Dynamic collection pages | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/pages/dynamic-collection-page.html> |
| Date de scraping | 2025-04-08T15:55:02.736487+00:00 |

## Dynamic collection pages ​

Overview  
Display data from a specific item on a dedicated page.  
Example: We have a list of characters of Rick and Morty and a dynamic page to show character details of a specific character.

## Two Options for Creating a Dynamic Collection Page ​

Bind to variable data Bind to collection data  
In WeWeb, binding consists of displaying external data in your app. This is often parsed JSON coming from 3rd-party APIs or tools like Airtable or Xano.  
When bound, the data is displayed in the app. It can be displayed as text, as an image, as a link, as a list, etc. Every time the data is updated, the app is updated too.  
For more details check out the binding data article.  
Option 1: Binding data  
This method involves binding the page content directly to a variable. When a user accesses the page, the content is retrieved from this variable. Instead of making an API call to our back end and waiting for its response, we simply update a local variable and display that data on our dynamic page. For example, instead of making an API call to fetch data, you might set a variable with data received from another page. This can be quick and efficient but has limitations. Downside: If a user tries to access that URL directly to the page without going through the initial step that updates the variable, they will land on a page with no information because the variable will be empty.  
Option 2: Bind to collection data  
In this approach, the page content is bound to data retrieved from a collection. When a user accesses the page, the data is fetched from the collection, ensuring the content is up-to-date and displayed. This method is more reliable, especially for scenarios where users might access the page directly without going through a specific sequence.  
Choosing the Right Approach: Dynamic Collection Pages offer flexibility in displaying content based on various criteria. In many use cases, you will want to make an API call to your backend to get the information from the specific item you want to display on the dynamic collection page.  
What you’ll need  
2 Pages  
A template page to display dynamic data An initial page with items (for this example, we will use a page with a table of data)  
Plugins  
Install Rest API (a data source to display data)

## Bind to variable data ​

Step 1: Create a dynamic collection page  
Create a template page of the information you want to display.  
Here’s a basic example of a page for the character details. The page has sections for texts, a picture, and a back button to return to my initial page of items.  
After you create the dynamic collection page, navigate to the initial page with the items you want to have on the dynamic collection page.  
For this example, we’re starting with the Rick and Morty Database page with all the show's characters.  
Step 2: Add a Variable Workflow  
Step 3: Bind the Information on the Dynamic Collection Page  
Go to the template page and bind the information found in the variable.  
For our example, we’ll bind the following information: Name of character, picture of the character, species of character, and location.  
Congratulations! 🥳 You just created a dynamic collection page using the bind to variable data!

## Bind to collection data ​

Step 1: Create a dynamic collection page  
Create a template page of the information you want to display.  
Here’s a basic example of a page for the character details. The page has sections for texts, a picture, and a back button to return to my initial page of items.  
After you create the dynamic collection page, navigate to the initial page with the items you want to have on the dynamic collection page.  
For this example, we’re starting with the Rick and Morty Database page with all the show's characters.  
Step 2: Create a New Collection  
To retrieve a specific character instead of the entire collection, create and configure a new collection as an API endpoint for a GET call.  
Step 3: Bind Data on the Dynamic Collection Page  
Step 4: Add a variable to the URL path on our template page  
Step 5: Add the parameter to the API call  
Step 6: Link to collection page with dynamic id  
🥳Congratulations! You just created a dynamic collection page using the bind to collection data!