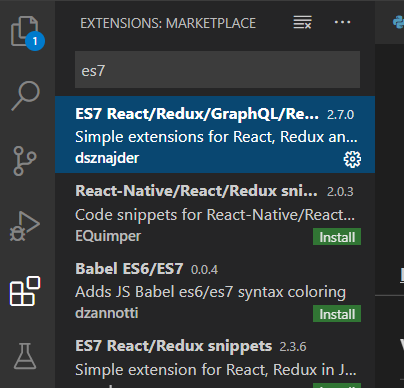
# React Native

# Install react in windows

## Install VS Code IDE

VS Code : - <https://code.visualstudio.com/download>

1. Install extension for react native



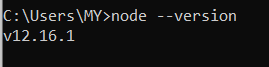
## Install NodeJS and NPM

Download :- <https://nodejs.org/en/>

## Create React APP

### Check version

Open terminal and type:



### Install React from terminal:

1. Change current working directory to project directory.

cd D:

1. Create a project directory for react app

mkdir REACT

1. Move to REACT folder

cd REACT

1. Install React from terminal:

npm install -g create-react-app

If successful, you should be able to get version:

create-react-app --version

1. Create react project:

create-react-app <projectname>

Example – This step may take 5 -15 minutes

D:\REACT>create-react-app workout

Creating a new React app in D:\REACT\workout.

Installing packages. This might take a couple of minutes.

Installing react, react-dom, and react-scripts with cra-template...

> core-js@2.6.11 postinstall D:\REACT\workout\node\_modules\babel-runtime\node\_modules\core-js

> node -e "try{require('./postinstall')}catch(e){}"

> core-js@3.6.4 postinstall D:\REACT\workout\node\_modules\core-js

> node -e "try{require('./postinstall')}catch(e){}"

> core-js-pure@3.6.4 postinstall D:\REACT\workout\node\_modules\core-js-pure

> node -e "try{require('./postinstall')}catch(e){}"

+ cra-template@1.0.3

+ react@16.13.1

+ react-scripts@3.4.1

+ react-dom@16.13.1

added 1574 packages from 750 contributors and audited 930020 packages in 384.891s

54 packages are looking for funding

  run `npm fund` for details

found 2 low severity vulnerabilities

  run `npm audit fix` to fix them, or `npm audit` for details

Git repo not initialized Error: Command failed: git --version

    at checkExecSyncError (child\_process.js:630:11)

    at execSync (child\_process.js:666:15)

    at tryGitInit (D:\REACT\workout\node\_modules\react-scripts\scripts\init.js:46:5)

    at module.exports (D:\REACT\workout\node\_modules\react-scripts\scripts\init.js:266:7)

    at [eval]:3:14

    at Script.runInThisContext (vm.js:120:20)

    at Object.runInThisContext (vm.js:311:38)

    at Object.<anonymous> ([eval]-wrapper:10:26)

    at Module.\_compile (internal/modules/cjs/loader.js:1158:30)

    at evalScript (internal/process/execution.js:94:25) {

  status: 1,

  signal: null,

  output: [ null, null, null ],

  pid: 1648,

  stdout: null,

  stderr: null

}

Installing template dependencies using npm...

npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node\_modules\webpack-dev-server\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node\_modules\watchpack\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node\_modules\jest-haste-map\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ @testing-library/jest-dom@4.2.4

+ @testing-library/react@9.5.0

+ @testing-library/user-event@7.2.1

added 36 packages from 54 contributors and audited 930226 packages in 36.973s

54 packages are looking for funding

  run `npm fund` for details

found 2 low severity vulnerabilities

  run `npm audit fix` to fix them, or `npm audit` for details

Removing template package using npm...

npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node\_modules\webpack-dev-server\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node\_modules\watchpack\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node\_modules\jest-haste-map\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

removed 1 package and audited 930225 packages in 12.015s

54 packages are looking for funding

  run `npm fund` for details

found 2 low severity vulnerabilities

  run `npm audit fix` to fix them, or `npm audit` for details

Success! Created workout at D:\REACT\workout

Inside that directory, you can run several commands:

  npm start

    Starts the development server.

  npm run build

    Bundles the app into static files for production.

  npm test

    Starts the test runner.

  npm run eject

    Removes this tool and copies build dependencies, configuration files

    and scripts into the app directory. If you do this, you can’t go back!

We suggest that you begin by typing:

  cd workout

  npm start

Happy hacking!

1. Run project:

cd <projectname>

npm start

Expected output in terminal (and should auto open browser to port 3000):

Compiled successfully!

You can now view <project\_name> in the browser.

  Local:            http://localhost:3000/

  On Your Network:  http://<your\_ip\_address>:3000/

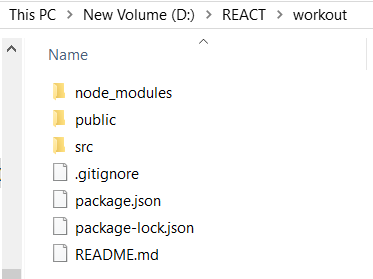
Note that the development build is not optimized.

To create a production build, use yarn build.

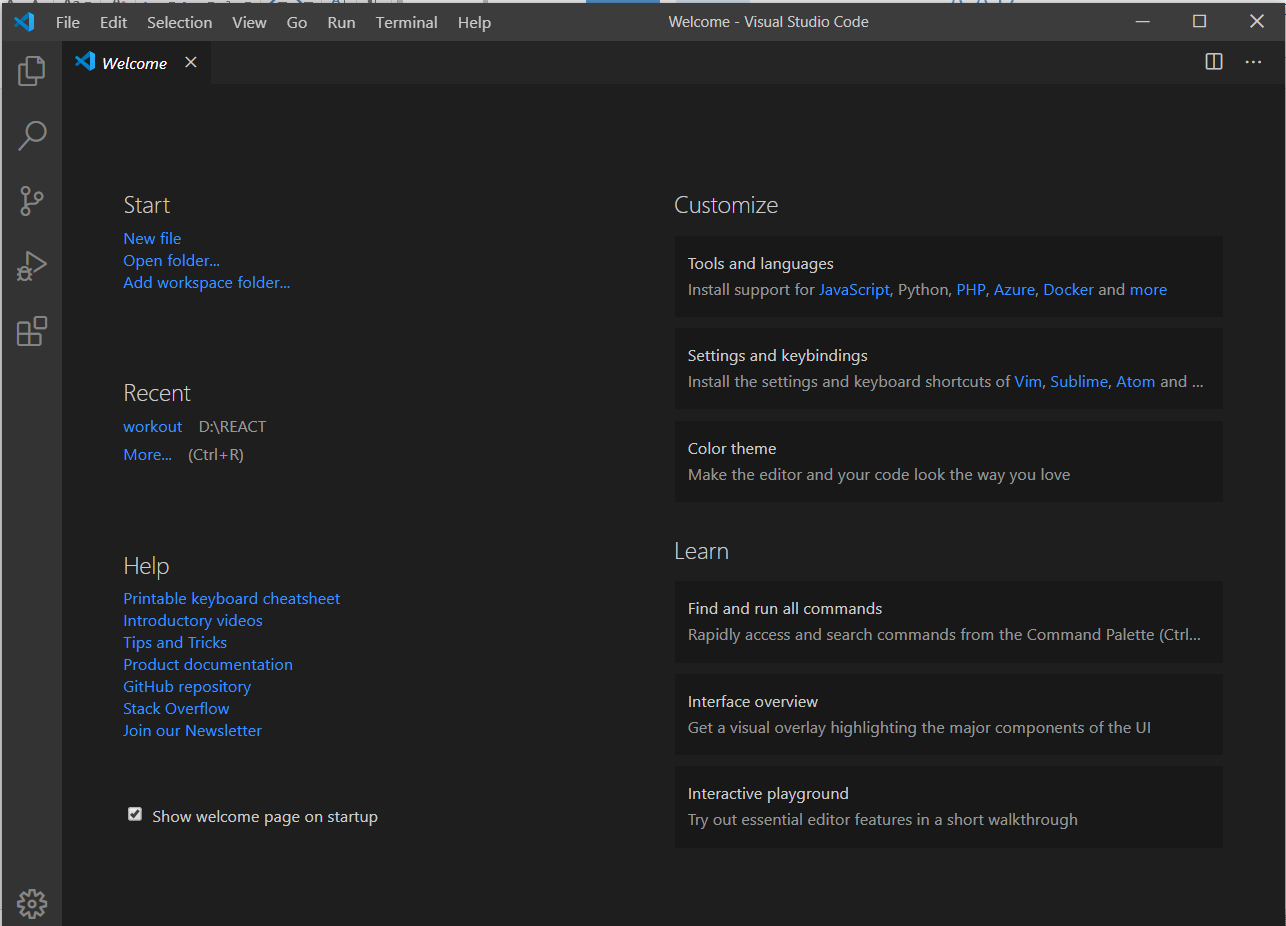
# Create Hello World app in react

Now the project is created in the following directory,

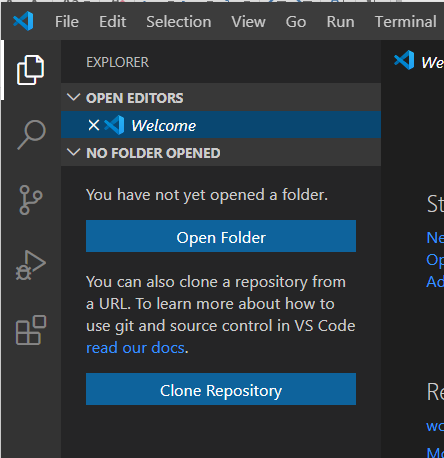
Location:- D:\REACT\workout



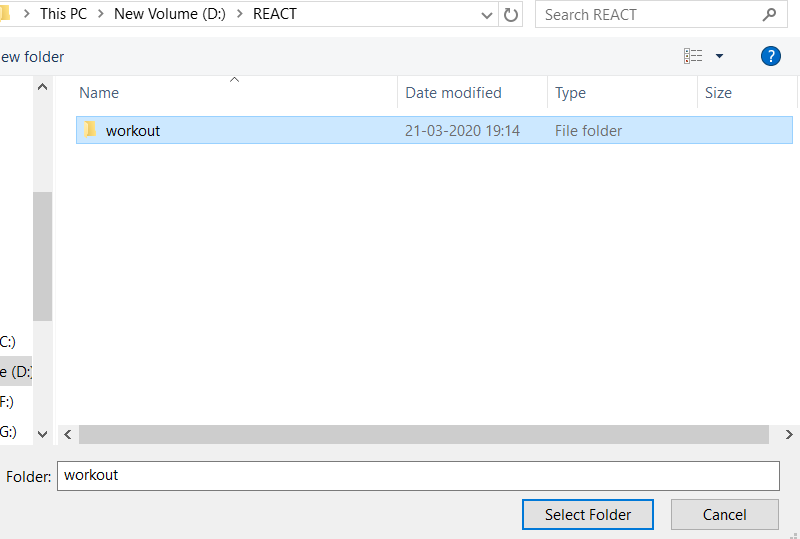
1. Open VS Code



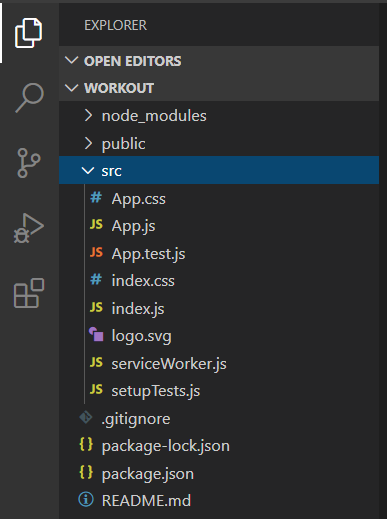
1. Click first option in the left side corner



1. Click open folder option
2. Select your project folder

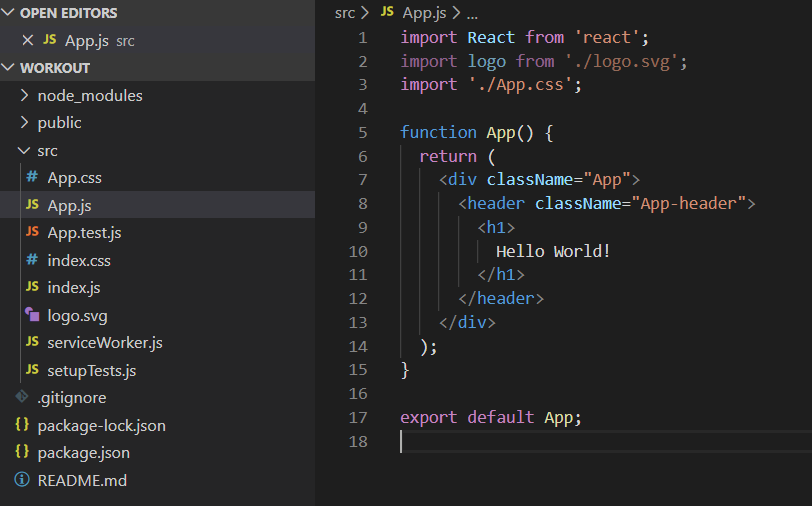


1. Open “src”



1. Click “App.js” & Edit the component inside <header> tag

You can refer the following code



# Install react in android studio - not working

1. Install node js (if already installed skip this step)
2. Install android studio

<https://developer.android.com/studio>

# React for android app – not working

1. Install chocolate in windows
   1. Open windows powershell in administrative mode
   2. Copy and paste the following code

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

* 1. Wait a few seconds for the command to complete.
  2. If you don't see any errors, you are ready to use Chocolatey! Type choco or choco -?

1. Change current working directory to project directory
   1. In poershell > cd D:
   2. cd REACT\
   3. make a new directory for android apps> mkdir android
   4. cd android
2. React Native also requires a recent version of the Java SE Development Kit (JDK), as well as Python 2. Both can be installed using Chocolatey.
   1. In powershell > choco install -y nodejs.install python2 jdk8

# Create android app in react

1. Open cmd and run the following code – this step takes (30 min – 1 hr)

npm install -g expo-cli

1. Change current working directory to project directory
   1. In cmd > cd D:
   2. cd REACT\
   3. make a new directory for android apps> mkdir android
   4. cd android
2. Then run the following commands to create a new React Native project called "AwesomeProject": (time: 5 – 10min)

expo init AwesomeProject

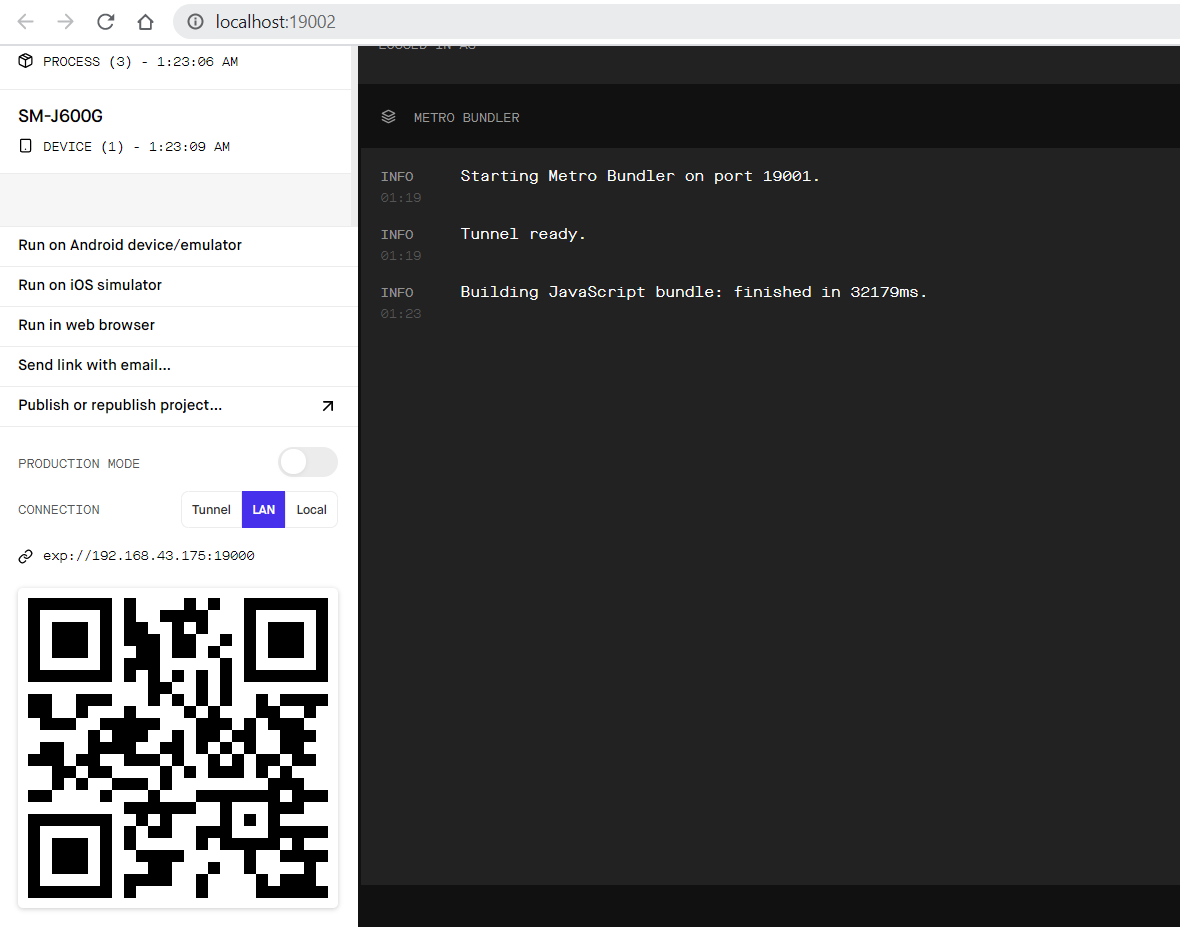
cd AwesomeProject

1. To open project in vs code run the command in cmd

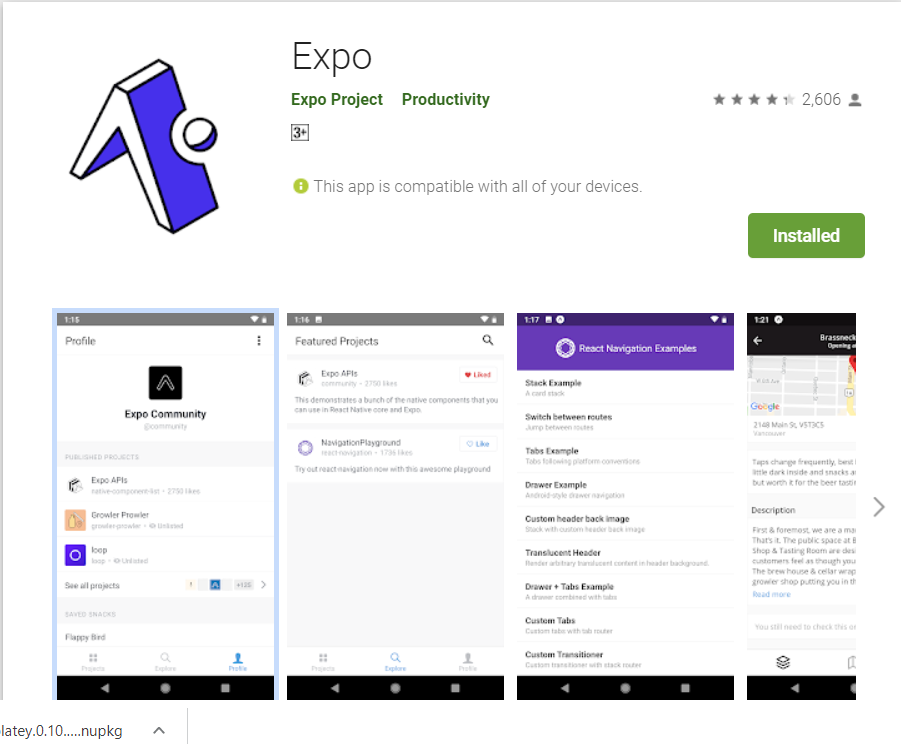
code .

1. To start react project. Run this code in cmd

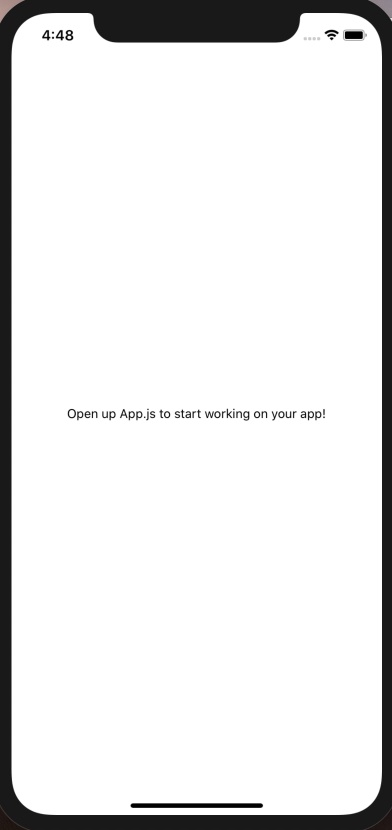
npm start # you can also use: expo start



1. Download “Expo Client” app from android play store



1. Mobile phone and computer should be in same network
2. Scan QR code



# View, Style

Reference: <https://www.youtube.com/watch?v=_YydVvnjNFE&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=3&pbjreload=10>

1. Create view and style – open “App.js” file

import React from 'react';

import { StyleSheet, Text, View } from 'react-native';

export default function App() {

  return (

    <View style={styles.container}>

      {/\* view for hello world \*/}

      <View style={styles.header}>

        <Text style={styles.boldText}>Hello World</Text>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

  // style for hello world view

  header: {

    backgroundColor: 'pink',

    padding: 20

  },

});

1. Create another view and add style to this

import React from 'react';

import { StyleSheet, Text, View } from 'react-native';

export default function App() {

  return (

    <View style={styles.container}>

      <View style={styles.header}>

        <Text style={styles.boldText}>Hello World</Text>

      </View>

      <View style={styles.body}>

        <Text>Lorem ipsum dolor sit amet</Text>

        <Text>Lorem ipsum dolor sit amet</Text>

        <Text>Lorem ipsum dolor sit amet</Text>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

  header: {

    backgroundColor: 'pink',

    padding: 20

  },

  boldText:{

    fontWeight: "bold"

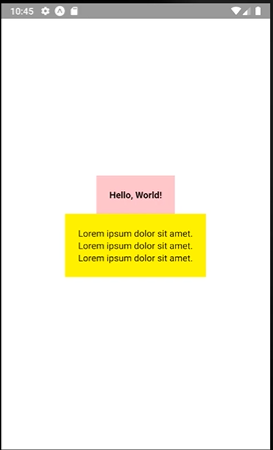
  },

  body:{

    backgroundColor:'yellow'

  }

});



# Using state

Reference : <https://www.youtube.com/watch?v=1FiIYaRr148&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=4>

1. import {useState} using following step

import React, {useState} from 'react';

1. create a constant variable inside app() – App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  return (

    <View style={styles.container}>

      <Text></Text>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});

1. Add variable name inside <Text> component

<Text>My name is {name}</Text>

Complete code

import React, {useState} from 'react';

import { StyleSheet, Text, View } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  return (

    <View style={styles.container}>

      <Text>My name is {name}</Text>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});



## Create button

1. Import Button widget

import { StyleSheet, Text, View, Button } from 'react-native';

1. Create a view inside main view for button

<View style={styles.buttonContainer}>

        <Button title="update status" />

      </View>

Complete code – App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, Button } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  return (

    <View style={styles.container}>

      <Text>My name is {name}</Text>

      <View style={styles.buttonContainer}>

        <Button title="update status" />

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});

## Onpress action to button

1. Create an arrow function clickHandler

const clickHandler = () => {

    setName("chun-li");

  }

1. Add onPress event to button

<Button title="update status" onPress={clickHandler} />

Complete code

import React, {useState} from 'react';

import { StyleSheet, Text, View, Button } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  const clickHandler = () => {

    setName("chun-li");

  }

  return (

    <View style={styles.container}>

      <Text>My name is {name}</Text>

      <View style={styles.buttonContainer}>

        <Button title="update status" onPress={clickHandler} />

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});

1. Press update status button in app

## Add key value pair state

1. Create new state

const [person, setPerson] = useState({name:'mario', age: 40});

1. Add <text> tag to display this

<Text>His name is {person.name} and his age is {person.age}</Text>

Complete code – App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, Button } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  const [person, setPerson] = useState({name:'mario', age: 40});

  const clickHandler = () => {

    setName("chun-li");

  }

  return (

    <View style={styles.container}>

      <Text>My name is {name}</Text>

      <Text>His name is {person.name} and his age is {person.age}</Text>

      <View style={styles.buttonContainer}>

        <Button title="update status" onPress={clickHandler} />

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

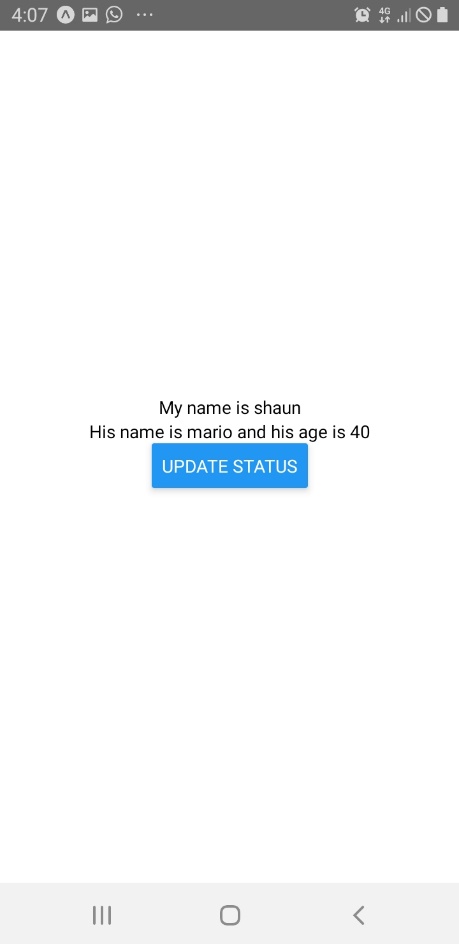
    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

});



1. To add onPress handler add setPerson functioin inside clickHandler function

const clickHandler = () => {

    setName("chun-li");

    setPerson({name:'luigi', age: 45});

  }

## Text input

1. Import TextInput

import { StyleSheet, Text, View, Button, TextInput } from 'react-native';

1. Create two constants

  const [name, setName] = useState('shaun');

  const [age, setAge] = useState(20);

1. Add textinput field

<TextInput />

1. Add placeholder and onChangeText to <TextInput> field

Complete code -> App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, Button, TextInput } from 'react-native';

export default function App() {

  const [name, setName] = useState('shaun');

  const [age, setAge] = useState(20);

  const clickHandler = () => {

    setName("chun-li");

    setAge(45);

  }

  return (

    <View style={styles.container}>

      <Text>Enter your name:</Text>

      <TextInput

        placeholder="e.g John Doe"

        style={styles.input}

        onChangeText={(val) => setName(val)}

        />

      <Text>Name is {name} and age is {age}</Text>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

  },

  input: {

    borderWidth:1,

    borderColor: "#777",

    padding:8,

    margin:8,

    width:200,

  }

});

## Textinput properties

1. Multiline – multiple line input (press enter to add new line)

<TextInput

        multiline

        placeholder="e.g John Doe"

        style={styles.input}

        onChangeText={(val) => setName(val)}

        />

1. KeyboardType

keyboardType = 'numeric'

## Add list

import React, {useState} from 'react';

import { StyleSheet, Text, View, Button, TextInput } from 'react-native';

export default function App() {

  const [people, setPeople] = useState([{name:'shaun', key:'1'},

                                      {name:'yoshi', key:'2'},

                                      {name:'mario', key:'3'},

                                      {name:'luigi', key:'4'},

                                      {name:'peach', key:'5'},

                                      {name:'toad', key:'6'},

                                      {name:'bowser', key:'7'},]);

  const clickHandler = () => {

  }

  return (

    <View style={styles.container}>

      { people.map ((item) => {

        return (

          <View key={item.key}>

            <Text style = {styles.item}>{item.name}</Text>

          </View>

        )

      })}

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    paddingTop:40,

    paddingHorizontal:20,

    // alignItems: 'center',

    // justifyContent: 'center',

  },

  item:{

    marginTop:24,

    padding:30,

    backgroundColor: 'pink',

    fontSize:24,

  }

});

## Scrolling list item

1. Import scroll view

import { StyleSheet, Text, View, ScrollView } from 'react-native';

1. Add scrollview component

<View style={styles.container}>

      <ScrollView>

        { people.map ((item) => {

          return (

            <View key={item.key}>

              <Text style = {styles.item}>{item.name}</Text>

            </View>

          )

        })}

      </ScrollView>

    </View>

Complete code

import React, {useState} from 'react';

import { StyleSheet, Text, View, ScrollView } from 'react-native';

export default function App() {

  const [people, setPeople] = useState([{name:'shaun', key:'1'},

                                      {name:'yoshi', key:'2'},

                                      {name:'mario', key:'3'},

                                      {name:'luigi', key:'4'},

                                      {name:'peach', key:'5'},

                                      {name:'toad', key:'6'},

                                      {name:'bowser', key:'7'},]);

  const clickHandler = () => {

  }

  return (

    <View style={styles.container}>

      <ScrollView>

        { people.map ((item) => {

          return (

            <View key={item.key}>

              <Text style = {styles.item}>{item.name}</Text>

            </View>

          )

        })}

      </ScrollView>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    paddingTop:40,

    paddingHorizontal:20,

    // alignItems: 'center',

    // justifyContent: 'center',

  },

  item:{

    marginTop:24,

    padding:30,

    backgroundColor: 'pink',

    fontSize:24,

  }

});

## Flat list

Flat list only loads some items into the screen. So we can load huge amount of data in the view.

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList } from 'react-native';

export default function App() {

  const [people, setPeople] = useState([{name:'shaun', key:'1'},

                                      {name:'yoshi', key:'2'},

                                      {name:'mario', key:'3'},

                                      {name:'luigi', key:'4'},

                                      {name:'peach', key:'5'},

                                      {name:'toad', key:'6'},

                                      {name:'bowser', key:'7'},]);

  const clickHandler = () => {

  }

  return (

    <View style={styles.container}>

      <FlatList

        data = {people}

        renderItem={({ item }) => (

        <Text style= {styles.item}> { item.name }</Text>

        )}

      />

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    paddingTop:40,

    paddingHorizontal:20,

    // alignItems: 'center',

    // justifyContent: 'center',

  },

  item:{

    marginTop:24,

    padding:30,

    backgroundColor: 'pink',

    fontSize:24,

  }

});

Add some properties to Flat list

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList } from 'react-native';

export default function App() {

  const [people, setPeople] = useState([{name:'shaun', key:'1'},

                                      {name:'yoshi', key:'2'},

                                      {name:'mario', key:'3'},

                                      {name:'luigi', key:'4'},

                                      {name:'peach', key:'5'},

                                      {name:'toad', key:'6'},

                                      {name:'bowser', key:'7'},]);

  const clickHandler = () => {

  }

  return (

    <View style={styles.container}>

      <FlatList

        numColumns = {2}

        keyExtractor = {(item) => item.key}

        data = {people}

        renderItem={({ item }) => (

        <Text style= {styles.item}> { item.name }</Text>

        )}

      />

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    paddingTop:40,

    paddingHorizontal:20,

    // alignItems: 'center',

    // justifyContent: 'center',

  },

  item:{

    marginTop:24,

    marginHorizontal:10,

    padding:30,

    backgroundColor: 'pink',

    fontSize:24,

  }

});

## Touchable Component

<https://www.youtube.com/watch?v=QhX25YGf8qg&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=8>

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity } from 'react-native';

export default function App() {

  const [people, setPeople] = useState([{name:'shaun', key:'1'},

                                      {name:'yoshi', key:'2'},

                                      {name:'mario', key:'3'},

                                      {name:'luigi', key:'4'},

                                      {name:'peach', key:'5'},

                                      {name:'toad', key:'6'},

                                      {name:'bowser', key:'7'},]);

  const pressHandler = (id) => {

    console.log(id);

    setPeople((prevPeople)=>{

      return prevPeople.filter(person => person.key != id);

    });

  }

  return (

    <View style={styles.container}>

      <FlatList

        numColumns = {2}

        keyExtractor = {(item) => item.key}

        data = {people}

        renderItem={({ item }) => (

          <TouchableOpacity onPress={() => pressHandler(item.key)}>

            <Text style= {styles.item}> { item.name }</Text>

          </TouchableOpacity>

        )}

      />

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    paddingTop:40,

    paddingHorizontal:20,

    // alignItems: 'center',

    // justifyContent: 'center',

  },

  item:{

    marginTop:24,

    marginHorizontal:10,

    padding:30,

    backgroundColor: 'pink',

    fontSize:24,

  }

});

## TO –DO test app

<https://www.youtube.com/watch?v=uLHFPt9B2Os&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=9>

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity } from 'react-native';

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switch', key: '3'},

  ]);

  return (

    <View style={styles.container}>

      {/\* Header \*/}

      <View style={styles.content}>

        {/\* To do form \*/}

        <View style={styles.list}>

          <FlatList

          data = {todos}

          renderItem = {({ item }) => (

          <Text> { item.text }</Text>

          )}

          />

        </View>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

  },

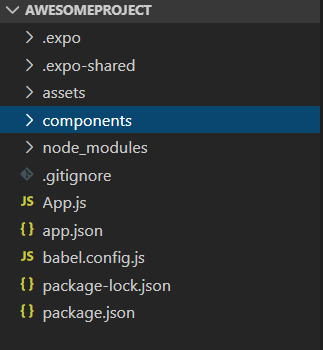
  list: {

    marginTop: 20,

  }

});

Create a folder components in project directory



Create a file “header.js” inside components

### Header.js

import React from 'react';

import { StyleSheet, Text, View} from 'react-native';

export default function Header(){

    return (

        <View style={styles.header}>

            <Text style={styles.title}>My ToDos</Text>

        </View>

    )

}

const styles = StyleSheet.create({

    header: {

        height: 80,

        paddingTop: 38,

        backgroundColor: 'coral'

    },

    title: {

        textAlign: 'center',

        color: '#fff',

        fontSize: 20,

        fontWeight: 'bold'

    }

});

App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity } from 'react-native';

import Header from "./components/header"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switch', key: '3'},

  ]);

  return (

    <View style={styles.container}>

      <Header />

      <View style={styles.content}>

        {/\* To do form \*/}

        <View style={styles.list}>

          <FlatList

          data = {todos}

          renderItem = {({ item }) => (

          <Text> { item.text }</Text>

          )}

          />

        </View>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

  },

  list: {

    marginTop: 20,

  }

});

Create component for todos

Create a “todoitems.js” file

### Todoitems.js

import React from 'react';

import { StyleSheet, Text, TouchableOpacity} from 'react-native';

export default function ToDoitem({item, pressHandler}){

    return (

        <TouchableOpacity onPress={() => pressHandler(item.key)}>

            <Text style={styles.item}> { item.text } </Text>

        </TouchableOpacity>

    )

}

const styles = StyleSheet.create({

    item: {

        padding: 16,

        marginTop: 16,

        borderColor: '#bbb' ,

        borderWidth: 1,

        borderStyle :'dashed',

        borderRadius: 10

    },

    title: {

        textAlign: 'center',

        color: '#fff',

        fontSize: 20,

        fontWeight: 'bold'

    }

});

App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity } from 'react-native';

import Header from "./components/header"

import ToDoitem from "./components/todoitems"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switchs', key: '3'},

  ]);

  const pressHandler = (key) => {

    setTodos((prevToDos) => {

      return prevToDos.filter(todo => todo.key != key);

    });

  }

  return (

    <View style={styles.container}>

      <Header />

      <View style={styles.content}>

        {/\* To do form \*/}

        <View style={styles.list}>

          <FlatList

          data = {todos}

          renderItem = {({ item }) => (

            <ToDoitem item={item} pressHandler={pressHandler} />

          )}

          />

        </View>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

  },

  list: {

    marginTop: 20,

  }

});

AddTodo

Create a folder “addtod.js” inside components folder

Addtodo.js

import React, { useState } from 'react';

import { StyleSheet, Text, View, TextInput, Button } from 'react-native';

export default function AddTodo({submitHandler}) {

    const [text, setText] = useState('');

    const changeHandler = (val) => {

        setText(val);

    }

    return (

        <View>

            <TextInput

            style = {styles.input}

            placeholder = "new to do..."

            onChangeText = {(val) => changeHandler(val)}

            />

            <Button onPress={() => submitHandler(text)} title="add todo" color="coral" />

        </View>

    )

}

const styles = StyleSheet.create({

    input:{

        marginBottom:10,

        paddingHorizontal: 8,

        paddingVertical: 6,

        borderBottomWidth: 1,

        borderBottomColor: '#ddd',

    }

})

App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity } from 'react-native';

import Header from "./components/header"

import ToDoitem from "./components/todoitems"

import AddTodo from "./components/addTodo"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switchs', key: '3'},

  ]);

  const pressHandler = (key) => {

    setTodos((prevToDos) => {

      return prevToDos.filter(todo => todo.key != key);

    });

  }

  const submitHandler = (text) => {

    setTodos((prevToDos) => {

      return [

        { text: text, key: Math.random().toString()},

        ...prevToDos

      ]

    });

  }

  return (

    <View style={styles.container}>

      <Header />

      <View style={styles.content}>

        <AddTodo submitHandler={submitHandler}/>

        <View style={styles.list}>

          <FlatList

          data = {todos}

          renderItem = {({ item }) => (

            <ToDoitem item={item} pressHandler={pressHandler} />

          )}

          />

        </View>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

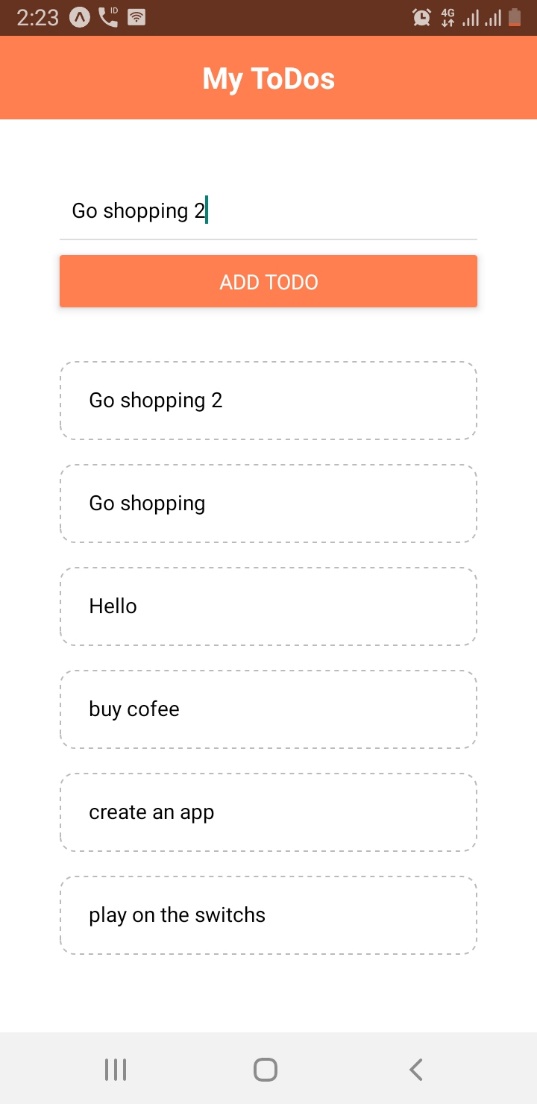
  },

  list: {

    marginTop: 20,

  }

});



## Alert button

App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity, Alert } from 'react-native';

import Header from "./components/header"

import ToDoitem from "./components/todoitems"

import AddTodo from "./components/addTodo"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switchs', key: '3'},

  ]);

  const pressHandler = (key) => {

    setTodos((prevToDos) => {

      return prevToDos.filter(todo => todo.key != key);

    });

  }

  const submitHandler = (text) => {

    if (text.length > 3){

      setTodos((prevToDos) => {

        return [

          { text: text, key: Math.random().toString()},

          ...prevToDos

        ]

      });

    }

    else{

      Alert.alert("Opps!", "Todos must be 3 char long.", [

        {text:"Understood", onPress: () => console.log("alert closed")}

      ]);

    }

  }

  return (

    <View style={styles.container}>

      <Header />

      <View style={styles.content}>

        <AddTodo submitHandler={submitHandler}/>

        <View style={styles.list}>

          <FlatList

          data = {todos}

          renderItem = {({ item }) => (

            <ToDoitem item={item} pressHandler={pressHandler} />

          )}

          />

        </View>

      </View>

    </View>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

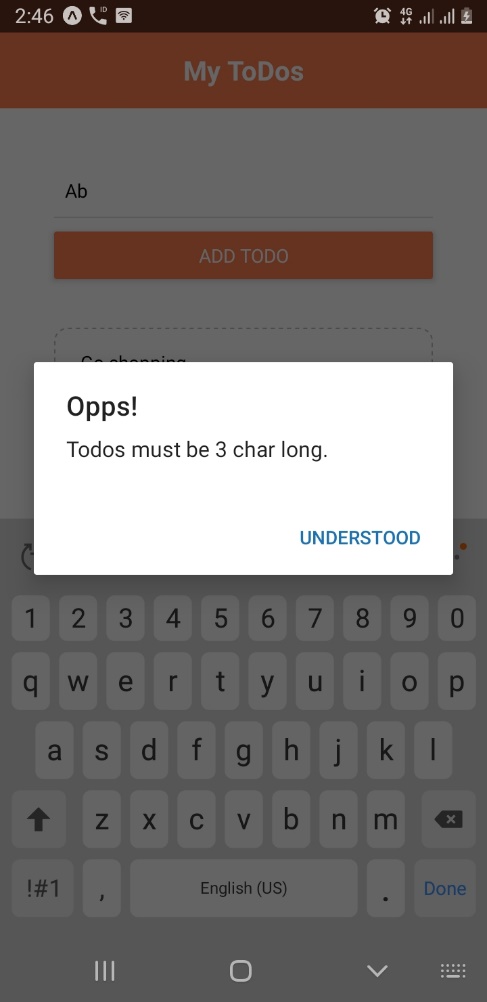
  },

  list: {

    marginTop: 20,

  }

});



## Dismissing the keyboard

<https://www.youtube.com/watch?v=R2eqAgR_KlU&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=14>

App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity, Alert, TouchableWithoutFeedback, Keyboard } from 'react-native';

import Header from "./components/header"

import ToDoitem from "./components/todoitems"

import AddTodo from "./components/addTodo"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switchs', key: '3'},

  ]);

  const pressHandler = (key) => {

    setTodos((prevToDos) => {

      return prevToDos.filter(todo => todo.key != key);

    });

  }

  const submitHandler = (text) => {

    if (text.length > 3){

      setTodos((prevToDos) => {

        return [

          { text: text, key: Math.random().toString()},

          ...prevToDos

        ]

      });

    }

    else{

      Alert.alert("Opps!", "Todos must be 3 char long.", [

        {text:"Understood", onPress: () => console.log("alert closed")}

      ]);

    }

  }

  return (

    <TouchableWithoutFeedback onPress={() => {

      Keyboard.dismiss();

      console.log('dismiss keyboard')

    }}>

      <View style={styles.container}>

        <Header />

        <View style={styles.content}>

          <AddTodo submitHandler={submitHandler}/>

          <View style={styles.list}>

            <FlatList

            data = {todos}

            renderItem = {({ item }) => (

              <ToDoitem item={item} pressHandler={pressHandler} />

            )}

            />

          </View>

        </View>

      </View>

    </TouchableWithoutFeedback>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

  },

  list: {

    marginTop: 20,

  }

});

## FlexBox

<https://www.youtube.com/watch?v=R2eqAgR_KlU&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=14>

Create sandbox.js file in components folder

import React from 'react';

import { StyleSheet, Text, View} from 'react-native';

export default function Sandbox(){

    return (

        <View style={styles.container}>

            <Text style={styles.boxOne}>one</Text>

            <Text style={styles.boxTwo}>two</Text>

            <Text style={styles.boxThree}>three</Text>

            <Text style={styles.boxFour}>four</Text>

        </View>

    )

}

const styles = StyleSheet.create({

    container:{

        // flex:1,

        flexDirection: 'row',

        justifyContent : 'space-around',

        alignItems: "flex-end",

        paddingTop:40,

        backgroundColor:"#ddd"

    },

    boxOne:{

        flex: 1,

        backgroundColor:"violet",

        padding:10

    },

    boxTwo:{

        flex: 2,

        backgroundColor:"gold",

        padding:20

    },

    boxThree:{

        flex: 1,

        backgroundColor:"coral",

        padding:30

    },

    boxFour:{

        flex: 3,

        backgroundColor:"skyblue",

        padding:40

    },

});

App.js

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity, Alert, TouchableWithoutFeedback, Keyboard } from 'react-native';

import Header from "./components/header"

import ToDoitem from "./components/todoitems"

import AddTodo from "./components/addTodo"

import Sandbox from "./components/sandbox"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switchs', key: '3'},

  ]);

  const pressHandler = (key) => {

    setTodos((prevToDos) => {

      return prevToDos.filter(todo => todo.key != key);

    });

  }

  const submitHandler = (text) => {

    if (text.length > 3){

      setTodos((prevToDos) => {

        return [

          { text: text, key: Math.random().toString()},

          ...prevToDos

        ]

      });

    }

    else{

      Alert.alert("Opps!", "Todos must be 3 char long.", [

        {text:"Understood", onPress: () => console.log("alert closed")}

      ]);

    }

  }

  return (

    <Sandbox></Sandbox>

    // <TouchableWithoutFeedback onPress={() => {

    //   Keyboard.dismiss();

    //   console.log('dismiss keyboard')

    // }}>

    //   <View style={styles.container}>

    //     <Header />

    //     <View style={styles.content}>

    //       <AddTodo submitHandler={submitHandler}/>

    //       <View style={styles.list}>

    //         <FlatList

    //         data = {todos}

    //         renderItem = {({ item }) => (

    //           <ToDoitem item={item} pressHandler={pressHandler} />

    //         )}

    //         />

    //       </View>

    //     </View>

    //   </View>

    // </TouchableWithoutFeedback>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

  },

  list: {

    marginTop: 20,

  }

});



App.js – adding flex property

import React, {useState} from 'react';

import { StyleSheet, Text, View, FlatList, TouchableOpacity, Alert, TouchableWithoutFeedback, Keyboard } from 'react-native';

import Header from "./components/header"

import ToDoitem from "./components/todoitems"

import AddTodo from "./components/addTodo"

import Sandbox from "./components/sandbox"

export default function App() {

  const [todos, setTodos] = useState([

    { text: 'buy cofee', key: '1'},

    { text: 'create an app', key: '2'},

    { text: 'play on the switchs', key: '3'},

  ]);

  const pressHandler = (key) => {

    setTodos((prevToDos) => {

      return prevToDos.filter(todo => todo.key != key);

    });

  }

  const submitHandler = (text) => {

    if (text.length > 3){

      setTodos((prevToDos) => {

        return [

          { text: text, key: Math.random().toString()},

          ...prevToDos

        ]

      });

    }

    else{

      Alert.alert("Opps!", "Todos must be 3 char long.", [

        {text:"Understood", onPress: () => console.log("alert closed")}

      ]);

    }

  }

  return (

    // <Sandbox></Sandbox>

    <TouchableWithoutFeedback onPress={() => {

      Keyboard.dismiss();

      console.log('dismiss keyboard')

    }}>

      <View style={styles.container}>

        <Header />

        <View style={styles.content}>

          <AddTodo submitHandler={submitHandler}/>

          <View style={styles.list}>

            <FlatList

            data = {todos}

            renderItem = {({ item }) => (

              <ToDoitem item={item} pressHandler={pressHandler} />

            )}

            />

          </View>

        </View>

      </View>

    </TouchableWithoutFeedback>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

  content:{

    padding: 40,

    // backgroundColor: "pink",

    flex:1,

  },

  list: {

    marginTop: 20,

    // backgroundColor: "yellow",

  }

});

## Icons

<https://www.youtube.com/watch?v=C4ikFaP0a5o&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=15>

<https://expo.github.io/vector-icons/>

1. Import icons

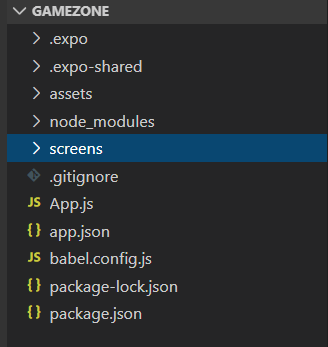
import { Ionicons } from '@expo/vector-icons';

# Review App

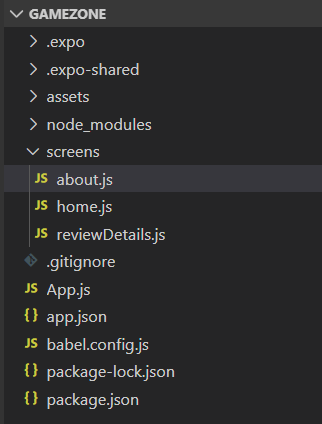
<https://www.youtube.com/watch?v=cFVHTazhb7I&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=16s>

## create base screens

1. Create project gamezone
   1. Open cmd
   2. > expo init gamezone
   3. > cd gamezone
   4. > code .
   5. npm start
2. create new folder – screens inside gamezone



1. create about.js, home.js, reviewDetails.js files inside screens folder



App.js

import React from 'react';

import Home from "./screens/home"

export default function App() {

  return (

    <Home />

  );

}

home.js

import React from 'react';

import {StyleSheet, View, Text} from 'react-native';

export default function Home() {

    return (

        <View style = {styles.container}>

            <Text>

                Home Screen

            </Text>

        </View>

    )

}

const styles = StyleSheet.create({

    container:{

        padding:24,

    }

})

about.js

import React from 'react';

import {StyleSheet, View, Text} from 'react-native';

export default function About() {

    return (

        <View style = {styles.container}>

            <Text>

                About Screen

            </Text>

        </View>

    )

}

const styles = StyleSheet.create({

    container:{

        padding:24,

    }

})

reviewDetails.js

import React from 'react';

import {StyleSheet, View, Text} from 'react-native';

export default function ReviewDetails() {

    return (

        <View style = {styles.container}>

            <Text>

                ReviewDetails Screen

            </Text>

        </View>

    )

}

const styles = StyleSheet.create({

    container:{

        padding:24,

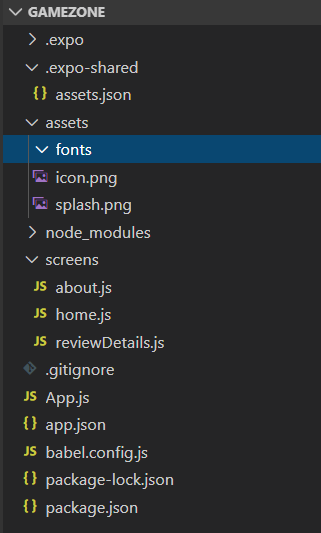
    }

})

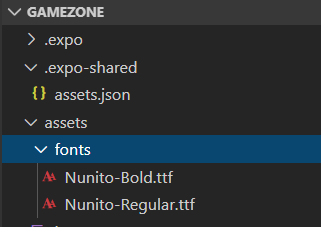
## Custom fonts

<https://www.youtube.com/watch?v=IY5OBeL9LNE&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=17>

1. google fonts: <https://fonts.google.com/>
2. download – nunito
3. create a fonts folder inside asset folder



1. extract nunito.zip file
2. copy “Nunito-Bold”, “Nunito-Regular” into fonts folder



Home.js

import React from 'react';

import {StyleSheet, View, Text} from 'react-native';

export default function Home() {

    return (

        <View style = {styles.container}>

            <Text style={styles.titleText}>

                Home Screen

            </Text>

        </View>

    )

}

const styles = StyleSheet.create({

    container:{

        padding:24,

    },

    titleText:{

        fontFamily:'nunito-bold',

        fontSize:18,

    }

})

App.js

import React, {useState} from 'react';

import Home from "./screens/home"

import \* as Font from 'expo-font';

import {AppLoading} from 'expo'

const getFonts = () => {

  return Font.loadAsync({

    'nunito-regular':require('./assets/fonts/Nunito-Regular.ttf'),

    'nunito-bold':require('./assets/fonts/Nunito-Bold.ttf'),

  });

}

export default function App() {

  const [fontsLoaded, setFontsLoaded] = useState(false);

  if (fontsLoaded){

    return (

      <Home />

    );

  }else{

    return(

      <AppLoading

        startAsync={getFonts}

        onFinish={()=>setFontsLoaded(true)}/>

    );

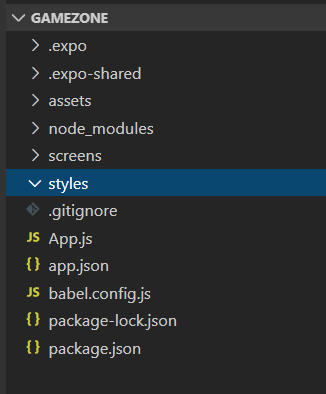
  }

}

## Global styles

<https://www.youtube.com/watch?v=wtvpQ9liu4g&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=18>

1. create styles folder in main directory



1. create new file global.js inside styles folder

import {StyleSheet} from 'react-native';

export const GlobalStyles = StyleSheet.create({

    container:{

        flex:1,

        padding:20,

    },

    titleText:{

        fontFamily:'nunito-bold',

        fontSize:18,

        color:"#333",

    },

    paragraph:{

        marginVertical:8,

        lineHeight: 20,

    }

})

Home.js

import React from 'react';

import {StyleSheet, View, Text} from 'react-native';

import { GlobalStyles } from '../styles/global'

export default function Home() {

    return (

        <View style = {GlobalStyles.container}>

            <Text style={GlobalStyles.titleText}>

                Home Screen

            </Text>

        </View>

    )

}

## Navigation setup

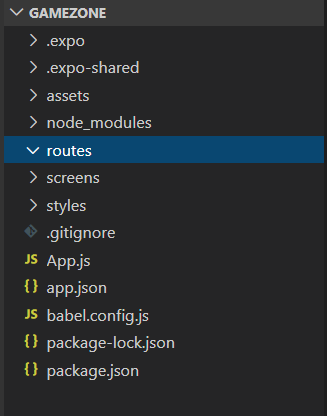
<https://www.youtube.com/watch?v=OmQCU-3KPms&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=19>

1. install react navigation
   1. > npm install react-navigation
   2. > expo install react-native-gesture-handler react-native-reanimated

## Stack navigator

<https://www.youtube.com/watch?v=cS4PgI3zBzY&list=PL4cUxeGkcC9ixPU-QkScoRBVxtPPzVjrQ&index=20>

1. Install react navigation stack
   1. > npm install react-navigation-stack
2. Create new folder routes in project directory



1. Create homeStack.js in routes folder

homeStack.js

import {createStackNavigator} from 'react-navigation-stack';

import {createAppContainer} from 'react-navigation';

import Home from '../screens/home';

import ReviewDetails from '../screens/reviewDetails';

const screens = {

    Home : {

        screen: Home

    },

    ReviewDetails : {

        screen : ReviewDetails

    }

}

const HomeStack = createStackNavigator(screens);

export default createAppContainer(HomeStack);

App.js

import React, {useState} from 'react';

import Home from "./screens/home";

import \* as Font from 'expo-font';

import {AppLoading} from 'expo';

import Navigator from './routes/homeStack';

const getFonts = () => {

  return Font.loadAsync({

    'nunito-regular':require('./assets/fonts/Nunito-Regular.ttf'),

    'nunito-bold':require('./assets/fonts/Nunito-Bold.ttf'),

  });

}

export default function App() {

  const [fontsLoaded, setFontsLoaded] = useState(false);

  if (fontsLoaded){

    return (

      <Navigator />

    );

  }else{

    return(

      <AppLoading

        startAsync={getFonts}

        onFinish={()=>setFontsLoaded(true)}/>

    );

  }

}