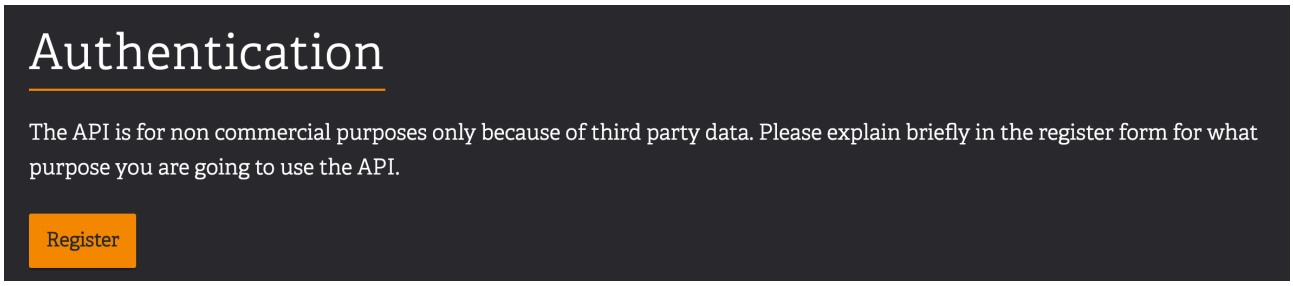
Dr. Cinema

In this assignment we are going to create an application which displays movies that are currently playing in cinemas. We will be using an external API to get information related to cinemas and their movies. Good luck!

External API

We will be using an API provided by kvikmyndir.is which is documented here: http://api.kvikmyndir.is/. In order to use the API you must register by pressing the 'Register' button shown below:



When you have successfully filled out the form and submitted it, you can start by getting your authentication header. The authentication flow works as follows:

1. Send a HTTP POST request to http://api.kvikmyndir.is/authenticate and include within the body a JSON object consisting of two properties: **username** and **password** (*which is the username and password provided when registering on the website*)

2. If this was successful you will get a response which includes your access token, which is valid for 24 hours

3. All future requests must include this token either as a query parameter, e.g. http://api.kvikmyndir.is/movies?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUz or by populating a HTTP header called **x-access-token** with the value of the token

4. When the token expires (*after 24 hours*) these steps need to be reiterated

I would recommend using either **fetch** (https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) or **axios** (https://github.com/axios/axios) to make the actual requests in **React Native**

Note: In the documentation it uses **http://api.biomynd.is** for some demonstrations but the actual URL is **http://api.kvikmyndir.is**

Structure

If the code does not follow the principles laid down here below, each group can receive up to -2 in deduction for their assignment. Here are the rules of structure:

- Code should be broken up into components which follow the **S**ingle **R**esponsibility **P**rinciple

- Common logic should reside in a separate module and imported into components which make use of this logic

- The folder structure should be in accordance to the course lectures

- Consistency in code, meaning all group members should follow the same set of rules, e.g. 4 spaces as indent, Egyptian style curly braces, etc… hint: *eslint can help*

Assignment description

Here below is an enlisting of all the functionality this application should implement:

- **(15%)** Cinemas screen
  - **(10%)** A user should see a list of all cinemas
    - **(2.5%)** Alphabetically ordered (*ascending order*)
    - **(7.5%)** Displaying name and website
  - **(5%)** Each cinema in the list should be clickable and on click should navigate to a detailed screen of the selected cinema
- **(20%)** Cinema detail screen
  - **(5%)** A user should see detailed information on the selected cinema
    - **(1%)** Name
    - **(1%)** Description
    - **(1%)** Complete address (*including street name and city*)
    - **(1%)** Phone
    - **(1%)** Website
  - **(15%)** A user should see all movies which are being shown in the selected cinema
    - **(10%)** A movie should display a thumbnail, name, release year and genres
    - **(5%)** Each movie in the list should be clickable and when clicked the app should navigate to a detailed screen for the selected movie
- **(30%)** Movie screen
  - **(10%)** A user should see detailed information about the selected movie
    - **(1%)** Name
    - **(1%)** Image (*poster*)
    - **(1%)** Plot
    - **(1%)** Duration (*in minutes*)
    - **(1%)** Year of release
    - **(5%)** Genres
  - **(20%)** A user should be able to see the show times of the movie (*only in the cinema which was selected when this particular movie was selected*) and a way to purchase a ticket via a link

- **(10%)** Upcoming movies screen
    - **(10%)** A user should see a list of all upcoming movies
        - **(2.5%)** Ordered by release date (*descending order*)
        - **(7.5%)** An upcoming movie should display a thumbnail, name and release date
    - **Extra points**:
        - **(10%)** A trailer which is associated with the upcoming movie can be watched within the application (*note that not all upcoming movies have a trailer, so only those who have*)
- **(25%)** Redux
    - **(10%)** All network requests go through asynchronous action creators which update the Redux store state according to the resulted data
    - **(10%)** Components make use of mapStateToProps to retrieve the state changes from the Redux store
    - **(5%)** State is portioned with multiple reducers

Submission

A single compressed file (***.zip, *.rar**) should be submitted. Don't forget to delete **node_modules/** from the folder before compressing it and eventually submitting.