**Rick and Morty Documentation**

Hi! We are excited your interest in our web page and hope this document been useful and enjoyable! Have a nice day!

**What is Rick and Morty web page?**

It is a web page that uses the API of **rickandmorty** to show information of their episodes and characters!

This rick and morty web page stands out for:

● Responsive.

● Animated

● Artistic, loyal to the art concept of the show

**Requirements**

**●** Internet connection

● Other browser than IE

● Interest in RIck and Morty show

**API documentation**

As we told you the web page uses and API so here is the link <https://rickandmortyapi.com/api/episode>, it gives you an object with all the episodes into an array with hyperlinks to get its characters, which have hyperlinks too, so this API works with **HATEOAS**

Take in count that this web page uses **jquery** and **axios** libraries, so we use the way response.data instead of just data as we used **axios** for the requests instead of **jquery.**

The first link gives the first page of episodes because uses pagination, so what you must do to get the next one is get **response.data.info.next** and **response.data.results** gives the array of chapters

**The Rick and Morty web page is made up of an HTML, a CSS, a JQuery library, a JS**

**and a folder with images called Assets.**

**INDEX.HTML**

In Index.html it is organized in such a way that in the head part we get the AXIOS and JQUERY library, then the Styles.css page and finally the project title.

We have divided the page in 2 sections the header with the title and a button to show the chapters if the screen is very small like a phone and a div which contains 3 sections 2 elements at the sides of the screen and one in the center, the left side contains the chapters and the other one just an image while the centered one would contain the characters of the chapter selected, characters whose origin is in some location, chapters where the character appear, pending on the element clicked.

**STYLES.CSS**

The style sheet is structured firstly with the classes that give a lot of style and like define objects like **header\_\_title**, then the ones that add specific style **absolute--center**, as you can see the way the classes are named is based on the type of class and finally we have the media query where the classes change its rules pending on the screen width.

**JQUERY.JS**

Jquery library downloaded at jquery.com

**SCRIPT.JS**

It begins appending an image to the html so while charging the page, the user would see a gif in the middle of the page 2

It gives the click event to the button that only is displayed when the page is with a short width, so it shows the nav of chapters

Then are declared the variables who stores objects which represent a episode, containing its date, characters and name like S01E03

Then execute a request to get the first page, ‘cause the API uses pagination and store the episodes representation in one of the variables after that it executes a function that do the same for the second page

Then the function to do the request of the second page and so on is created

After that begins what is going to work after the page is ready so when it is ready the gif of charging is removed and a button to show chapters created and appended to the side left nav with a click event to disappear and show instead the chapters of the first page in the left side nav and a button to show add to this nav the chapters of the second page

Then the function that append the element returned by a function that create the episode element and uses animation and setTimeout so the episodes are shown in line from the bottom of the page to be all of them in the side left nav with a scroll to see the ones above the size of the height of the side nav.

Then is the function which create the episodes elements, it gets the text to be shown and and array of the characters that appear in there to store them in an attribute of the element and a click event so cards of the characters that appear in that chapter are showed in the main element

Then the function that contains the animation so it just get as parameter the element that is going to be animated

Then the function that we uses as click event for the button to show the next page of chapters

Then the function that append the characters cards to the main element, so it make a request per character in the attribute of the clicked episode and put the data as an argument to the function that returned a card to show that data.

In the next function ,the one that create a card with the parameters it has for a character card, it’s placed some data as attribute, as the episodes where it appears, origin, gender, while other stuff are shown, and get a click event which call the next function when the card is clicked.

The next function change the style of the card clicked making it bigger with more data and an click event in a link on it to show the origin of this character, and replace the other cards per episodes cards where the character clicked appeared this calling a function that return as other functions a card created by the argument (data from an axios request)

Then this function that returns a html card for the episode with some data and the array of characters that appear in the chapter as an attribute with the same click event as click one chapter of the side nav.

Finally we have a function to show the location of the character clicked, so when the link of the character card is clicked the cards of the main disappear and cards of the characters with that origin shown reusing the function that return a card with the data of a character.