How to use WeWeb Public API | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/settings-billing-code-export/code-export-api.html> |
| Date de scraping | 2025-04-08T15:55:53.870501+00:00 |

## WeWeb Public API ​

The WeWeb Public API enables customers to automate and streamline their deployment processes for WeWeb projects.  
This API is designed for organizations managing multiple WeWeb applications across various domains or server instances.

## Key benefits ​

Automated deployments : integrate WeWeb projects into your CI/CD pipelines. Version control : programmatically manage different versions of your WeWeb applications. Multi-environment support : easily deploy to staging and production environments. Scalability : efficiently manage deployments across multiple domains or server instances. Customized workflows : tailor the deployment process to fit your organization's specific needs.  
The WeWeb Public API offers significant productivity gains for teams handling complex deployment scenarios.  
Instead of manually downloading the application code and/or then manually deploying it to a web server running on your infrastructure, you can trigger custom deployment scripts from within the WeWeb Editor.  
This documentation provides detailed information on authentication, available endpoints, and typical usage patterns to help you integrate WeWeb deployments into your existing systems and workflows.

## Access to WeWeb Public API ​

Please contact the WeWeb team to get access to WeWeb Public API.

## Authentication ​

WeWeb Public API authentication is done using an Authorization header in every requests.  
The value of this header should be Bearer <YOUR WORKSPACE PRIVATE KEY> .  
You workspace Private Key can be found under the Settings tab in your workspace.  
You can generate a new Private Key at any time but this will invalidate the old one.

## Available requests ​

The WeWeb Public API allows you to make the following requests:

### Code

Langage: unknown

{  
 "message": "Fetching data", //Progress message  
 "status": "deploying", //Status of the deployment. Can be : deploying / deployed / failed  
 "version": 33, //Version of current publish  
 "createdAt": "2022-12-12T16:13:47.142Z" //Date of creation  
}

Langage: unknown

{  
 "message": "Fetching data", //Progress message  
 "status": "deploying", //Status of the deployment. Can be : deploying / deployed / failed  
 "environment": "production", //Target environment  
 "version": 33, //Version of current publish  
 "logs": "deploy logs", //Full logs of the deployment  
 "rawZip": true, //rawZip option entered on deploy start  
 "builtZip": true, //builtZip option entered on deploy start  
 "githubEnabled": false, //githubEnabled option entered on deploy start  
 "createdAt": "2022-12-12T16:13:47.142Z" //Date of creation  
}

Langage: unknown

{  
 "success": true, //Success, can be `true` or `false`  
}

## Monitoring a publication ​

Check the status field in the response to determine when the publication is complete. The value will be deployed when the publication is finished.

## Typical auto deploy ​

A typical auto deploy script should be:

### Code

Langage: unknown

1. Start the publication of a project with "builtZip" set to true.  
 2. Store the version provided by the previous request  
 3. While the project is publishing, check the publication status of a project. The publication is done when the progress is 100 and status is "published".  
 6. Download project files ZIP by version using the version provided in the start publication step  
 7. Save the downloaded ZIP file  
 8. Extract the download zip file to your weweb-server storage at the right place (defined by FILES\_PATH)  
 9. Activate the published version in you weweb-server using the version provided in the start publication step