

**Workshop Practice 2**

**CS - Mobile Development**

In this practical lab, we will be practicing on calling open-source API page. This practice we will try to attempt build a simple weather app that calls from an open-source API.

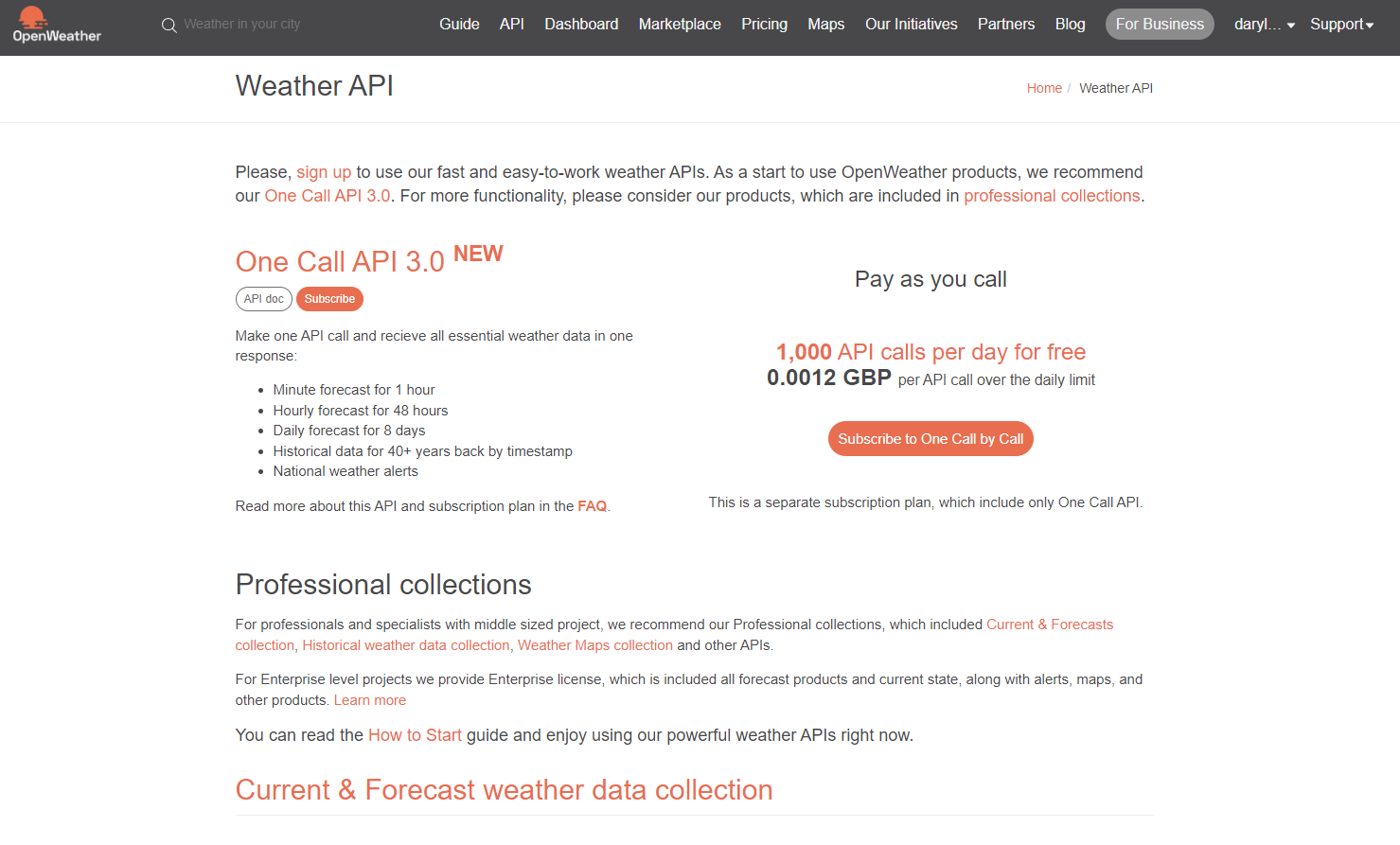
Below is how the final outlook of the weather App.



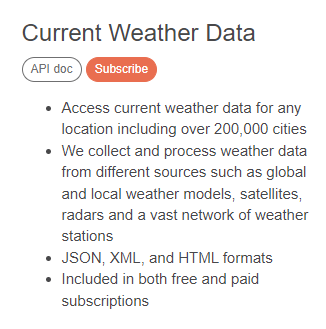
The screenshot data are pulled from an open-source weather API, which we will be trying.

Follow the steps below to start your weather RS app.

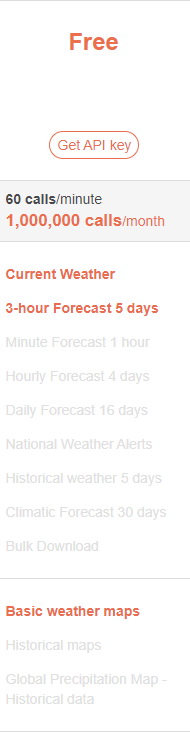
1. Create a new project with the name “weatherapp”.
2. Install the following library to your root project.
   1. expo install @expo/vector-icons (The vector-icons is a library that support nice and simple library icons, which we will be using in the application)
   2. expo install @react-native-community/geolocation. (Similar to expo-location, we require this to get our current position which the weather API requires later on. In order to pass us accurate weather data)
   3. Run this if only you are facing any issues with the geolocation library. (**react-native link @react-native-community/geolocation**)
3. Next, we will setup the weather API from the following link:
   1. <https://openweathermap.org/api>



* 1. Create an account from the above link, in order to have access to the API.
  2. Once you have created and verified your account, head over to the API link.
  3. Head down to the **Current Weather Data** section and hit on subscribe.



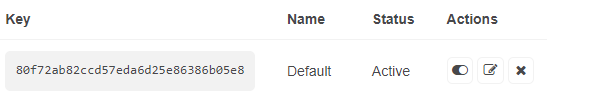
* 1. Scroll to the bottom table and click on the **Get API key** under the **Free** section.



* 1. In the next page, click on API keys tab.



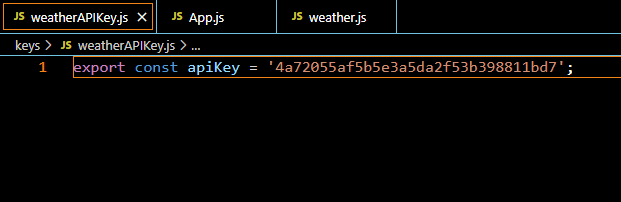
* 1. You will be able to find an **active key** automatically created for you which we will be using later.



1. Now you are all setup, let’s start building the application.
2. In your project folder, create 2 folders in the root project.
   1. **components** folder (We will be creating a sub component to perform some of the interface parts in the weather app)
   2. **keys** folder (This folder is meant for storing the API key from the open-source weather site)

**Programming the <weatherAPIKey.js> File**

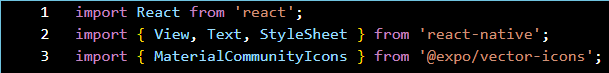
1. Under the **keys** folder, create a new Javascript file called “weatherAPIKey.js”.
2. Add your personal API key from the earlier open-source weather site. Something like this:



1. Great! Next, we will be adding a new component that forms the interface of the weather app.
2. Under the **components** folder, create a new Javascript file called “weather.js”.

**Programming the <weather.js> File**

1. We will perform the imports that we have installed earlier on.



1. Next, we will be creating the weather interface and its style.



The above Weather const, sets TWO (2) argument that we require to insert the information to the respective elements.

We will be using the **vector-icons –** **MaterialCommunityIcons** to have some nice and simple icons.

1. Next, we will style the following elements.

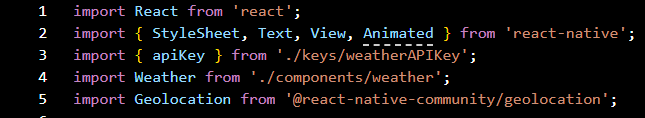


1. Lastly, as this is not the main **App** file. We have to perform the export, so that our **App.js** file is able to access this weather component that we have set up.



**Programming the <App.js> File**

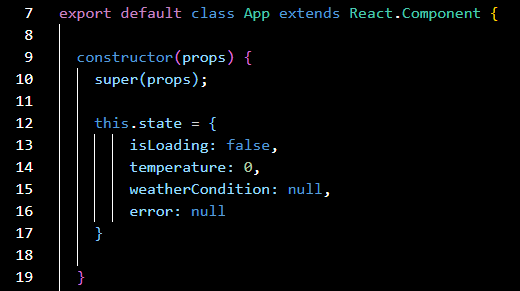
1. Lastly, as this is not the main **App** file. We have to perform the export, so that our **App.js** file is able to access this weather component that we have set up.
2. We will perform some imports in the **App.js** file.



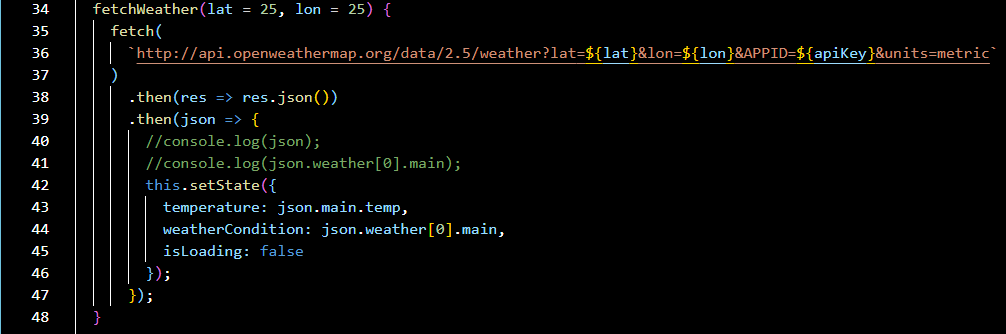
1. For this **App.js** structure, we will be extending to the **React.Component**.



1. Next, we will setup a few variable states that we will be using throughout the app.



1. Now let’s write the codes to fetch the weather data from the open-source, using the following link. **(`http://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&APPID=${apiKey}&units=metric`)**

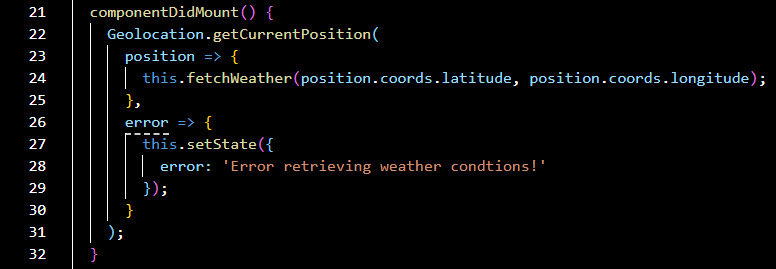


The above code snippet, performs a **fetch** to the source and in return the weather data in **json** structure.

You may perform the console logs to test out the data first.

1. Now we are left with some final pieces. Let’s perform a **componentDidMount()** to run the **fetchWeather()** function in 19.

**\* The componentDidMount() method allows us to execute the React code when the component is already placed in the DOM (Document Object Model). This method is called during the Mounting phase of the React Life-cycle i.e after the component is rendered.**

****

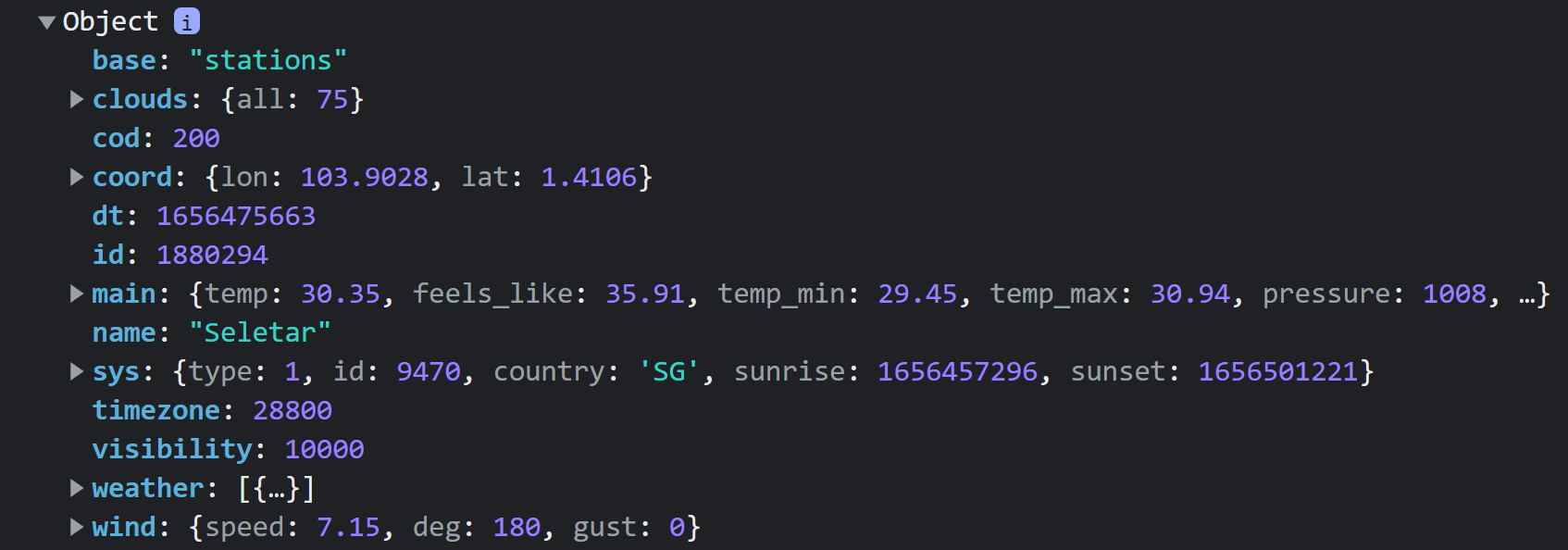
1. Great! Now we are left with adding the elements and its styles.



As the data requires the internet to retrieve the json weather data, we have to perform the **isLoading** to inform the user that the app is fetching the data. Similar to how apps perform load animation while retrieving contents.

The **<Weather weather={….} temperature={….} />** element uses the component that we have created from the **weather.js** file.

1. Awesome! You are done with your first simple weather app linked to an open-source API.
2. Try customizing the app with the respective json data we have gotten. Here’s a list of the json data provided by the weather API.



1. Add some data like the following:

