**Report on React-Native SilverOctoMoviebase**

Github: <https://github.com/Dannylyhn/SilverOctoMoviedbReactNative>

I use the Expo Navigation library to navigate the switching between different pages. I chose to nest my navigators where my “Home” component contains tab navigators so that I can navigate through important pages using the tab navigators at the bottom and enter different pages through stack navigators.

My HomeScreen.js is a page which contains popular movies. I am using the hook useState() to store and update my json dataset with its function. Using useState() is better than just using a variable but it rerenders the view which hinders the state of the app getting out of sync. I set the data using the hook useEffect () to update the json data on every render. The data displayed on the homescreen is from a json file from themoviedb because the application would sometimes take a long time to get the data from an API fetch. To display the posters, I use the poster\_path from each item which is passed to a uri. The Image is wrapped in a TouchAbleWithoutFeedback to make it possible to press the Image and navigate to the next page using Navigation. I am using a FlatList to render the data that is currently shown on the display in the Popular Movies page instead of rendering all the data resulting in better performance as a moviedb can contain a lot of data. The movies are scrolled through sideways. For this project I could have used ScrollView for showing movies, but it would not scale well because ScrollView loads all content at once resulting in performance issues. I am using the Dimensions component to get height and width of the current device to scale the pictures properly.

Et billede, der indeholder tekst

Automatisk genereret beskrivelseTo navigate to the details of a movie, you can either press on the picture or the title which will navigate you to the next page containing more information about the movie. I pass the movie\_id data to the details\_screen because I am using the movie\_id to fetch specific movie data from TheMovieDB API. I use the same hooks as the HomeScreen.js because I want the data to be rendered and in sync. The image is fetched using the data about poster\_path from the API. I use ScrollView in DetailsScreen.js to make it possible to scroll the page.