WeWeb Server and Server API (DEPRECATED) | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/settings-billing-code-export/weweb-server-api-(deprecated).html> |
| Date de scraping | 2025-04-08T15:56:04.885121+00:00 |

## weweb-server & WeWeb Server API (DEPRECATED) ​

## weweb-server ​

In the past, to self-host a WeWeb app with private pages, you had to host it using the weweb-server.  
On June 26th, 2024, we released an update to change the way we handle private pages. As a result, you no longer need to use the weweb-server to self-host your WeWeb apps , even if those apps use authentication.  
If you're using auth in your app (be it WeWeb auth, Xano, Auth0 or Supabase) and have been using WeWeb for a while, you might already be hosting the WeWeb auth server, which checks for authentication on the backend.  
Here's a schema of the previous self-hosting architecture on a WeWeb app:  
  
If you wish to continue self-hosting this way for now, please watch this 5 min video showing you how to self-host with the weweb-server:  
The WeWeb auth server is available here

### Images

<https://docs.weweb.io/assets/self-hosting-schema.BhmEmMc0.jpeg>

## WeWeb Server API ​

For now, WeWeb continues to provide an API to interact with your weweb-server.

## Principle ​

weweb-server will be the entry point of you self-hosted WeWeb project : it will manage Authentication, pages accesses and file serve for one or multiple of your WeWeb projects.  
weweb-server needs to be connected to a Postgres Database that you will also have to host. We advise you to add a CDN in front of this server to keep some files in cache.  
weweb-server needs some storage to retreive your projects' files. This storage can be local, distant or an AWS S3 bucket.  
Each time you publish a new version of one of your projects, you will need to update the project's files and configure the server's DB info using WeWeb's Dashboard (The configure button can be found in the project's Dashboard > Self-Hosting > Configure WeWeb Server)

## Database ​

You need to create a new Postgres Database that will be used by this server to store some non sensible data about your projects' pages (private accesses, user groups and a list of all pages). You will need to set some environment variables in weweb-server for it to be able to connect to the Database.

## Storage ​

You will need some storage to put your projects' files on it.  
This storage can be :  
Local to weweb-server (add a folder in the server) Distant (in some http url) AWS S3 Bucket (the bucket doesn't need to be dedicated to this storage)  
The storage location is configured using environment variables.

## Environment variables ​

## Main variables : ​

## FILES\_PATH : ​

For local storage, FILES\_PATH should start with ./ else distant storage will be used. You can use two variables in the path that can be found in the name of the ZIP of the project :  
:projectId :filesVersion  
Ex: ./projects/:projectId/:filesVersion will fetch the frontend project's files localy. FILES\_PATH should end without a / .

## SERVER\_PATH : ​

If you setup something like https://my-domain.com/servers/weweb-server/ as WeWeb Server URL in your Dashboard, you should set SERVER\_PATH to the value /servers/weweb-server . SERVER\_PATH should end without a / .

## Database configuration : ​

## AWS S3 configuration (not required if a different storage is used) : ​

## Publish a new version of a project ​

Publish your project on WeWeb Dashboard and wait for it to finish Go to your project's Dashboard > Self-Hosting and download the project's ZIP archive Unzip the archive in your storage. We advise you to use subfolders for each projects and each versions of your projects to prevent downtime. Go to your project's Dashboard > Self-Hosting and click on Configure WeWeb Server  
The new version of your project is now live !