Debugging workflows | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/workflows/debugging.html> |
| Date de scraping | 2025-04-08T15:57:12.267917+00:00 |

## Debugging workflows ​

Debugging is an inherent part of app building.  
When you start working with workflows, there's a good chance you'll run into errors at times.  
In this article, we'll show you how to find out why a Workflow is failing, review common errors and go through a few real-life examples.

## The Workflow Debugger ​

If you're experiencing issues with a Workflow, the easiest way to find out why it's not working is to test the entire Workflow.  
In the example below, we see that the issue comes from the first action – which is highlighted in red – and if we look into the API's response , we can see the issue is that the user tried to login with invalid credentials:  
  
If you're working with a more complex Workflow, you can test an individual Action instead of the entire Workflow.  
In the example below, we test the first action and see the same API response informing us that the user tried to login with invalid credentials:

### Images

<https://docs.weweb.io/assets/debug-workflows1.DYIS00km.png>

<https://docs.weweb.io/assets/debug-workflows2.DW3dDblh.png>

## Status Codes ​

When you test your Workflow, the Debugger will display a status code for every API request you make.  
It would be impossible to go through all the status codes you may encounter but let's review the ones you'll encounter most often.

## 200 Success ​

When your API request was processed as expected, you'll get a 200 success message.  
For example, when you successfully created a new record in your database:

### Images

<https://docs.weweb.io/assets/debug-workflows4.BDJWmVd1.png>

## 401 Unauthorized ​

When the server will not process your request because it lacks valid authentication credentials.  
For example, when you try to create a new record in a table that's protected by an auth system without providing valid auth credentials:  
  
The solution: check the credentials you are using to authenticate and try again.

### Images

<https://docs.weweb.io/assets/debug-workflows3.mrqDKQwy.png>

## 403 Forbidden ​

When the server understood the request but refuses to fulfill it.  
For example, when a normal user tries to perform an action that is reserved to users with an admin role.  
The solution: make sure you are testing the Workflow with a user who has the right to perform the action.

## 422 Invalid value ​

When the server doesn't like the data type you're sending  
In the example below, you can see the Mileage value we send on the left is a string ("7864") and returns an error whereas the value we send on the right is an integer (506) and returns a success message:  
  
The solution: make sure you are sending the data in a format that is expected by your database.  
In the example above, we mistakenly used a Short answer input on our first tried and finally got a success message when we changed the input type to Number .

### Images

<https://docs.weweb.io/assets/debug-workflows5.COFHcI4J.png>

## 429 Too many requests ​

Something that can happen often when you are using third-party APIs is that you can run into API rate limits.  
When that happens, you will see a 429 status error message.  
This has nothing to do with a mistake you made but everything to do with rate limits imposed by the API you're using.  
When that happens, you might want to read your API provider's documentation to check if there's something you can optimize on your side or if you can upgrade your plan with them to set higher limits.

## Network error ​

Some REST APIs, like the Rick & Morty API or Google Maps, accept client-side requests, i.e. requests from your users browsers.  
But many REST APIs, like Twitter, only accept server-side requests.  
By default, being a frontend builder, WeWeb's REST API plugin makes client-side requests.  
In the example below, you can see Twitter returns a Network error when we use the default configuration of a REST API request:  
  
This is known as a CORS issue . You can fix it by enabling the Proxy the request to bypass CORS issues option:

### Images

<https://docs.weweb.io/assets/logs-cors.B9pNRusX.png>

<https://docs.weweb.io/assets/logs-proxy.CyfnUBPH.png>

## Debugging Checklist ​

Ok, now that we've gone through a few examples of errors you can run into when working with workflows (and/or making API calls), let's recap the questions that will help you uncover mistakes and fix errors.  
Am I hitting API rate limits? Did I bind the correct variable to each key? Do my form input types match the field types in my database? Does the API request I am making require authentication headers? Does the API I am working with accept client-side requests or do I need to enable the server-side option? Do I have the correct trigger on the correct element? (e.g. "on submit" if I'm using a "Form Container") Did I use the correct API endpoint URL? Did I use the correct API request method? (e.g. POST, GET, DELETE) Do the names of my keys in WeWeb match the names of my keys in my backend? (bearing in mind these are case sensitive)