Intro to collections | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/data/intro-to-collections.html> |
| Date de scraping | 2025-04-08T15:53:15.203650+00:00 |

## Intro to collections ​

In WeWeb, a collection is a set of data fetched from a 3rd-party API or backend. You can then use this data to display it in your app.

## Pre-requisites ​

## Create a collection ​

To create a collection, go to the Data tab in the left sidebar, then click on + New to start setting up the collection.  
The setup process is different for every data source plugin, but it revolves around the following steps:  
Give your collection a name Select the data source plugin you want to use Select the connection mode Configure how you want to fetch the data Fetch the data Apply any frontend sort, filter or pagination  
Let's look at an example:  
To deep-dive into how to setup a collection for each data source plugin, check out the following guides:  
Airtable REST API Xano Supabase Algolia

## How to use a collection ​

Collections are typically used to display data in a list or a grid. To do so, you can use the Collection list element, or any div.  
To learn more about how to display a collection, check out the following guide .  
But in the end, collection are simple arrays of data that can be used in any workflow or formula.  
They look like this inside the data explorer:  
Let's deep dive into this:  
first, you'll find the collection as a whole. It contains your data, and some metadata about the collection (like the number of items, the total number of items, etc.) that WeWeb created. secondly, you'll find the data property. This is the array of data without any metadata .  
data[index] shows the data for the item at the index index in the collection. It's useful to bind to a specific item in the collection. To change the item, you can use the index dropdown next to the data property. Length is the number of items there are in the collection.  
isFetching is a boolean that tells you if the collection is currently fetching data. It's useful to display a loading state while the data is being fetched. isFetched is a boolean that tells you if the collection has already been fetched.  
limit is the max index for the items that are currently displayed in the collection, given any active frontend filter, sort or pagination. total is the total number of items in the collection, given any active frontend filter, sort or pagination. offset is the index of the first item in the collection, given any active frontend filter, sort or pagination.

### Images

<https://docs.weweb.io/assets/collection1.COHp_YUX.png>

## Optimize collection load ​

By default, a collection is:  
fetched automatically when a user navigates to a page that uses that collection, and is preserved (not fetched again), when a user navigates to other pages in the app.  
To optimize the performance and UX of your web app, you can change that default behavior in the Fetch data step of the collection setup:  
  
Fetch this collection automatically tells WeWeb if it should fetch the collection on every page load, or if you want to fetch it manually. If you want to fetch it manually, you can trigger the Fetch collection workflow action.  
Preserve on navigation tells WeWeb if it should keep the collection in memory when the user navigates to a new page. If you want to fetch the collection again, you can trigger the Fetch collection workflow action.  
There may be use cases where you want to change these default settings.

### Images

<https://docs.weweb.io/assets/collections-fetch.BYeQoqpS.png>

## Why disable automatic fetch? ​

If you have a collection that you don’t need immediately on page load, you should consider disabling the automatic fetch to improve performances. For example, you might want to fetch a collection when the user opens a modal on a page and not on page load.  
If you disable the automatic fetch on a collection, you'll need to remember to trigger a workflow that fetches the collection every where you need it in the app.

## Why disable preserve on navigation? ​

A common use case where you don't want to preserve on navigation: a collection that fetches a single item selected on a previous page (if you preserve on navigation, there might be a blink where the info from the previously selected item appears while the API call takes place to fetch the info of the new selected item)  
This is especially problematic if the collection data concerns a specific user with potentially private information. You want to make sure that data is not preserved on navigation in case 2 users connect to your app using the same browser.

## Collection modes ​

Static : the collection will be fetched once, on our servers, while the app is built. Everytime this collection is used in your app, the data will be pre-rendered (better for SEO, can slow down the publish process). Dynamic : the collection will be fetched on the client side, when the app is loaded. This is the best option if you want to fetch data that changes often (like a list of products). Cached : the collection will be fetched on the client side, but from our servers. This is the best option if you want to fetch data that changes often, but you want to avoid hitting any API rate limit (but you'll need to refresh the data yourself when it changes).

## Frontend vs Backend filters ​

When setting up a collection, the Sort, Filter and Pagination step can be before or after the fetching one.  
If you set it before, it's called a backend filter . It means that the data is fetched from the 3rd-party API, already filtered, sorted and paginated by the server.  
If you set it after, it's called a frontend filter . It means that the data is fetched from the 3rd-party API, then filtered, sorted and paginated on the client side:

### Images

<https://docs.weweb.io/assets/collections-filters.DvfiQ0By.png>

## When to use frontend vs backend filters? ​

If you have a small collection, without any private data: use a frontend filter. It's the easiest to setup, and it's the fastest. If you have a big collection, or need to fetch private data: use a backend filter. It's a bit more complicated to setup, but it's the more secure.

## Filter options ​

Pagination is the number of items that are displayed per page. It's useful if you have lot of data to display, and you want to split it into multiple pages (using the Pagination element). Filter is a formula that filters the data. It's useful if you want to display only a subset of the data.  
Let's look at an example:  
Here, we're filtering the data to only display companies where their name contains the letter t or, (where their domain contains www and their number\_of\_employees is greater than 100 ).  
As you can see, you can create groups of filters using condition groups .  
For each of these groups or conditions, you can set them under certain conditions (using the Apply if... ). For example, when a variable is true or false in your app.  
You can also use the AND and OR operators to combine filters.  
The toggles are Ignore if empty options that tell WeWeb not to apply the filter if the value is empty.  
Sort is a formula that sorts the data. It's useful if you want to display the data in a specific order (like ascending or descending order for alphabetical or numerical values, etc).

### Images

<https://docs.weweb.io/assets/collection5.BfyuFmka.png>

<https://docs.weweb.io/assets/collection6.DmUTfLZq.png>