Build a Job Board with WeWeb and Xano | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/build-guides/job-board-with-xano.html> |
| Date de scraping | 2025-04-08T15:52:55.570763+00:00 |

## Build a Job Board with WeWeb and Xano ​

In this use case walkthrough, we use two templates to get started quickly:  
WeWeb’s “Job Board” template for the frontend, and Xano’s “Careers” template for the backend.  
You can follow a video walkthrough here or read a step-by-step guide below:

## 1- Setup new projects in Xano and WeWeb ​

## In Xano ​

Create a free XANO account and choose the “Careers” template in the Marketplace:  
  
You’ll notice 3 tables and 16 API endpoints are created automatically:  
job application category  
This is where you’ll find your list of jobs, applications, and job categories respectively.  
The 16 API endpoints are how you’ll be able to fetch data from your Xano tables to display it in WeWeb.  
If you’re not familiar with APIs, you might want to pause and learn more here .

### Images

<https://docs.weweb.io/assets/build-job1.C9lBNK0C.png>

## In WeWeb ​

In WeWeb, create a new project and choose the “Job board” template.  
You’ll notice a page where you can add your list of jobs and a page where you can display the details of a job offer:

### Images

<https://docs.weweb.io/assets/build-job2.BzM4MlQ0.png>

## 2- Display a list of jobs in WeWeb ​

In order to display data in WeWeb, we'll need to:  
Add the Xano data source plugin Get data from a Xano endpoint Bind that data to an element on the page

## Step 1: Add Xano data source ​

Go to Plugins > Data sources > Xano :  
  
You will be invited to add your Xano API and choose the instance you want to work with.

### Images

<https://docs.weweb.io/assets/build-job3.LEJGznVO.png>

## Step 2: Get the data from Xano ​

Go to Data > Collections > Add collection to create a Jobs collection that will call the API endpoint in Xano that returns the 6 items we have in the job table:

### Images

<https://docs.weweb.io/assets/build-job4.WF1-D05v.png>

## Step 3: Bind the collection list and collection list item on the page ​

On the “Jobs” page, bind the “Collection List” of jobs to the items in the Jobs collection:  
  
You will notice that, once you bind your collection, even if you only have one Job container in the navigator, you’ll have as many repeated items on the page as there are items in your Xano database.  
Inside the “Job container”, i.e. the first child element of the container where you bound the collection, you can then bind the information from a repeated item.  
In the example below, we bound the job title and type of contract:

### Images

<https://docs.weweb.io/assets/build-job5.--wxmvkb.png>

<https://docs.weweb.io/assets/build-job6.B1Q2Dvhj.png>

## 3- Search, sort & filter a list of jobs in WeWeb ​

When you have a list of items on a page, it is common to let users filter that list by searching for a term or selecting an option for example.

## Searching through a list in WeWeb ​

There is a search bar element by default on the “Jobs” page of the “Job board” template in WeWeb.  
However, if you ever needed to add a search, know that you can find them in the Add menu of the Editor:  
  
Whenever a search bar element is dragged and dropped onto the page, a new component variable is created in your project.  
By default, it is named “Search bar - value” but you can change the name in the element’s settings. You can also change the placeholder of the element in its settings:  
  
In preview mode, you can type in search terms in the search bar to see the “Search bar - value” variable change in your navigator:  
  
Once you’ve set up the search bar element on the page, you’ll need to add a filter to your list of jobs.  
You could do this several ways.

### Images

<https://docs.weweb.io/assets/build-job7.CANK9Grt.png>

<https://docs.weweb.io/assets/build-job8.BHXJI57o.png>

<https://docs.weweb.io/assets/build-job9.Cfquh1G1.gif>

## Backend filter ​

If you were working with a large collection of items, say more than 10Mb, the best practice would be to add a search filter to your backend so that the REST API collection in WeWeb only fetched data that matches the search criteria.  
This takes a little bit more time, usually a couple of seconds, but has two main benefits:  
You can choose what data you load in your user’s browser (hint: nothing sensitive!) You don’t load as much data in your user’s browser and therefore don’t risk crashing it.  
For the purposes of this tutorial, since WeWeb is a frontend builder, we’ll explain how to add a search filter to your frontend but please bear in mind that, if you’re working with large collections, you should spend a little more time learning how to add a search filter in your backend .

## Frontend filter ​

To search in the frontend in WeWeb:  
Add a filter to your Collection List, Where the field you want to search “Contains” the “Search bar - value” component variable, and Apply this filter if… the “Search bar - value” component variable is not empty.  
In the example below, we are filtering the jobs list based on whether the search term is in the title or summary of the job description:

### Images

<https://docs.weweb.io/assets/build-job10.BJj5LndS.gif>

## Filtering a list with a select element ​

Now let’s say you also want to filter the job list by job category.  
First, you’ll want to create a new collection for job categories in WeWeb, using the API endpoint to query all categories from Xano.  
Not sure how to do that? Might want to revisit how we created the Jobs collection above 😉  
Once you have your Categories collection setup in WeWeb, you’ll want to drag and drop the “Select” element on the page:  
  
As with the search bar element, this will automatically create a component variable in the navigator:  
  
In the screenshot below, you can see that:  
On the select element We bound the “Categories” collection Used the rollup formula to get only the values from the category field  
  
Now, to filter your list of jobs based on the job category selected, all you need to do is add a filter to your existing Collection List where the category field of the job contains the Input select - value variable:  
  
When the user lands on the page, before they select anything, we don’t want the filter to apply, so we can add a condition to the condition group, i.e. Apply if… the Input select - value variable is empty.

### Images

<https://docs.weweb.io/assets/build-job12.Djh6vMxA.png>

<https://docs.weweb.io/assets/build-job13.CbXZ6z9y.png>

<https://docs.weweb.io/assets/build-job14.DdNG2PLr.png>

<https://docs.weweb.io/assets/build-job15.C7UJJq1M.png>

## 4- Redirect a user to a job description page in WeWeb ​

Ok great. We have a page with a list of job offers.  
Now we want to let users access individual job pages.  
There are several ways we could do this in terms of user interface.  
Let’s do it in the most obvious way by creating a button that users can click on to learn more about a specific job.

## Step 1 – Add a button ​

In the Add menu, search for the Button element, and drag and drop it in your repeated item container.  
In the example below, we got rid of the information we didn't need to keep a cleaner design:

### Images

<https://docs.weweb.io/assets/build-job16.CW0u_zHS.png>

## Step 2 – Add a parameter to the Job template page ​

To change the Job template page into a dynamic page, we will add a parameter to the URL with the id of the job:

### Images

<https://docs.weweb.io/assets/build-job17.D5uIAkZO.png>

## Step 3 – Create a collection to get the job selected ​

In the Data menu, call the Xano endpoint that gets the job based on the id of the current page:

### Images

<https://docs.weweb.io/assets/build-job18.BDsCX1Md.png>

## Step 4 – Create a workflow on the button ​

Back on the Jobs page, link the button to the Job template page and pass the current item's id as a parameter:

### Images

<https://docs.weweb.io/assets/build-job19.BVbKG3F8.png>

## Step 5 – Create a workflow on page load ​

Now, when a user navigates to the Jobs template page, you'll want to fetch the Job collection:

### Images

<https://docs.weweb.io/assets/build-job20.BUhuucmX.png>

<https://docs.weweb.io/assets/build-job21.Ck2r_p32.png>

## Step 6 – Display the job data on the template page ​

Once you’re on the template page, you can bind fields as you would on any other page, finding the data you need in the Job collection:

### Images

<https://docs.weweb.io/assets/build-job22.DaNWeDzU.png>

## 5 – Allow users to upload their resume and apply to a job through your WeWeb application ​

Our web application wouldn’t be complete without the option for users to create, update, or delete a record.  
For our use case, we want to let users upload their resume and apply to a job by submitting a form on our WeWeb app.  
By default, on the “Job template” page of the “Job board” template, there is a Contact form container with 4 component variables:  
User name User email User location User bio  
You can see these component variables when you open the Variables sub-menu of the Navigator panel:  
  
To allow users to upload a resume, we’ll drag and drop a “File upload” element inside our form container:  
  
You’ll notice that, as soon as you drop the element on the page, two new component variables are created by default:  
“File 1 - progress”, and “File 1 - value”  
Both these variable names will change if you change the file upload element’s name.  
You can test this in Preview mode by typing in data and uploading a file as if you were a user. You’ll notice the value of the component variables in the navigator are updated in real time:

### Images

<https://docs.weweb.io/assets/build-job23.DW9BaEYC.png>

<https://docs.weweb.io/assets/build-job24.7hJfAnm8.gif>

<https://docs.weweb.io/assets/build-job26.BskhLrXf.gif>

## 6 – Update your Xano database with the user’s job application ​

By default, the “resume” field in Xano’s “application” table is a “file resource.”  
However, in WeWeb, when a user uploads a file, we upload it to our CDN and the value of the file is a URL, i.e. a “text” field.  
Before we show you how to send data from WeWeb to Xano, you therefore need to change the type of the “resume” field in Xano so it’s a “text” type:  
  
Then, to update your Xano database with the user’s job application, follow these steps:  
Create a workflow on the form container Name: “Submit application”, or anything you find more descriptive Trigger: On Submit, i.e. when the user submits the form  
Action 1:  
Upload file Select upload element from this page: select the component variable you setup Test Action 1 before moving on to Action 2  
This is very important because you will need the result of Action 1 to set up Action 2, where you will send data to Xano using the POST /application endpoint:  
  
Fields: for each column in the Xano application table, bind the corresponding value in WeWeb.  
What does this mean?  
It means that, in Xano, you have 6 columns with data. The keys for these columns are as follows:  
job\_id name resume email phone status  
In WeWeb, you have a corresponding variable for each key except phone . In the walkthrough below, you can see we:  
select which fields we want to update in Xano, give Xano the id of the job we are applying to in our Job collection, map the form input variables to the fields in Xano, hard code the fact that every application should have the status "To be reviewed" get the link from the file we uploaded in the previous action  
  
Once that’s done, you can test Action 2, refresh the application table in Xano and jump with joy when the test application comes through 😀

### Images

<https://docs.weweb.io/assets/build-job25.CjsSoEds.png>

<https://docs.weweb.io/assets/build-job26.CszjQbzd.png>

<https://docs.weweb.io/assets/build-job27.CmhUF450.gif>

## 7 – Add Xano authentication ​

Let’s say you want to force users to authenticate themselves before they can apply to a job.  
In order to do that, you’ll need to:  
Add a user table to your Xano database Add authentication endpoints to Xano Set up Xano authentication plugin in WeWeb Prevent users who are not authenticated from applying to a job (create login page and gate content)

## Step 1 - Setup a User Table in Your Xano Database ​

By default, there is no user table in Xano’s “Careers” template but you can easily add one.  
Got to Database > Add Table in the top right corner.  
Your user table should include at least two fields:  
one field where you store the login. For example, an email field type, and one field of type password where you store the encrypted password.

### Images

<https://docs.weweb.io/assets/build-job28.C05dTNZX.jpeg>

<https://docs.weweb.io/assets/xano-2.CbTMPcnS.gif>

## Step 2 - Create Authentication API Endpoints in Xano ​

For Xano authentication to work, you need three API endpoints:  
signup login me  
If you’re using Xano’s “Careers” template, you’ll need to create these three endpoints yourself by going to API > Add API Endpoint > Authentication :

### Images

<https://docs.weweb.io/assets/xano-5.DFoYFdcX.gif>