Xano data source | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/plugins/data-sources/xano-data.html> |
| Date de scraping | 2025-04-08T15:55:31.619611+00:00 |

## Xano ​

Xano is a visual development backend platform that lets you build custom APIs, manage databases, and handle server-side logic without writing traditional code. Through its visual interface, you can create data storage systems, user authentication, and API endpoints - essentially everything needed to power modern web applications.

## Add Xano plugin ​

In order to get data from Xano, you first need to add Xano as a data source in WeWeb, in Plugins > Data sources :

### Images

<https://docs.weweb.io/assets/xano-data1.BP6rS7ec.png>

<https://docs.weweb.io/assets/xano-dataold.U4BH0UiG.png>

## Plugin configuration ​

When you add the Xano data source plugin, you will be invited to:  
Add your personal access token Select the Xano instance you want to connect Add the custom domain of your instance (if you have one) Select the workspace you want to connect

### Images

<https://docs.weweb.io/assets/xano-plugin-config.BpOBNe4-.png>

<https://docs.weweb.io/assets/xano-access-token.CFdE1Tbn.png>

## Generate access token in Xano ​

To generate or find your access token in Xano, you have 2 options.  
Option 1 - go to Account > Metadata API :  
  
Option 2 - go to the Metadata API section in the Settings of your instance:

### Images

<https://docs.weweb.io/assets/xano-data3.C3i-7AH4.png>

<https://docs.weweb.io/assets/xano-data4.DaYbb2L3.gif>

<https://docs.weweb.io/assets/xano-data5.DTob_4XG.png>

## Data sources (optional) ​

Depending on your Xano plan, you may have the option to support multiple data sources .  
For example, you could have a live data source and a test data source:  
  
In WeWeb, you could decide to work with different data sources depending on whether you're working:  
inside the WeWeb editor, on the app published in the WeWeb staging environment, or on the published app in production.  
In the example below, we chose to work with the data from the test data source inside the WeWeb editor, and on the app in staging:  
  
As a result, when working in the WeWeb editor or viewing the app published in staging, the data that is displayed will be the data from the test data source in Xano. We will not be able to see or work with the data in the live data source.

### Images

<https://docs.weweb.io/assets/xano-data6.CkMSzfkY.png>

<https://docs.weweb.io/assets/xano-plugin-sources.Dlx9DTJJ.png>

## Branching (optional) ​

Depending on your Xano plan, you may have the option to support multiple branches .  
In Xano, a branch is "a copy of a workspace's business logic (APIs, Functions, Addons, and Tasks)."  
Creating a branch in Xano allows you to make changes to your API endpoints, functions, add-ons and tasks in a safe environment, without affecting your app in production.  
In the example below, we created a v2 branch in Xano to add features related to admin permissions in our app:  
  
In WeWeb, we continue to use the Xano v1 branch in our app in production, but we use the Xano v2 branch to test things in the WeWeb Editor and in our app published in staging:  
  
When we're ready, we can decide to make the v2 branch live in Xano and update the Xano data source plugin in WeWeb.

### Images

<https://docs.weweb.io/assets/xano-data8.-89Os7qX.gif>

<https://docs.weweb.io/assets/xano-plugin-branches.D6gwke9g.png>

## Global headers (optional) ​

If you need to add custom global headers to all your Xano calls, you can configure these in the Xano data source plugin:  
  
Or at collection level:

### Images

<https://docs.weweb.io/assets/xano-plugin-headers.DFKbt-fk.png>

<https://docs.weweb.io/assets/xano-collections-headers.BW2UaXyU.png>

## Get data from Xano ​

Once you have configured the Xano data source plugin, you will be able to create a collection to get Xano data into WeWeb:

### Images

<https://docs.weweb.io/assets/xano-collections-add.BW4YdBCA.gif>

## Select a source ​

In the Data tab, when you create a new collection, you will be invited to:  
name the collection, select a data source (for example Xano), and choose a collection mode (for example dynamic).  
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).

## Configuration ​

When you create the collection, you will be invited to choose the API group and endpoint you want to use in WeWeb.  
In the example below, we call a GET endpoint in Xano that returns all the tickets related to the user who is currently authenticated to our app:

### Images

<https://docs.weweb.io/assets/xano-collections-config.CD_NjBXK.png>

## Fetch data ​

By default, Xano collections are fetched automatically and preserved on navigation:  
  
This means that:  
if a collection is used on a page of your app, it will be fetched automatically, and if the user navigates to another page, the collection will not be refetched.  
There may be use cases where you want to change these default settings.  
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.

### Images

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

## Sort, filter, and pagination ​

Once you have fetched data from Xano, you can add sort it, filter it, and paginate it in the frontend:  
Once you have added a data collection in WeWeb, you will be able to display that data on a WeWeb page. Learn how to display collection data .

## Update Xano data ​

To update a table in Xano from a WeWeb app, you'll have to create a workflow that makes a request to Xano.  
In the screenshot below, you can see that:  
we make a request to Xano that will affect the Joyce's tutorial - careers API group we're making a POST call in the application table updating three fields in the table we pass the values of these three fields  
  
Use case: when an authenticated user applies to a job, we make a request to Xano to update the application table with the id of the selected job and authenticated user as well as the URL of the resume that was uploaded by the user.

### Images

<https://docs.weweb.io/assets/xano-crud-fields.oAyR9G88.webp>

<https://docs.weweb.io/assets/xano-crud.B6ZRSYCO.gif>

## Using Xano Auth? ​

If you have added the Xano data source plugin and are using the Xano Auth plugin for authentication , you do NOT need to bind the user's auth token when adding a Xano collection or using a Xano action in your workflows.  
The Xano data source plugin will automatically recognize the current user's authentication token and send it to Xano when making an API call.  
You only need to manually bind the current user's authentication token if you are making API calls to Xano via the REST API plugin and not the Xano data source plugin.

## Migrating from Airtable to Xano? ​

It's fairly common for Airtable enthusiasts to switch to backends like Xano or Supabase when scaling web-apps.  
WeWeb's Xano plugin facilitates the switch from Airtable to Xano.  
Step 1 – You can use Xano's import function to import Airtable tables. The import will keep the same field names and references between tables:  
Step 2 – In WeWeb, you can then change the data source of your collection from Airtable to Xano.  
Your collection lists and collection list items will remain unchanged on the page because WeWeb will automatically recognize the data structure.

### Images

<https://docs.weweb.io/assets/xano-data-6-airtable-import-in-xano.zPTZlzuR.gif>