



pecode.



React.js test task

Rick and Morty test task

Before you read the description please take a look at the APIs that you will work with:

<https://rickandmortyapi.com/documentation/#introduction>

Task Description

The goal of the task is to create a react web application that will display all the information about the Rick and Morty Tv series. It Should have 4 tabs in the navigation panel: Characters, Episodes, Locations, My watch list.

1. The Characters page should have a list of all characters that are in the tv series (Please implement pagination for that. So that there are 20-character cards per page). The list should have a filter by “species”, “status”, “gender” fields. The character card should have a character picture (if any), and some additional info about that character (your choice). When you click on the character card – the popup should be opened with all the character info.
2. The episode page should be really similar to the character page, the episodes should have been filtered by “name” field (if you will have inspiration please add other filters). As there is no image in episodes, please make this not a list but a table, also there should be 20 episodes per page(pagination).
3. The Locations page should also be a table with pagination and should have filtered by “name”, “type”, “dimension” field.
4. My watch list should have a simple to do function. There should be an input field where user can write down a new episode to watch later. There are no APIs for this subtask so please use local storage to store the To Do items. The toDo list item should have a remove button to remove it from the list and the checkbox to mark it as complete(when the page has reloaded the status of the task and the list should not change).

Technical description

Here is the list of APIs that you will need - <https://rickandmortyapi.com/documentation/#introduction>

For this task please use the following:

- React.js
- React-redux/context (optional)
- Typescript (optional)
- Material UI - <https://material-ui.com/> for all popups, buttons, etc. (Or use any other UI library, optional)
- Scss/CSS

There is no design for this task, So it is also up to you, but the design should look good and user friendly.

For API calls use fetch.

Please create as many decomposed components as possible. And try to reuse them as much as possible.

The code architecture should be logically correct. There should be a separate folder for each page, small components should be separated from the big ones. For shared components create a folder UI and store them there (The components from the episodes page should only be used in the episodes folder).

Good luck!

