

Rick and Morty application

Instructions

Application Overview

The primary idea of the application is to provide a simple and user-friendly interface for interacting with the open API at <https://rickandmortyapi.com>.

The application consists of three main pages:

1. **Character List Page:** This page displays a list of all characters from the Rick and Morty series.
2. **Character Details Page:** Upon selecting a character, this page presents more detailed information about the chosen character.
3. **Settings Page:** The page where you can go to the theme changing section

Core Requirements

Below is a more detailed description of the required features and functionalities for each page of the application:

- 1) Characters list screen
 - Display an infinite scrolling list.
 - Data should be loaded in pages (pagination mechanism).
 - Data loading should be seamless for the user (ensure there are no delays or freezes).
 - Add a content filter by status: "Alive" | "Dead" | "unknown", **and species:** "Alien" | "Human". There should be a dropdown at the top of the screen for filter selection, with the character list scrolling beneath it.
 - When a user clicks on a card, they should be taken to a screen with detailed information about the character.
- 2) Character detailed information
 - The page should have a header with a back button and the character's name.
 - This page should display all detailed information about the character available from the API.
 - This page should include an image of the character.

Common Requirements

- 1) Use your own design for the application
- 2) Please use state management (of your choice, but Redux is preferred).
- 3) Bottom navigation should include 2 pages "Main" (All Characters) and "Settings" .
Using react native navigatuin
- 4) A custom Loader must be present in the application, when loading a new chunk of data
- 5) The app should support dark and light theme. You should add another screen with "Settings" where user can change the theme color and save it to LocalStorage of the mobile device. Every time the user closes the app, the theme should be saved and be the same as before the session was terminated
- 6) Make sure you consider and handle all possible states of the app (offline, data not loading, errors, etc).
- 7) Make offline mode with display minimum last 10-20 characters (save it on your locally db, choose any preferable solution)
- 8) Make a check for internet availability and in case of its absence display a screen with:
 "text": "Please check your internet connection and try again",
 "button": "Try again"

Result:




- Code should be pushed to a public repository on GitHub
- Repository should contains commit history instead of a single one
- Select icons by yourself. App should contains non-default launch icon
- Link should be provided to interviewer

Additional Notes:

While the core requirements outlined above are your primary focus, we encourage you to implement best practices throughout the project. Considerations for performance optimization, accessibility, responsive design, as well as code organization and quality, will be highly valued. We are particularly interested in seeing how you tackle these challenges and incorporate them into your solution.

Please ensure your code is well-structured, with comments where necessary, and keep in mind the scalability and maintainability of the application.

Example of a possible front page design

	<p>Alien Rick</p> <ul style="list-style-type: none">● Unknown - Alien <p>Last known location: Citadel of Ricks</p> <p>First seen in: Close Rick-counters of the Rick Kind</p>
	<p>Randy Dicknose</p> <ul style="list-style-type: none">● Alive - Human <p>Last known location: Interdimensional Cable</p> <p>First seen in: Interdimensional Cable 2: Tempting Fate</p>
	<p>Sperm Queen</p> <ul style="list-style-type: none">● Dead - Unknown <p>Last known location: Earth (Replacement Dimension)</p> <p>First seen in: Rickdependence Spray</p>