

CODING CHALLENGE

TURKCELL TV+

BIGSCREEN DEV TEAM

2023

1. Introduction

Purpose

This coding exercise consists from a single project, which we expect you to use state-of-art Javascript technologies that you prefer to implement. Examples of such technologies we expect to see are (not all of them are required):

- A core library (React, Vue, Angular...) (preferably react + next or pure react)
- An application state management tool (Redux, Vuex...), hooks or micro state management libraries
- A middleware such as Redux-Thunk or Redux-Saga
- EcmaScript 6 (typescript, await-async instead of promises, arrow functions, template strings, object destructuring...)
- Internationalization (i18n) and Routing
- Node.js libraries (express, webpack, babel...)
- Code formatting and linting (Eslint, Jslint...)
- Testing (jest, karma, mocha...)
- A CSS extension (sass...)

Assumptions and Constraints

Please deliver your solutions within a week.

Please deliver your solutions without “node_modules” folder.

Please deliver your solutions by compressing them into a zip file “NAME_SURNAME_tvplus.zip”.

Please do not forget to add your name also to e-mail's subject.

2. Project Detail

In this project, we expect you to have 2 pages:

- HOME PAGE (Consists of 3 sections)

- Header

- . There will be only a logo on left side

- List the popular movies from the “moviedb API” in a horizontal list (API details are explained in Step 3)

- . List should display 20 movies at most
- . List should display 4 items initially
- . List should navigate between movies using left-right keys and handle “Enter” key press
- . List should display the next movie correctly when right key pressed on last visible movie
- . List should also display back the previous movie when left key is pressed on first visible movie
- . List should stop navigating at 0th and 20th item
- . List should have a current focus and focus animation

- List items should consist of :

- . Backdrop image (250x141px)
- . Movie name
- . Genres

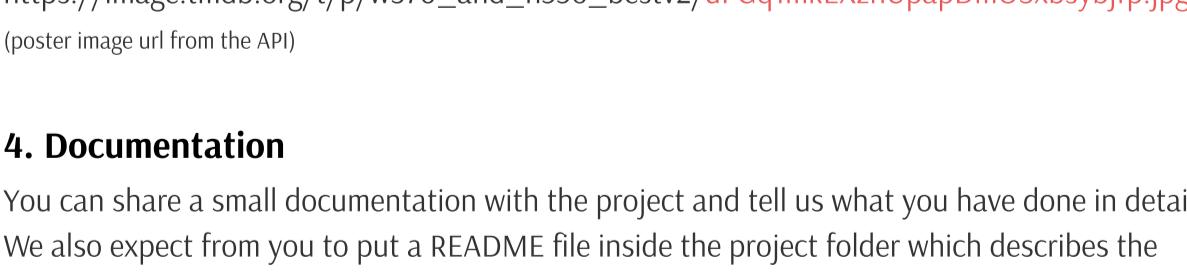
. Short description (max 2 lines) – cut short if description is long

. Tagline (Written and Directed by)

. Release Date

- Footer

- . There will be only a copyright text on left side



Home page

- MOVIE DETAIL PAGE (Consists of 4 sections)

- Header

- . There will be only a logo on left side

- A “Go Back” button

- . To navigate last page on top of the route stack

- Movie Detail Box

- . Display the poster on left side (370x556px)

. Display:

- .. Movie Title
- .. Release Date (Year only)

.. Genre

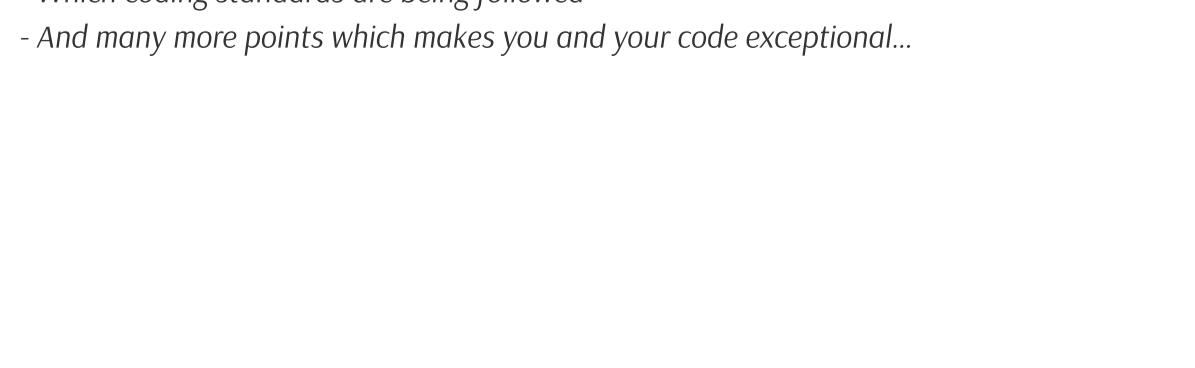
.. Tagline

.. Full Description

.. Rating (displayed with stars) – Number of Votes

- Footer

- . There will be only a copyright text on left side



Detail page

3. APIs

You can create a free account from The Movie Database website (<https://www.themoviedb.org/>) for personal use. After you verify your registered e-mail address, you can login to this website and then from your account settings page, you can create an API key.

You can access the API list from this [url](#). You can also check their [website](#) which uses their own APIs as a reference point. Inside this project we expect you to use:

- Get Popular Movies API which you can access from this [url](#)

- Get Movie Details API which you can access from this [url](#)

As an extra information:

- You can get the backdrop image of a movie:

https://image.tmdb.org/t/p/w500_and_h282_face/6fkqwqLEcDZOEAnBBfKAniwNxtx.jpg (backdrop image url from the API)

- You can get the poster image of a movie:

https://image.tmdb.org/t/p/w370_and_h556_bestv2/uPGq1mkEXznUpapDmOSxbsybjfp.jpg (poster image url from the API)

4. Documentation

You can share a small documentation with the project and tell us what you have done in detail. We also expect from you to put a README file inside the project folder which describes the necessary steps to run the project.

5. Acceptance Criteria

There is no acceptance criteria except for the code to run and display a home page. The code you will share will be evaluated by our Senior Software Engineers based on:

- How much the project defined in this document is coded
- How proficient the solution is
- Which state-of-art Javascript technologies are used and how well they are being used
- How clean is the code
- Which coding standards are being followed
- And many more points which makes you and your code exceptional...