Token-based authentication | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/plugins/auth-systems/token-based-auth.html> |
| Date de scraping | 2025-04-08T15:55:19.216906+00:00 |

## Token-based authentication ​

## Add plugin ​

To connect your WeWeb app to a token-based authentication system, go to Plugins > Authentication > Token Based Auth :

### Images

<https://docs.weweb.io/assets/token-auth-plugin.BZnfF4-H.png>

## Configure plugin ​

When you add the Plugin, you will be invited to choose the authentication type:  
Auth Bearer Token, Auth Basic Token, or Custom Header  
  
Depending on the authentication type you choose, you will be invited to add further information.  
For Auth Bearer Token or Auth Basic Token authentication, you will be invited to provide a user endpoint (required) and refresh token endpoint (optional):  
  
For Custom Header authentication, you will also be invited to fill out the custom header name:

### Images

<https://docs.weweb.io/assets/token-2-auth-configuration.BHk5f2Yu.png>

<https://docs.weweb.io/assets/token-3-bearer.CBY_aVHE.png>

<https://docs.weweb.io/assets/token-4-custom-header.B9fk3iSX.png>

## Define dedirection ​

Choose a public page where you want to redirect users when they are not signed in:

### Images

<https://docs.weweb.io/assets/token-5-redirections.DSh2qcsV.png>

## User roles & user groups ​

## In your backend ​

In order to gate content in WeWeb based on user roles and permissions, you first need to have information about roles in your backend and that information needs to be accessible to WeWeb via API.  
For example, you could create a role column in your user table and make it available in the me endpoint so that, when a user logs in to your app, you fetch their role at the same time.

## In the WeWeb plugin ​

Assuming you have roles defined in your backend, you'll have to tell the Token Based Auth plugin where to find the user role in the backend:

### Images

<https://docs.weweb.io/assets/token-auth-role-config.D7t5ulSt.png>

## In the WeWeb Auth tab ​

Once you've told the plugin where it can find the user roles in your backend, you'll need to create the user roles and user groups in WeWeb:  
go to the Auth tab, add the user roles you have in your backend, and add user groups and decide what role(s) a user needs to be part of a group.

### Images

<https://docs.weweb.io/assets/token-auth-roles-groups-pages.DqCsid8b.png>

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

## Private pages ​

When you setup the Token Based Auth plugin, an Authenticated users user group will be created by default even if you don't define additional user groups.  
You can use this user group to restrict access to a page:  
  
If you have user roles in your backend, you can add user groups in WeWeb that refer to those user roles and restrict access to pages further:

### Images

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

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

## Signup and login workflows ​

When you add the Token Based Auth Plugin to a WeWeb project, new elements will appear in the Add menu under the Plugins section:  
signup forms, login forms, and user profile forms.  
  
You can add an authentication form by drag-and-dropping it on the Canvas.  
You will find a no-code workflow on the Form Container which you can configure to suit your needs.  
In step 1, we make an API request to POST the user's email and password to our auth provider. The URL endpoint and requested information will depend on your auth provider:  
  
In step 2, we store the user token from the result of the previous action:  
  
In step 3, we fetch the user before redirecting the user to another page in step 4.  
In the example above, we are using a Custom Header authentication type but the logic remains the same for signup, login, custom header, auth bearer, and auth basic.  
Based on the information provided by the user in the Form, i.e. the Component Variables in the Form Container, you:  
make an API call to your auth provider with the API request action, store the user token with the Store token action, fetch the user with the Fetch user action, based on the token stored in action 2, and redirect the user to the appropriate page with the Change page action.

### Images

<https://docs.weweb.io/assets/token-auth-add.Ca9b1m_-.png>

<https://docs.weweb.io/assets/token-9-auth-workflow.hX0QMIeU.png>

<https://docs.weweb.io/assets/token-10-user-token.Nw2XhA0G.webp>