

**Dokumentacja techniczna**

Spis treści

[1. Środowisko uruchomieniowe i konfiguracja 3](#_Toc18255877)

[2. Struktura aplikacji 3](#_Toc18255878)

[3. Komponenty aplikacji 7](#_Toc18255879)

[Komponenty ogólne 7](#_Toc18255880)

[Komponent: Landing Page 7](#_Toc18255881)

[Komponent: Layout 7](#_Toc18255882)

[Komponent: Loading 7](#_Toc18255883)

[Komponent: LoaderPanel 8](#_Toc18255884)

[Komponent: withErrorHandling 8](#_Toc18255885)

[Komponenty sekcji głównej (wyszukiwanie i przeglądanie informacji o serialu) 8](#_Toc18255886)

[Komponent: Shows 8](#_Toc18255887)

[Komponent: SearchBox 8](#_Toc18255888)

[Komponent: InputBox 8](#_Toc18255889)

[Komponent: ResultList 9](#_Toc18255890)

[Komponent: Pagination 9](#_Toc18255891)

[Komponent: DetailsBox 9](#_Toc18255892)

[Komponent: DetailsHeader 9](#_Toc18255893)

[Komponent: DetailsMeta 9](#_Toc18255894)

[Komponent: SeasonsList 10](#_Toc18255895)

[Komponent: MediaBox 10](#_Toc18255896)

[Komponent: RecommendationBox 10](#_Toc18255897)

[Komponent: FavoritesBox 10](#_Toc18255898)

[Komponenty osi czasu 11](#_Toc18255899)

[Komponent: Timeline 11](#_Toc18255900)

[Komponenty zarządzania danymi użytkownika 11](#_Toc18255901)

[Komponent: UserAdministration 11](#_Toc18255902)

# Środowisko uruchomieniowe i konfiguracja

Aplikacja **showsdb** składa się z 2 modułów:

* Moduł prezentacyjny – został napisany z wykorzystaniem frameworka React, wraz z React-router, a o za tym idzie serwer musi zostać skonfigurowany zgodnie z wymaganiami (przekierowanie wszystkich zapytań na plik index.html).
* Baza danych Firebase – baza służy do zapisywania danych o użytkownikach, jak również do przechowywania listy ulubionych seriali. Adres bazy jest zdefiniowany w pliku **helpers/axios-firebase.js**. Dodatkowo należy zdefiniować klucz API aplikacji (plik **helpers/APIKey\_firebase.js**) aby poprawnie działały funkcji autoryzacji.

Aplikacja korzysta z API serwisu TMDB – aby jej to umożliwić należy podać poprawny klucz API – skonfigurowany w pliku **helpers/APIKey.js**)

# Struktura aplikacji

|  |  |
| --- | --- |
| <root>  | App.css  | App.js  | index.css  | index.js  | serviceWorker.js  | version.info.json  |  +---assets  | \---images  | backdrop.jpg  | cancel.png  | episode\_image.jpg  | logo\_transparent.png  | no\_picture.jpg  | timeline\_empty.png  |  +---components  | +---LandingPage  | | | LandingPage.js  | | | LandingPage.module.css  | | |  | | \---TrendBox  | | TrendBox.js  | | TrendBox.module.css  | |  | +---Layout  | | Layout.js  | | Layout.module.css  | |  | +---UI  | | +---Backdrop  | | | Backdrop.js  | | | Backdrop.module.css  | | |  | | +---Loading  | | | Loading.js  | | | Loading.module.css  | | |  | | +---MainMenu  | | | | MainMenu.js  | | | | MainMenu.module.css  | | | |  | | | \---LoaderPanel  | | | LoaderPanel.js  | | |  | | +---Message  | | | Message.js  | | | Message.module.css  | | |  | | \---Modal  | | Modal.js  | | Modal.module.css  | |  | \---withErrorHandler  | withErrorHandler.js  |  +---containers  | +---Shows  | | | Shows.js  | | | Shows.module.css  | | |  | | +---DetailsBox  | | | | DetailsBox.js  | | | | DetailsBox.module.css  | | | |  | | | +---DetailsHeader  | | | | DetailsHeader.js  | | | | DetailsHeader.module.css  | | | |  | | | +---DetailsMeta  | | | | | DetailsMeta.js  | | | | | DetailsMeta.module.css  | | | | |  | | | | \---MetaInfoList  | | | | | MetaInfoList.js  | | | | | MetaInfoList.module.css  | | | | |  | | | | \---MetaInfo  | | | | MetaInfo.js  | | | | MetaInfo.module.css  | | | |  | | | +---MediaBox  | | | | | MediaBox.module.css  | | | | | MediaBoxMovies.js  | | | | |  | | | | +---MediaPlayer  | | | | | MoviePlayer.js  | | | | | MoviePlayer.module.css  | | | | |  | | | | \---MediaSummaryMovie  | | | | MediaSummaryMovie.js  | | | |  | | | +---RecommendationBox  | | | | RecommendationBox.js  | | | | RecommendationBox.module.css  | | | |  | | | \---SeasonsList  | | | | SeasonsList.js  | | | | SeasonsList.module.css  | | | |  | | | +---EpisodesList  | | | | | EpisodesList.js  | | | | | EpisodesList.module.css  | | | | |  | | | | \---Episode  | | | | Episode.js  | | | | Episode.module.css  | | | |  | | | \---SeasonsPagination  | | | SeasonsPagination.js  | | | SeasonsPagination.module.css  | | |  | | +---FavoritesBox  | | | | Favorite.class.js  | | | | FavoritesBox.js  | | | | FavoritesBox.module.css  | | | |  | | | \---FavoriteItem  | | | FavoriteItem.js  | | | FavoriteItem.module.css  | | |  | | \---SearchBox  | | | SearchBox.js  | | | SearchBox.module.css  | | |  | | +---InputBox  | | | InputBox.js  | | | InputBox.module.css  | | |  | | +---Pagination  | | | Pagination.js  | | | Pagination.module.css  | | |  | | \---ResultList  | | | ResultList.js  | | | ResultList.module.css  | | |  | | \---ResultItem  | | ResultItem.js  | | ResultItem.module.css  | | ResultItemNoResults.js  | |  | +---Timieline  | | | Timeline.class.component.js  | | | Timeline.class.js  | | | Timeline.js  | | | Timeline.module.css  | | | Timeline.test.js  | | |  | | \---TimelineBuilder  | | | TimelineBuilder.js  | | |  | | +---Event  | | | | Event.js  | | | | EventDetailed.js  | | | |  | | | +---ShowDetailed  | | | | ShowDetailed.js  | | | | ShowDetailed.module.css  | | | |  | | | \---ShowSummary  | | | ShowSummary.js  | | | ShowSummary.module.css  | | |  | | \---HorizontalRule  | | HorizontalRule.js  | |  | \---UserAdministration  | UserAdministration.js  | UserAdministration.module.css  |  +---helpers  | APIKey.js  | APIKey\_firebase.js  | authMessages.js  | axios-external.js  | axios-firebase.js  | format-date.js  | genre-mapper.js  | limitWords.js  | mappers.js  | proper-digit-number.js  |  \---store  | authReducer.js  | reducer.js  |  \---actions  actionTypes.js  auth.js  configuration.js  currentShow.js  favorites.js  index.js | Główny folder aplikacji  App.js zawiera główną deklarację routingów oraz reduxa z globalną konfiguracją dla całej aplikacji (związanej z serwisem TMDB)  Katalog assets zawiera wszystkie statyczne obrazki wykorzystywane w projekcie.  Katalog zawierający komponenty związane z interfejsem użytkownika, panelami logowania, tłami. Komponenty te z natury są wielokrotnie wykorzystywane w projekcie. Mogą, choć nie muszą być bezstanowe. Tam gdzie jest to zasadne (szczególnie komponenty logowania) wykorzystują React-redux aby dzielić stan z aplikacją.  Katalog zawiera komponenty, które grupują pozostałe komponenty; występują najczęściej tylko raz w aplikacji i stanowią złożony system zależności.  Wsyępuje to podobny podział do logicznego podziału aplikacji – Shows (podstrona Home, z wyszukiwaniem, przeglądaniem szczegółów i panelem ulubionych seriali), Timeline (podstrona Timeline z osią czasu ulubionych seriali) oraz UserAdministartion (podstrona do zarządzania kontem użytkownika)  Katalog zawiera pliki i skrypty pomocnicze, które są wykorzystywane na wielu płaszczyznach aplikacji – konfiguracje połączeń asynchronicznych axios, adresy bazy danych, klucze API, a także skrypty do mapowania obiektów JSON zwracanych z zewnętrznych serwisów.  Katalog, w którym znajdują się reducery React-redux – w podziale na dot. Autoryzacji użytkownika i zarządzania ulubionymi serialami.  Katalog actions zawiera ActionCreatory dla React-redux, które dotyczą konfiguracji, autoryzacji, obsługi ulubionych oraz aktualnie przeglądanych seriali. |

# Komponenty aplikacji

Aplikacja wykorzystuje React-redux do zarządzania globalnym stanem (również asynchronicznie pobieranym z zewnętrznych źródeł za pomocą store). Tam gdzie dane zewnętrzne nie muszą być dostępne dla innych komponentów, ich zarządzanie jest pozostawione w konkretnych komponentach i ewentualnie przekazywane komponentom-dzieciom za pomocą właściwości (props).

Poniżej przedstawiono najważniejsze komponenty w aplikacji:

## Komponenty ogólne

|  |  |  |
| --- | --- | --- |
| Komponent: Landing Page | | |
| **Opis komponentu** | | |
| Główny komponent aplikacji, który zostaje wyświetlony jako „strona startowa”. Kiedy użytkownik nie wyświetla informacji o konkretnym serialu, właśnie ten komponent zostaje wyświetlony. Komponent pobiera asynchronicznie informacje o serialach z ostatniego tygodnia. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** – konfiguracja oraz zarządzanie wyświetlanym serialem | **TAK** – proces ładowania danych (asynchroniczny) oraz informacje o serialach z tygodnia | **TrendBox** – komponent bez stanu, wyświetlający listę seriali z tygodnia na podstawie przekazanej listy (props) |

|  |  |  |
| --- | --- | --- |
| Komponent: Layout | | |
| **Opis komponentu** | | |
| Komponent dzieli logiczną strukturę aplikacji na menu i zawartość. Służy do poprawnego podziału HTML. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **NIE** | **NIE** | **brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: Loading | | |
| **Opis komponentu** | | |
| Komponent wyświetla obiekt w interfejsie podczas działań asynchronicznych. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **NIE** | **NIE** | **brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: LoaderPanel | | |
| **Opis komponentu** | | |
| Komponent odpowiada za wyświetlenie, jak również logikę autoryzacji użytkownika. Wersyfikuje poprawność wprowadzonych danych i komunikuje się z zewnętrzną bazą danych. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** – zarządzanie stanem autoryzacji użytkownika | **TAK** – walidacja pól nazwy użytkownika i hasła | **Brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: withErrorHandling | | |
| **Opis komponentu** | | |
| Komponent pomocniczy, który steruje działaniem axios i przechwytuje potencjalne błędy połączenia (brak internetu, timeout itp.). Wykorzystywany tylko w momencie bezpośredniej komunikacji komponentu z asynchronicznym zapytaniem AJAX przez axios. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **NIE** | **NIE** | **brak** |

## Komponenty sekcji głównej (wyszukiwanie i przeglądanie informacji o serialu)

|  |  |  |
| --- | --- | --- |
| Komponent: Shows | | |
| **Opis komponentu** | | |
| Komponent odpowiadający za podstronę „Home” w aplikacji. Grupuje komponenty i zarządza ich stanem. Dodatkowo w przypadku szczegółów wyświetlanego serialu, wywołuje pobieranie danych serialu i zarządza stanem ładowania danych. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** – konfiguracja aplikacji, aktualnie wyświetlany serial, lista ulubionych | **TAK** – ładowanie danych szczegółów serialu | **DetailsBox, FavoriteBox, SearchBox** |

|  |  |  |
| --- | --- | --- |
| Komponent: SearchBox | | |
| **Opis komponentu** | | |
| Komponent grupujący komponenty odpowiadające za pole wyszukiwania serialu, jak również wyświetlenie i przeglądanie wyników. Odpowiada również za tzn. lazy loading – dociąganie wyników wyszukiwania w zależności od dalszych podston przeglądania. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** – konfiguracja aplikacji, aktualnie wyświetlany serial | **TAK** – ładowanie danych, wyniki wyszukiwania, obsługa paginacji | **InputBox, Pagination, ResultList** |

|  |  |  |
| --- | --- | --- |
| Komponent: InputBox | | |
| **Opis komponentu** | | |
| Komponent odpowiadający za wyświetlanie okna wyszukiwania serialu po nazwie. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **NIE** | **NIE** | **brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: ResultList | | |
| **Opis komponentu** | | |
| Komponent wyświetlający listę wyników odpowiadających zadanemu wyrażeniu. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** – konfiguracja aplikacji | **NIE** | **ResultItem** – komponent odpowiadający za pojedynczy wynik wyszukiwania – wywołujący podgląd szczegółow serialu |

|  |  |  |
| --- | --- | --- |
| Komponent: Pagination | | |
| **Opis komponentu** | | |
| Komponent zarządzający wynikami wyszukiwania w zależności od oglądanej podstrony. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **NIE** | **NIE** | **brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: DetailsBox | | |
| **Opis komponentu** | | |
| Komponent grupujący komponenty odpowiadające za wyświetlanie szczegółów serialu, listy odcinków, listy mediów przypisanych do serialu, listy proponowanych seriali. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –aktualnie wyświetlany serial | **NIE** | **DetailsHeader, DetailsMeta, MediaBox, SeasonsList, RecommendationBox** |

|  |  |  |
| --- | --- | --- |
| Komponent: DetailsHeader | | |
| **Opis komponentu** | | |
| Komponent wyświetlający nagłówek aktualnie przeglądanego serialu. Dodatkowo komponent obsługuje zarządzanie dodawaniem/usuwaniem serialu z listy ulubionych. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –aktualnie wyświetlany serial, konfiguracja aplikacji, lista ulubionych | **NIE** | **brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: DetailsMeta | | |
| **Opis komponentu** | | |
| Komponent wyświetlający dane szczegółowe aktualnie przeglądanego serialu. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –aktualnie wyświetlany serial, konfiguracja aplikacji | **NIE** | **MetaInfoList** – komponent z pojedynczą informacją o serialu |

|  |  |  |
| --- | --- | --- |
| Komponent: SeasonsList | | |
| **Opis komponentu** | | |
| Komponent zarządzający wyświetlaniem listy odcinków z danego sezonu, jak również komponentem odpowiedzialnym za poruszanie się po liście sezonów. Dodatkowo spełnia funkcje dodawania serialu do ulubionych, kiedy zaznaczymy obejrzany odcinek jako obejrzany. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **NIE** | **TAK** – obsługa nawigacji listy sezonów | **EpisodesList** – wyświetlenie listy odcinków z sezonu,  **SeasonsPagination –** wyświetlenie nawigacji po sezonach |

|  |  |  |
| --- | --- | --- |
| Komponent: MediaBox | | |
| **Opis komponentu** | | |
| Komponent wyświetlający listę powiązanych mediów (pobieraną asynchronicznie), obsługujący odtwarzacz youtube z powiązanych filmów, jak również paginację wyników. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –aktualnie wyświetlany serial, konfiguracja aplikacji | **TAK** – ładowanie danych, obsługa paginacji, obsługa odtwarzacza youtube | **MediaPlayer –** komponent obsługujący odtwarzacz youtube i przekazany (props) film,  **MediaBoxMovies –** komponent wyświetlający pojedynczy wynik i pobirający odpowiednią miniaturkę z serwera youtube |

|  |  |  |
| --- | --- | --- |
| Komponent: RecommendationBox | | |
| **Opis komponentu** | | |
| Komponent wyświetlający listę powiązanych seriali z aktualnie przeglądanym. Dane komponent pobiera asynchronicznie. Wybór konkretnego serialu powoduje odpowiednią nawigację. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –aktualnie wyświetlany serial, konfiguracja aplikacji | **TAK** – ładowanie danych, obsługa paginacji | **Brak** |

|  |  |  |
| --- | --- | --- |
| Komponent: FavoritesBox | | |
| **Opis komponentu** | | |
| Komponent zarządza listą ulubionych seriali. Wyświetla w odpowiedni sposób listę (oznaczając obejrzane/ nieobejrzane seriale). Komponent również odpowiada za wywołanie zapisu ulubionych do chmury (po zmianie na liście). | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –konfiguracja aplikacji, lista ulubionych, autoryzacja | **NIE** | **FavoriteItem –** komponent wyświetlający pojedynczy kafelek dla serialu z listy ulubionych. Wszystkie dane przekazywane są za pomocą właściwości (props) |

## Komponenty osi czasu

|  |  |  |
| --- | --- | --- |
| Komponent: Timeline | | |
| **Opis komponentu** | | |
| Komponent odpowiada za logikę grupowania listy ulubionych w formę osi czasu (pod względem dat premiery). Komponent pobiera asynchronicznie dane dotyczących odcinków i ich dat premier. Dane są pobierane za każdym razem kiedy lista ulubionych ulegnie zmianie.  Komponent przekazuję stworzoną listę do komponentu-dziecka odpowiedzialnego za wyświetlenie danych. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –konfiguracja aplikacji, lista ulubionych, autoryzacja | **TAK** –ładowanie danych, dane zgrupowane w oś czasu | **TimeLineBuilder –** komponent, który na podstawie przekazanej listy osi czasu (props) wyświetla ją w odpowiedni sposób |

## Komponenty zarządzania danymi użytkownika

|  |  |  |
| --- | --- | --- |
| Komponent: UserAdministration | | |
| **Opis komponentu** | | |
| Komponent odpowiada za zmianę hasła użytkownika. W sposób asynchorniczny sprawdza czy stare, podane hasło jest poprawne, i odpowiednio reaguje – zmienia stare na nowe hasło, bądź wyświetla komunikat o błędzie. | | |
| **Wykorzystanie react-redux** | **Zawiera stan wewnętrzny** | **Istotne komponenty-dzieci** |
| **TAK** –autoryzacja | **TAK** –walidacja hasła (starego, nowego i powtórzonego nowego) | **brak** |

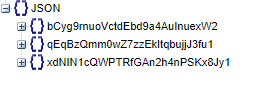
# Komunikacja z bazą danych

Baza danych z której korzysta aplikacja opiera się na architekturze Firebase (Realtime Database). Autoryzacja użytkowników również wykorzystuje wbudowany w Firebase mechanizm autoryzacji (zgodny z API). Do komunikacji z bazą wymagany jest adres bazy oraz klucz API aplikacji (patrz sekcja 1.Środowisko uruchomieniowe i konfiguracja).

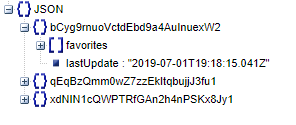
Dane przechowywane w bazie to lista ulubionych seriali dla każdego z użytkowników (identyfikowanych przez unikalne ID nadawane przez bazę).

## Struktura danych

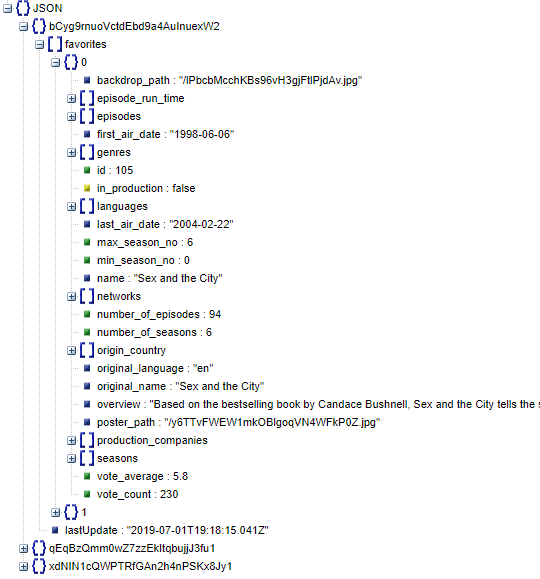
Dane są pogrupowane po ID użytkownika:



Każdy z użytkowników posiada listę ulubionych oraz znacznik czasowy o ostatnim odświeżeniu osi czasu:



Lista ulubionych zawiera obiekty opisujące każdy z seriali:



Warto zwrócić uwagę iż seriale, dla których użytkownik zaznaczył obejrzane odcinki znajdują się w opcjonalnej liście „watched”. Są one identyfikowane przez ID nadawanego przez TMDB:

