React Native on Android

Installing Dev Environment

On ubuntu you can install the android SDK with `apt-get install android-sdk` .

You'll need a target device to test your app against. You can plug it in and use adb to do the hard work. Make sure to install the Expo App on your target device - this will serve as a host on the device that allows it to run your new app in development and do hot-reloading of your code.

Creating a Project

Using npx create a new project with the tooling provided by React Native like so:

```
npx create-expo-app AwesomeProject
cd AwesomeProject
npm start # you can also use: npx expo start
```

Running the Project

Start the app with `npm run android` you might get error messages about not knowing where android home is. On ubuntu android home installed from deb packages is normally at **`/usr/lib/android-sdk`** (source) so just run:

`export ANDROID_HOME=/usr/lib/android-sdk` before you try to launch the app.

Theming

I'm looking at using Native Base as a component library in my apps.

Persisting Data

React Native is pretty modular so you need to use libraries to do most things. Use a storage library to store data. This MMKV storage library appears to be pretty efficient and allows you to store state across multiple databases very easily and quickly

Navigation

Using react-navigation to manage navigation within the app and show different views. The hello world tutorial shows how this works

Using the Camera

The expo-camera library seems to be a nice way to do cross-device camera stuff. I got it working very quickly.

Working Minimal Example Camera App

```
import { StatusBar } from 'expo-status-bar';
import { Camera, CameraType } from 'expo-camera';
import { useState } from 'react';
import { Button, StyleSheet, Text, TouchableOpacity, View } from 'react-native';

export default function App() {
  const [type, setType] = useState(CameraType.back);
  const [permission, requestPermission] = Camera.useCameraPermissions();

  if (!permission) {
    return (<View><Text>No camera m8</Text></View>)
  }

  if (!permission.granted) {
    return(<View>
        <Text>Clicky to grant perms m8</Text>
        <Button title='Grant Perms' onPress={requestPermission}/>
      </View>)
  }

  function toggleCameraType() {
    setType(current => (current === CameraType.back ? CameraType.front :
```

```
CameraType.back));
  }

  return (
    <View style={styles.container}>
      <Camera style={styles.camera} type={type}>
        <View style={styles.buttonContainer}>
          <TouchableOpacity style={styles.button} onPress={toggleCameraType}>
            <Text style={styles.text}>Flip Camera</Text>
          </TouchableOpacity>
        </View>
      </Camera>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
  },
  camera:{
    flex: 1,
  },
  button: {
    flex: 1,
    alignSelf: 'flex-end',
    alignItems: 'center',
  },
  buttonContainer: {
    flex: 1,
    flexDirection: 'row',
    backgroundColor: 'transparent',
    margin: 64,
  },
  text:{

  }
});
```

###

Revision #8
Created 2022-10-30 10:04:20 UTC by James
Updated 2024-01-21 14:54:01 UTC by James