Supabase data source | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/plugins/data-sources/supabase-data.html> |
| Date de scraping | 2025-04-08T15:55:30.248129+00:00 |

## Supabase ​

Supabase is an open-source backend-as-a-service platform that simplifies app development with integrated tools like a real-time PostgreSQL database, user authentication, file storage, and serverless functions. As an open-source alternative to Firebase, it enables you to build secure, scalable applications without managing complex backend infrastructure.

## Plugin configuration ​

The Supabase plugin connects WeWeb directly to your Supabase project, enabling you to manage your tables, API endpoints, database functions, secrets and integrations right from within the WeWeb interface.  
To add the plugin in WeWeb:  
Navigate to Back-end > Add Supabase :  
  
Once you've added it, you will have the option to connect to your project in Guided or Custom mode for self-hosted Supabase projects:  
  
Grant the necessary permissions to Supabase.  
  
Back in WeWeb, you will see that your account is connected and that you will be able to select your existing Supabase project or create a new one:  
  
Upon clicking on "Continue" to confirm the setup, we can see additional plugin information, such as the configuration, and realtime collections:  
  
If we click on the Backend panel, the tables from our database:  
  
In this guide, we will be working with the properties and properties\_images tables.

### Images

<https://docs.weweb.io/assets/supabase-plugin1.CvwHOHvc.png>

<https://docs.weweb.io/assets/supabase-plugin2.BJz-K2iI.png>

<https://docs.weweb.io/assets/supabase-plugin3.OychXaxP.png>

<https://docs.weweb.io/assets/supabase-plugin4.BrR8Plsi.png>

<https://docs.weweb.io/assets/supabase-plugin5.Bly0VONv.png>

<https://docs.weweb.io/assets/supabase-plugin6.B2t0CCag.png>

## Fetching data ​

We can create a collection in WeWeb using Supabase as a data source. This will essentially allow us to get data from specific tables and store them in a collection, which is just a place to store data.  
Go to Data > New Choose Supabase as a Source Select which database table you want to pull data from:

### Images

<https://docs.weweb.io/assets/supabase-plugin7.CinsPWxH.png>

## Guided mode ​

Guided mode will be enabled by default, which is a mode that pre-configures the collection to pull in data from every available table column without requiring manual selection.  
In the example below, we decided to exclude the data from the created\_at , created\_by and updated\_ columns in our properties table:  
  
As a result, our collection will look like this:

### Images

<https://docs.weweb.io/assets/supabase-plugin8.B2y4OoS-.png>

<https://docs.weweb.io/assets/supabase-plugin9.DQ7U6PTU.png>

## Advanced mode ​

Advanced mode lets you get information from multiple connected tables at once, rather than just one table at a time.  
For example, think of your property listings like this:  
The properties table stores the basic details of each property with the columns:  
id , address , bathrooms , bedrooms , description , price , square\_feet , title  
The property\_images table stores multiple images for each property:  
id , image\_url , property\_id Each image is linked to its property using property\_id image\_url is where the actual image link is stored  
The relationship works because each row in property\_images has a property\_id that matches an id in the properties table, allowing you to know which images belong to which property.  
A collection can show data from two connected tables. To do this, use this format: name\_of\_second\_table:id(which\_columns\_you\_want)  
In our properties example, if we wanted to get the related images of a property we would write: property\_images:id(image\_url)  
  
This tells WeWeb to:  
Look at the property\_images table Connect it to the main properties table Show only the image\_url column from property\_images  
So for each property, you'll see its own details plus its related images, all in one collection:

### Images

<https://docs.weweb.io/assets/supabase-plugin10.BFX0bAco.png>

<https://docs.weweb.io/assets/advanced1.B8cIT748.png>

## Row-Level Security ​

Row-Level Security (RLS) is a security mechanism that controls who can access or modify which rows in a table. Without RLS, anyone with database access can view, update, or delete all rows.  
Imagine a school database with a students table. Without RLS, everyone could see all students' records. But with RLS:  
A student can only see their own grades. A teacher can see only their students' grades. The principal can see everyone’s grades.  
By default, Supabase allows full access to all data so you can quickly set up and test your apps without restrictions. However, once you enable RLS on a table, Supabase blocks all access until you create explicit policies defining who can read or modify data.  
Enable RLS for your table  
  
Define policies for the table:  
  
In the example below, we have the following policies:  
Enable insert for authenticated users only INSERT Enable read access for all users SELECT Enable delete for property owners DELETE Enable update for property owners UPDATE

### Images

<https://docs.weweb.io/assets/supabase-plugin11.Dab3qoej.png>

<https://docs.weweb.io/assets/supabase-plugin12.7MRNoHE5.png>

<https://docs.weweb.io/assets/supabase-plugin13.TI5TIDmV.png>

## Filtering, Pagination & Sorting ​

When working with large amounts of data in Supabase, you might need backend filtering, pagination and sorting. WeWeb handles this at the collection level. You don't need to set these up in Supabase - simply select your collection and go to the Query configuration section.  
This means even with, for example, 50,000 records or more, you can efficiently control how your data is loaded and displayed.  
In the example below, we are getting only five records because the pagination limit is 5!

### Images

<https://docs.weweb.io/assets/pagination-supabase.CpEHdbFE.png>