

Descrição dos Arquivos

fetchCharacters.js

Descrição

- **Bibliotecas usadas:**
 - **axios:** para realizar chamadas HTTP (neste caso, para a API de personagens).
 - **fs:** para criar pastas e salvar arquivos no sistema de arquivos.

Verificação de diretório:

- **Verificação de diretório:**
 - A função verifica se o diretório `./public/data/` existe. Se não existir, ele cria a pasta.
 - **Solicitação à API**
 - A função faz uma chamada GET para a API em `http://localhost:3000/api/characters/external`.
 - **Salvando dados:**
 - O arquivo JSON contendo os personagens retornados pela API é salvo em `./public/data/characters.json`.
 - **Tratamento de erros:**
 - Se houver um erro na requisição ou gravação do arquivo, ele é capturado e exibido no console.
-

package.json

Descrição

- Inserido no `start` para setar a porta como 3001, pois a API utiliza a porta 3000, e para rodar `fetchCharacters.js` para buscar os personagens da SWAPI através da API StarWars criada.
-

package-lock.json

Descrição

- **Descrição:** Não houve alterações.
-

starwars-frontend/public/index.html

Descrição

- **DOCTYPE:** Define o documento como um arquivo HTML5.
 - **<meta charset="utf-8">:** Define o conjunto de caracteres para UTF-8.
 - **Meta viewport:** Garante que a página seja responsiva.
 - **Título:** Define o título da página como "Desafio StarWars Zolly".
 - **Favicon:** Um ícone personalizado exibido na aba do navegador.
 - **<noscript>:** Exibe uma mensagem para usuários com o JavaScript desativado.
 - **Div root:** O `<div id="root">` é o ponto de entrada onde o React irá montar a aplicação.
-

starwars-frontend/public/manifest.json

Descrição

- **short_name:** Nome curto exibido quando o espaço é limitado.
 - **name:** Nome completo da aplicação.
 - **icons:** Contém as diferentes versões do ícone.
 - **start_url:** Define a URL de entrada padrão.
 - **display:** Define o modo de exibição da aplicação como standalone.
 - **theme_color:** Cor predominante usada para o tema da aplicação.
 - **background_color:** Cor de fundo que aparece enquanto a aplicação está sendo carregada.
-

starwars-frontend/public/images

Descrição

- Imagens usadas na aplicação.
-

starwars-frontend/public/data/characters.json

Descrição

- Arquivo criado ao iniciar o servidor de frontend que guarda todos os personagens buscados pela API criada para melhorar a performance.
-

starwars-frontend/src/assets/

Descrição

- Imagens e logos usados no frontend.
-

starwars-frontend/src/components/CharacterItem.js

Descrição

- **className:** Define a classe CSS para a estrela de favorito.

- **onClick:** Chama a função `onToggleFavorite` ao clicar na estrela.
 - **role:** Define o `span` como um botão para acessibilidade.
 - **aria-label:** Atualiza o `aria-label` baseado no estado.
 - **style:** Estilo `inline` define o cursor, tamanho da fonte, e a cor da estrela.
-

`starwars-frontend/src/components/CharacterList.js`

Descrição

- **Verificação:** Exibe uma mensagem "Nenhum personagem encontrado" se não houver personagens na lista.
 - **Renderização:** Cada `CharacterItem` recebe suas props para a renderização correta.
-

`starwars-frontend/src/components/SearchBar.js`

Descrição

- **Imports:** Importa `React`, `useState`, estilos `CSS` e o logo `SVG`.
 - **Componente `SearchBar`:** Recebe três props: `onSearch`, `onFavoriteSearch` e `resetCharacters`.
 - **Estado `search`:** Utiliza `useState` para armazenar o termo de busca.
 - **Função `handleSearch`:** Chama `onSearch` com o termo atual do estado `search`.
 - **Renderização:** Renderiza um contêiner com o logo, campo de entrada e botões.
-

`starwars-frontend/src/services/api.js`

Descrição

- **Imports:** Importa a biblioteca `Axios`.
 - **Instância do `Axios`:** Cria uma instância chamada `api` para configurar chamadas à API.
 - **Exportação:** A instância é exportada para ser usada em outros arquivos.
-

`starwars-frontend/src/styles.css`

Descrição

- **Estilos:** Organizados em seções com comentários explicativos para cada classe e estilo.
 - **Componentes:** Contém estilos para a barra de pesquisa, tabela de personagens, navegação e botões.
-

`starwars-frontend/src/reportWebVitals.js`

Descrição

- Não foi alterado.
-

`starwars-frontend/src/index.js`

Descrição

- Não foi alterado.
-

starwars-frontend/src/App.js

Resumo das Funções e Ferramentas Utilizadas:

- **React:** Biblioteca para construir interfaces de usuário.
- **useState:** Hook para gerenciar o estado local no componente.
- **useEffect:** Hook para realizar efeitos colaterais.
- **axios:** Para requisições HTTP.
- **fetchCharacters:** Carrega personagens do JSON local e filtra com base no termo de busca.
- **fetchFavorites:** Busca personagens favoritos da API.
- **handleSearch:** Chamado quando o usuário realiza uma busca.
- **handleFavoriteSearch:** Busca personagens favoritos da API.
- **toggleFavorite:** Alterna entre marcar e desmarcar um personagem como favorito.
- **goToMainPage:** Recarrega todos os personagens.
- **Paginação:** Funções para navegar entre páginas.