but we can choose to redirect them to any page we want. For example, we could redirect authenticated users to their profile page and unauthenticated users to the app's home page.

### Images

<https://docs.weweb.io/assets/open-id9.PBuAiMcI.png>

## Signup and login flows ​

Once you've added the OpenID plugin to your WeWeb project, you will get access to dedicated OpenID actions in workflows:  
  
When you use one of the login actions in a workflow, the user will be automatically redirected to the page you defined in the OpenID plugin after the authentication is successful.  
Even if you have a Change page action after the login action in your workflow, the OAuth flow will resolve on the redirect page defined in the plugin and authorized in the provider dashboard.

### Images

<https://docs.weweb.io/assets/open-id10.BgbN4Zok.png>

## Advanced redirection ​

If you want more flexibility on where to redirect authenticated users, you could update a destination page variable before the login action in your workflow to define where the user should be redirected after the login:  
  
Then, on the post-login page, you can trigger a workflow On Page Load that checks the destination page variable to decide if the user should stay on the post-login page or be redirected to another page like the user profile in the example below:

### Images

<https://docs.weweb.io/assets/open-id11.CiROrMWk.png>

<https://docs.weweb.io/assets/open-id12.ZBT7lEYi.gif>