

Epicture

Tek2 Project | Epitech { Lille }

Documentation

Table of contents

General informations	3
Presentation of the Epicture Project.....	3
How to use the Epicture ?	3
The Epicture Project	4
Architecture of the project	4
Explanations of the components	5
Image	5
Layout	5
Picker	5
Search	5
Vote	6
Dependencies	6
React-native-image-crop-picker	6
React-native-vector-icons	6
React-native-video.....	6
React-native-webview	6
React-navigation.....	6
React-navigation-stack.....	7

General informations

Presentation of the Epicture Project

The Epicture project is a project for third-year students at Epitech, the School of Innovation and Computer Expertise. The purpose of this project is to create a photo sharing platform. To do this, we used React Native technology, a mobile application framework created by Facebook.

How to use the Epicture ?

To use the project, clone the repository then execute:

- yarn install
- Define the ANDROID_SDK_ROOT environment variable
- react-native upgrade --legacy=true (If you don't have an android folde)
- react-native link
- react-native start
- react-native run-android

To build the android APK, clone the repository and execute these commands:

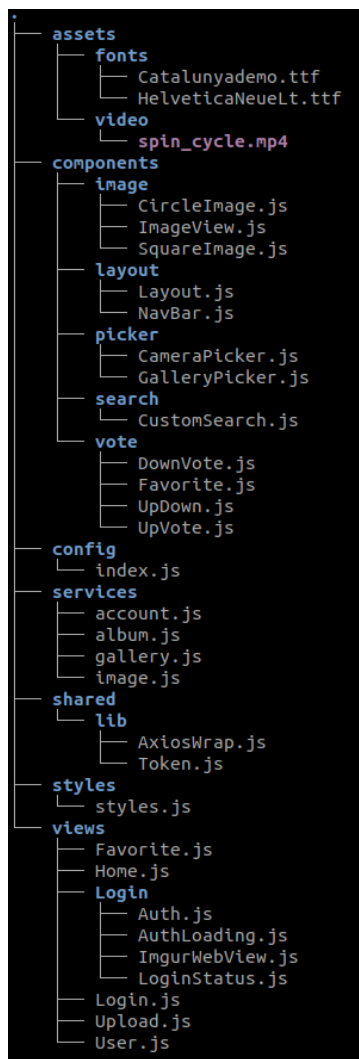
- yarn install
- Define the ANDROID_SDK_ROOT environment variable
- react-native upgrade --legacy=true (If you don't have an android folder)
- react-native link
- cd android && ./gradlew assembleRelease

You must have a device connected to your computer or an emulator launched to be able to run the project. After those commands, the application would appear on your device. You can now use it.

The Epicture Project

Architecture of the project

We have divided the project into several different folders to make it clearer. Here is the architecture of the folders that can be found in `src/`.



assets: contains the assets of the project, as fonts used in the application or videos.

components: contains the different components shared in the application. Don't confuse it with the *views* folder.

The *components* folder has sub-folders where you can find each shared component.

services: contains every function where there are Imgur API's calls. Each file has got its own lot of functions according to his name.

styles: contains a shared stylesheet in the one you can find general styles properties as 'white' or 'bold', to prevent inline styles.

views: contains the different pages you can find while navigating in the application. In those views are called files from the *components* folder.

Explanations of the components

There are few folders in the *components* one. This part of the documentation will allow you to understand each of them and thus be able to use them more easily.

Image

Name	Description	Props
CircleImage	Displays an image with a circle design	Pic: url of the picture Size: size of the circle image
ImageView	Display a complete image card with the name of the author, the title, the description, downs and ups votes. Used in Home view	Image: object containing every informations of the post (username, profilepic, url, title, description, ups, downs, vote, favorite)
SquareImage	Displays an image with a square image	Item: object containing a thumbnail and the id of the picture displayed

Layout

Name	Description	Props
Layout	Handle the layout of the application, containing the views	Title: title of the view Font: font used for the title fontSize: size of the font
Navbar	Handle the navigation bar	//

Picker

Name	Description	Props
CameraPicker	Display the camera image picker	//
GalleryPicker	Display the gallery image picker	//

Search

Name	Description	Props
CustomSearch	Display the search bar and the filter button to search images with the Imgur API	updateSearch: method called to refresh the images object from the parent changeSort: method which handle the filter of the search search: search string sort: filter

Vote

Name	Description	Props
DownVote	Display the down arrow and the number of downs.	imageHash : id of the image vote : current vote iconSize : size of the icon ups : number of ups votes of the image handleVote : method to handle the up and down votes
Favorite	Display the heart icon and handle the favorite management	imageHash : id of the image iconSize : size of the favorite icon
UpDown	Contains the UpVote and the DownVote. It is the one you need to use if you want manage the votes of an image	imageHash : id of the image
UpVote	Display the up arrow and the number of ups.	imageHash : id of the image vote : current vote iconSize : size of the icon ups : number of ups votes of the image handleVote : method to handle the up and down votes

Dependencies

React-native-image-crop-picker

This dependency is used to handle the camera and the gallery, to upload pictures from the mobile device.

React-native-vector-icons

To have responsive icons, we used the react-native-vector-icons dependency. Every icons you can see in the application is from this dependency.

React-native-video

We used the react-native-video to display a video in the background of the login screen.

React-native-webview

To handle the connection with the Imgur's API, we used a webview, displayed by the react-native-webview dependency.

React-navigation

The react-navigation is used to handle the navigation between each views in the mobile application.

React-navigation-stack

React-navigation-stack dependency handle the “back” arrow. Each new screen is placed on top of a stack.