

Rick&Morty API

Project organization

Basic Project organization:

1. Read the pill and it's requirement
2. Make a list with all task to do
3. Organized the way of how and when do each task

PILL INFORMATION

First. List of the pill requirements documentation:

Pill objectives

- Design a website for the famous television series Rick and Morty
- Information to display on the page must be consumed by a third-party API
- Make a documentation:
 - The requirements documentation
 - Make a record of issues that were detected during the project's execution
 - Make a documentation of how to use the API
 - Make a record of lessons learned

Pill requirements

- You must create a Postman collection with all the requests that you need for this project
- You must use Axios to make several request in parallel
- All the code, including comments, needs to be written in English
- Delete unused files
- You can use only one HTML file
- The app should be designed using a Mobile First strategy so that it is responsive and users can interact with it in any device size.
- Organize CSS files in a clear and orderly manner
- You must use Git, Github and features branches
- You must follow the Git commit best practices in the following guidelines:
- <https://chris.beams.io/posts/git-commit/>
- Every input elements and buttons (except checkboxes that should have a size of 20x20px with padding or margin around them) should have a minimum height or height of 44px such that users can interact with them on mobile devices
- You must use semantic HTML5 elements for all the contents of the Application

LIST OF TASK

Second. Read the pill and it's requirements to make a list with all the task to do.

- → **investigated about the information we want to get and how to display it**

List of task

- Choose the web apparence we want to implement

- Develop the HTML y CSS of this web:
 - Web structure requirements:
 - Header
 - Sidebar
 - Button
 - Main section where is going to be showed the information
 - List of the episodes (letter black and Orange if hover), button to load more
 - Web Funcionality:
 - Show episode list in the sidebar
 - Load more episodes when click the button
 - Scroll
 - Display information in the main container when user click
 - Name
 - Airdate
 - Episode code
 - For each character:
 - Name
 - Status
 - Specie
 - image
 - Work with the API and its response → postman → make a collection with all the requests that we are going to need for this Project → make a list with all the information needed and it request.

API INFORMATION

Base url: <https://rickandmortyapi.com/api>

The base url contains information about all available API's resources.

There are currently three available resources:

- Character: used to get all the characters.
- Location: used to get all the locations.
- Episode: used to get all the episodes.

Character

- Get all characters
- Get a single carácter
- Get varius singles carácter
- Get a carácter filtering by name,

Location:

- Get all locations
- Get a single location -> /[{id}]
- Get multiple locations
- Get locations filtering by

Episode

- Get a single episode
- Get multiple episodes
- Get episode filtering by

Information needed vs. Its request:

Episodes	Name	https://rickandmortyapi.com/api/episode/{id}
	Air date	https://rickandmortyapi.com/api/episode/?episode={episodecode}

	Episode code	
	Created date	
List of the Rick & Morty episodes of the first API pagination		https://rickandmortyapi.com/api/episode
Characters	Name	
	Status	
	Species	
	Gender	
Locations		

PROJECT ORGANIZATION

1. Design the basic html and css

Html

- a. Header
- b. Sidebar
- c. Main section → with a section for each type of information
 - i. Welcome section
 - ii. Episode section
 - iii. Character section
 - iv. Location section

Css → start with the display for mobile devices and add mediaQueries to adjust to other types

2. Make the page navigation
3. Make the API request to get the information and display it on the webpage

Learned and conflicts:

Works with loops in request to make more than one (show all the characters)

Async

Deferred, promises, then, etc

Events propagation

Web distribution

Show and hide element with effects