Xano authentication | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/plugins/auth-systems/xano-auth.html> |
| Date de scraping | 2025-04-08T15:55:21.969360+00:00 |

## XANO authentication ​

## Xano pre-requisites ​

In order to set up Xano authentication in your WeWeb project, there are three prerequisites:  
you already have an account with Xano ✅ you have a user table in your Xano database with at least two fields: login and password ✅ you have the three authentication API endpoints in Xano ✅  
Some Xano templates include a user table and the three authentication API endpoints.  
However, if you don't have both in your Xano workspace, you'll need to set them up manually.  
Here's how.

## User table in Xano ​

Your user table should include at least two fields:  
one field where you store the login, for example, a string field with a name or an email field with an email address, and one field of field type password where you store the encrypted password.

### Images

<https://docs.weweb.io/assets/xano-1.BWtaSA6r.webp>

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

## Authentication endpoints ​

For Xano authentication to work, you need three API endpoints:  
signup login me  
  
If you’re using a Xano template that includes authentication, you’ll find these endpoints in the API sub-menu.  
Otherwise, you’ll need to create these three endpoints yourself by going to API > Add API Endpoint > Authentication

### Images

<https://docs.weweb.io/assets/xano-4.46qz9GEH.webp>

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

## Plugin configuration ​

To add the Xano Auth plugin in WeWeb, go to Plugins > Authentication :  
  
You will be invited to provide the following information:  
a Xano access token what Xano instance and workspace you want to work with, your Xano custom domain if you have one, the me , login , and signup endpoints from your Xano project, and what 3rd-party providers you want to work with (assuming you have any activated in the Xano workspace you selected)

### Images

<https://docs.weweb.io/assets/xano-auth-plugin.DRKpRHVL.png>

<https://docs.weweb.io/assets/xano-auth6.3jn5_sO1.png>

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

## Data source & branching (optional) ​

Depending on your Xano plan, you may have the option to support multiple data sources and branches .  
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.  
The same logic applies if you want to work with different Xano branches. In the example below, we will test the Xano v2 branch in the WeWeb editor and the app we publish on the staging environment but the Xano v1 branch will still be live in our app in production:  
  
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-data6.CkMSzfkY.png>

<https://docs.weweb.io/assets/xano-auth-data-source-headers.SCg89knB.png>

<https://docs.weweb.io/assets/xano-auth-branch-headers.9rZFxUMT.png>

## Global headers (optional) ​

If you need to add custom global headers to a Xano call, you can configure these:  
in the Xano data source plugin, in the Xano Auth plugin, at collection level, or at action level when making a request in a workflow.

## Define redirections ​

Here, you should choose a PUBLIC page where users who are not signed in can be redirected.  
In the example below, we chose to redirect unauthenticated users to a login page but it could be any page as long as it's a public page:

### Images

<https://docs.weweb.io/assets/xano-auth7.DGe7DYR6.png>

## User role configuration ​

Assuming you have information about a user's role in Xano and that information is available in the me endpoint, you can tell the Xano Auth plugin in WeWeb where to find that information.  
In the example below, we see that the user returned by the me endpoint in Xano provides a single role in a single text in the role column:  
  
So we tell the Xano Auth plugin in WeWeb that it can find the role of our user in a text column named role :

### Images

<https://docs.weweb.io/assets/xano-auth8.B06Xt_yD.gif>

<https://docs.weweb.io/assets/xano-auth9.B3yWsWsO.png>

<https://docs.weweb.io/assets/xano-auth10.3sVhSIf2.png>

## Signup and login flows ​

When you add the Xano Auth plugin to a WeWeb project, you will get access to a new Xano Auth category of UI elements in the Add menu:  
  
The login and signup forms both include a workflow with login and signup actions respectively but you'll need to adapt it to suit your login and signup flows.  
For example, if you want to display a modal for users to login, you could add a Change variable value action to close the modal once the login is complete.  
In the example below, we added a Change page action to redirect the user after they log in:

### Images

<https://docs.weweb.io/assets/xano-auth-sections.DFYV7czX.png>

<https://docs.weweb.io/assets/xano-auth11.BBeWlcsD.png>

## Conditional display ​

One you've setup user authentication, you'll probably want to display some things conditionally.  
For example, you'll want to display a logout button when the user is authenticated.  
You can do this by binding the display of your login and logout buttons to the isAuthenticated variable in the User tab:

## Private pages ​

By default, once you've added the Xano Auth plugin to your WeWeb project, you'll be able to make pages private.  
You'll have the option to choose if a page should be accessible to everybody or authenticated users only:  
  
If you have gone through the User role configuration step in the Xano Auth plugin, you will also be able to:  
add user roles in WeWeb (these role names should match the role names you have in Xano), create user groups and decide what role or roles a user needs to be part of that group, and make a page private based on these user groups  
  
Note that you can also make a page private to a user group at page level:

### Images

<https://docs.weweb.io/assets/private-page1.CZSU0a-Q.png>

<https://docs.weweb.io/assets/xano-auth-user-groups.Df4X5I5t.png>

<https://docs.weweb.io/assets/private-page2.BU420XD5.png>

<https://docs.weweb.io/assets/private-page3.Bq7upkPc.png>

## Fetch Current User Data ​

Once you have setup authentication in your project, you'll probably want to display data that is related to the current user.  
For example, if you were building a web-app where users can view and download their pay slips, you'd want to fetch and display only the pay slips from the user who is currently logged in.

## Filter data in the backend ​

In theory, you could get all the pay slips from all the users and filter the Collection in WeWeb based on the current user but you definitely do not want to do that.  
Why?  
Because if you did that, you would be loading all the information from all your users in the frontend.  
Even if you added a filter and only displayed the current user's information on the page, other people could access it in the browser.  
So what should you do?  
You should filter the data on the current user at backend level, i.e. make sure Xano only sends you the information from the user who is currently authenticated.  
How do you do that?  
In Xano, you would have a table with all your users, and a table with all the pay slips. Each pay slip would be linked to a user.  
Now let's take a different example. Let's work on a web-app where users can apply to jobs.  
In Xano, we have a list of job applications. Each job application is linked to a user\_id in the user table:

### Images

<https://docs.weweb.io/assets/xano-15.DCQ1mPKc.webp>

## Clone the API endpoint closest to the data you need ​

In our case, we want to return a list of job applications for the current user.  
In the API menu, we have a default API to GET all the records from the application table.  
Let's clone this endpoint to create a new one that will only GET the applications of the authenticated user:

## Secure the endpoint with authentication ​

Now that we have this new user-application endpoint, we need to make sure our GET request will only go through if the user making it is authenticated:  
Pre-requisite: you have set up a user table with authentication endpoints in Xano.  
Why are we doing this?  
So that we can get the user's id from the authentication token provided by Xano when the user logs in. This ensures that no one can guess what API request to call with a user email for example.

## Add a filter to query user specific records ​

Instead of querying all the records from the application table, we want to query only the records where the linked user's id matches the id in the authentication token generated when a user logs into our web-app:

## Getting the filtered data to WeWeb ​

Now that we have an endpoint that requires authentication and sends back filtered data based on the authenticated user, we can create a data collection in WeWeb in the Data panel:  
  
And that's it!  
You've now ensured that only an authenticated user can access their own data.

### Images

<https://docs.weweb.io/assets/xano-auth-user-data.iHPWw2fP.gif>

## Troubleshooting Xano Authentication ​

If you're experiencing issues with the Xano signup or login forms, you can test the entire workflow or one action at a time and look at the response when it fails.  
Xano will return an error message in response > data > message :

### Images

<https://docs.weweb.io/assets/xano-auth-troubleshoot.Bv7GZz9Z.png>

## Most common errors ​

trying to sign up a user that already exists in the user base in Xano (need to login instead), trying to sign up a user with a password that doesn’t match Xano’s default requirements (at least 6 characters and 1 capital letter at the time of writing), trying to login a user that doesn’t exist in the user base in Xano (need to signup instead) trying to login with invalid credentials.  
If you don't find a solution using the WeWeb debugger, consider using the Run & Debug command in Xano directly to make sure everything is properly set up in the back-end.

## Forgot Password with Xano Auth ​

Once you've setup Xano and Sendgrid as explained above – barring the redirect URLs – you'll be able to set things up in your WeWeb user interface.

## Step 1: Create Dedicated Pages in WeWeb ​

In WeWeb, you'll want to create 2 pages.  
The first page with a "Change password" section:  
The second page with an "Reset password" section:  
Once you have created the "Update password" page, you'll be able to update the magic\_link\_redirect\_uri variable in your Xano settings:  
  
The value will be the URL of the page which contains the "Update password" form.

### Images

<https://docs.weweb.io/assets/xano-auth-change-password.Ckoy3F67.png>

<https://docs.weweb.io/assets/xano-auth-reset-password.CW5XJzv0.png>

<https://docs.weweb.io/assets/xano-auth-magic-link-redirect.qksa1Qj4.png>

## Step 2 – Workflow to Request Magic Link ​

On the "Forgot password" Form Container, you'll notice there is a Workflow.  
Here's how you update it to ensure the user receives an email with a magic link when they submit the form:  
copy the Xano API endpoint that requests a magic link, and bind the email query string field to the email value variable in the form.  
Depending on how you created your endpoints in Xano – using the Starter template or installing the Magic Link with Sendgrid extension for example – the endpoint may have a different name – request-magic-link or magic-link for example.

## Step 3 – Workflow to Reset the User's Password ​

You'll also need to update the Workflow on the "Update password" Form Container.

## Action 1: Login with the magic link ​

First, you'll need to copy the API endpoint that allows the user to login with a magic link:  
  
Again, the API endpoint may have a different path depending on how you created it in Xano. Make sure it's the one that allows the user to login with a magic token instead of a password.  
Then, in WeWeb, you'll need to create a Variable of type Query, named token :  
  
Why of type Query?  
Because when the user clicks on the magic link in the email, they will be redirected to the URL you pasted in the Xano settings with an additional query string at the end that contains a magic token.  
It will look something like this:  
  
Why name it token ?  
Because, as you can see from the screenshot above, in the magic link provided by Xano and Sendgrid, the parameter at the beginning of the query string is token .  
If we want WeWeb to get the value from this query string, we need to give our Query Variable the same name.  
Finally,  you'll need to bind that token variable to the magic\_token field:  
  
Why name the field key magic\_token ?  
Because that's the name of the input variable in Xano by default:

### Images

<https://docs.weweb.io/assets/xano-auth-magic-link-login.CypOYJCS.png>

<https://docs.weweb.io/assets/xano-auth-magic-link-token.BYjdCBFr.png>

<https://docs.weweb.io/assets/xano-auth-magic-link-preview.tTFux3Ma.png>

<https://docs.weweb.io/assets/xano-auth-token.DPSWUTRy.png>

## Action 2: Update the password ​

When the user submits the form, the first step will be to log in using the magic token in the magic link. That's what we saw in action 1.  
The second action will be to update that user's password in the user table with the information submitted in the form.  
Endpoint  
First, you'll need to paste the update-password endpoint from Xano.  
You should bind the password and confirm\_password values to the Form Variables and the Authorization field to the authToken returned by Action 1 of the Workflow:  
  
I'm sure the Xano team will fix this quickly but we ran into a couple of glitches during our setup using their Starter template.  
Here's how to make sure the update-password endpoint is setup correctly:

### Images

<https://docs.weweb.io/assets/xano-reset.sHHSmvlx.jpeg>

## 1- User field\_value ​

First, make sure the user inputs are setup correctly in the update\_password endpoint.  
The field\_value which should be auth > id :  
  
Depending on how you created the endpoint, it may be missing or appear as text > id .

### Images

<https://docs.weweb.io/assets/xano-auth-update-password-input.MxXaEN_H.png>

## 2- Input types of password and confirm\_password variables ​

In the Starter template, these two variables are of type Password.  
After chatting with Xano's support team, it turns out you should delete these two variables and add two Text Inputs instead:  
So your two text inputs should look like this:

### Images

<https://docs.weweb.io/assets/xano-auth-update-password-xano.DT8OIe39.png>

## Action 3: Redirect user to a login page ​

Once the user has updated their password, you'll usually want to redirect them to a login page.  
If you'd like to redirect them to another page, we recommend you add a step before the Change page Action where you will login the user with their new credentials.  
And that's it!  
You've successful setup a forgot password flow for your users using WeWeb, Xano Auth, and Sendgrid 🙂

## OAuth (social sign-in) ​

## Pre-requisites ​

Before you can setup a signup and/or login flow with a third-party provider in WeWeb, you'll need to:  
created and configured a project with the OAuth provider of your choice (e.g. Google, Slack, LinkedIn, etc.) installed and configured the related OAuth extension in Xano with the OAuth credentials provided by the provider  
Assuming you have added and installed an OAuth extension in Xano , you will be able to see the provider(s) in the configuration of the Xano Auth plugin:

### Images

<https://docs.weweb.io/assets/xano-auth-oauth1.Sb3qt3WM.png>

<https://docs.weweb.io/assets/xano-oauth2.B5IiETpB.png>

## Login or signup flow ​

The first step to allow users to signup or login with an OAuth provider is to trigger a workflow, for example, on click of a "Continue with Google" button.  
In that workflow, you'll need to:  
choose the Login Provider action, select which provider you want to work with, and redirect the user to a public, blank page  
  
This last step is crucial (more on this in a second).  
In our case, we named the page Login with provider redirect page to be super explicit but of course, you can choose a shorter name. It doesn't really matter since the user will not even notice they are routed through this page.

### Images

<https://docs.weweb.io/assets/xano-oauth3.C8vrn5-j.png>

## Public blank page for 3rd-party ​

Before your app is fully secured, it needs to render a page to save the authentication cookies and get back the user from the OAuth provider. That needs to happen on a public page.  
Then you can redirect the user to the page of your choice, public or private.  
The process is the following:  
user selects Oauth provider on your WeWeb app user is redirected to the Oauth provider consent screen user is redirected back to your WeWeb app, on a public page, where cookies are setup user is redirected to a private page on your WeWeb app  
We took care of steps 1 through 3 when we created the workflow with the Login Provider action above and redirected users to an empty public page.  
Now we need to trigger a workflow on that page so that the authenticated user is immediately redirected to the page we want them to see post signup/login:  
  
The workflow should have:  
a trigger On Page Load (before fetching collections) because you want to redirect the user as soon as possible a Change page action that redirects the user to the page you want them to see after signing up or logging in  
  
That's it, you've got OAuth configured on your Xano project.

### Images

<https://docs.weweb.io/assets/xano-oauth4.C992iREe.png>

<https://docs.weweb.io/assets/xano-oauth5.C-bcoFq6.png>