

Star Wars character app

This assessment is designed to test your abilities in UI implementation, API integration, error handling, code quality, testing, and application development using ReactJS and TypeScript.

Project Overview:

Create a web application that utilizes the public Star Wars API (SWAPI) (<https://swapi.dev/>) to display a list of Star Wars characters with the following features:

UI and API Integration:

- Fetch and display a list of Star Wars characters using the /people endpoint with pagination as infinite scroll.
- Implement a loader for fetching/refetching data (e.g. refresh) and handle error states gracefully (e.g. network errors).
- Display character cards with names, random images, colored based on species, and animations on hover.
- Clicking on a character's card opens a modal with detailed information about the character.
- Modal displays name, height, mass, date added, number of films, and birth year.
- Fetch and display information about the character's homeworld (name, terrain, climate, residents).

Search and Filtering:

- Implement a character search functionality by name, partial or complete.
- Allow filtering by homeworld, film, or species.
- Combine search and filters for enhanced user experience.

Authentication and Security:

- Implement JWT authentication with silent refresh for token expiration.
- Create a UI for logging in and out (mock authentication as the API doesn't require it).
- Ensure robust error handling and validation for security and user-friendliness.

Testing and Quality:

- Write integration tests for modal behavior and character information.
- Follow React and TypeScript best coding practices for improved code readability.
- Ensure code quality, error-free execution, and adhere to strict TypeScript typing.

Scalability and Compatibility:

- Design the app to be responsive and compatible with desktops, tablets, and mobile phones.

Additional Requirements:

- Utilize a public image library for character images.
- Avoid vulnerabilities in third-party packages.
- Implement Docker definitions for easy deployment using Docker Compose.
- Include a comprehensive README with instructions for running the project.
- Prioritize React and TypeScript best practices for optimal development.
- Push the changes to GitHub

UI Implementation (Suggestion)

The UI of the Star Wars Character App should be visually appealing, user-friendly, and intuitive. Here's an explanation of how the UI should work:

1. Character List and Pagination:

The main page displays a list of character cards, each showing the character's name and a random image.

Implement pagination at the bottom of the list to allow users to navigate through multiple pages of characters.

2. Loader and Error Handling:

When fetching data, display a loader to indicate that the app is fetching information.

Handle potential API errors gracefully by showing an error message and providing a way to retry fetching.

3. Character Card Animation:

Apply a subtle animation (such as a scale or color change) to character cards when users hover over them, providing a visually engaging experience.

4. Character Modal:

Clicking on a character card opens a modal displaying more detailed information about the character.

The modal header shows the character's name.

Display height in meters, mass in kilograms, birth year, and the number of films the character appears in.

Include a button to fetch additional details about the character's homeworld.

5. Homeworld Information:

Once the user clicks the "Fetch Homeworld Details" button, display information about the character's homeworld.

Show the homeworld's name, terrain, climate, and the number of residents.

6. Search and Filter:

Implement a search bar at the top of the page for searching characters by name (partial or complete).

Provide a dropdown or selection options for filtering characters by homeworld, film, or species.

Combining search and filters should update the character list dynamically.

7. Authentication UI:

Create a login/logout section at the top of the page.

Include fields for entering a username and password (mock authentication).

Display a success message upon successful login and a logout button.

8. Responsive Design:

Ensure the app's responsiveness by adapting the layout and design for various screen sizes, including desktops, tablets, and mobile phones.

Optimize the display of character cards, modals, and other UI elements for each device.

By following these UI design and functionality guidelines, you'll create an engaging and user-friendly experience for exploring Star Wars characters. Remember to pay attention to details, animations, and responsive design to provide a seamless and enjoyable browsing experience for users across different devices.