CLOUD CAMPUS

REACT NATIVE

SAIKOU BALDÉ CONSULTANT DEV FULLSTACK

OBJECTIF

- MAÎTRISER L'ENVIRONNEMENT DE DEV REACT NATIVE
- ZOOM FONCTIONNALITÉES FRONT
- LA PRATIQUE DU DEV REACT NATIVE
- PUBLICATION SUR LES PLATEFORMES

LES GRANDES LIGNES

ENV DEV POUR DÉMARRER UN PROJET

LES PLUGINS TIERS DANS REACT NATIVE

HELLO WORLD APP ET LE DEVICE

UTILISATION DE REDUX DANS REACT NATIVE

ZOOM SUR LES FONCTIONNALITÉS

OPTIMISATION ET MISE EN PROD

CAS PRATIQUES - FIL ROUGE PROJET

REACT NATIVE

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV

React Native *permet aux développeurs maîtrisant React de créer des applications natives. Dans le même temps, les développeurs android et ios natifs peuvent utiliser React Native pour obtenir la parité entre les plates-formes natives en écrivant une seule fois des fonctionnalités communes.*

La meilleure façon d'utiliser React Native est de l'utiliser avec un Framework du style Expo même si l'on peut l'utiliser sans

https://docs.expo.dev/get-started/introduction/

https://reactnative.dev/docs/environment-setup

https://reactnative.dev/docs/set-up-your-environment

https://reactnative.dev/docs/getting-started-without-a-framework

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

EXPO

Expo est un framework qui facilite le développement d'applications Android et iOS. EXPO fournit un routage basé sur des fichiers, une bibliothèque standard de modules natifs et bien plus encore. Expo est open source avec une communauté active sur GitHub et Discord.

Beaucoup d'applications célèbres ont été développées avec Expo, parmi lesquelles

- ***Facebook** : l'application mobile de Facebook*
- ***Uber Eats** : l'application de livraison de repas d'Uber*
- ***AirBnB** : l'application mobile d'AirBnB*
- ***TikTok** : l'application mobile de TikTok*

https://docs.expo.dev/get-started/introduction/

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

Prérequis

- ***Node.js*** *: Assurez-vous que Node.js est installé sur votre machine. Vous pouvez le télécharger depuis nodejs.org.*
- ***npm*** *: npm est inclus avec Node.js, donc si vous avez installé Node.js, vous devez déjà avoir npm.*
- ***Expo CLI*** *: Vous aurez besoin d'Expo CLI pour créer et gérer vos projets Expo*
- ***macOS***, ***Windows*** *(Powershell et WSL 2) et* ***Linux*** *sont pris en charge par Expo pour la créa d'appli React Native.*

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ node -v
v18.19.1
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npm -v
9.2.0
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx -v
9.2.0
```

https://reactnative.dev/docs/environment-setup

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

Création de Projet

Nous vous recommandons de commencer par le projet par défaut créé par create-expo-app. Le projet par défaut inclut un exemple de code pour vous aider à démarrer.

Pour créer un nouveau projet, exécutez la commande suivante :

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx create-expo-app@latest
Need to install the following packages:
  create-expo-app@2.3.5
Ok to proceed? (y) y
✔ What is your app named? … cloud-capus-react-native
✔ Downloaded and extracted project files.
> npm install
```

```
added 1545 packages, and audited 1546 packages in 52s

147 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd cloud-capus-react-native
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
```

NB *: Le modèle de projet par défaut inclut la bibliothèque Expo Router installée pour créer des applications multi-écrans avec Expo. Vous pouvez choisir un modèle différent en utilisant l'option --template. Par exemple, en utilisant cette option, vous pouvez choisir le modèle vierge “BLANK” pour créer un projet avec le minimum de bibliothèques requises installées et sans navigation configurée.*

https://reactnative.dev/docs/environment-setup

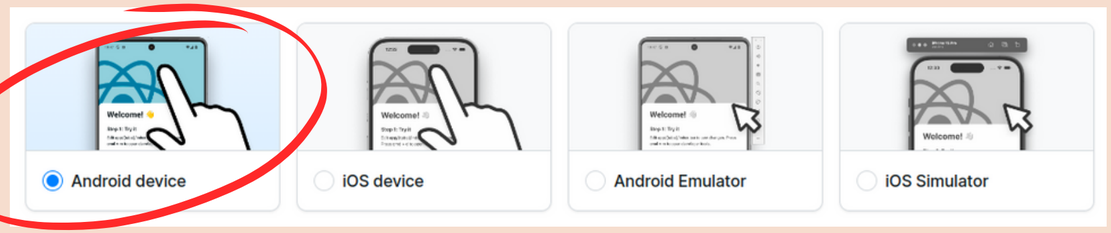
NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

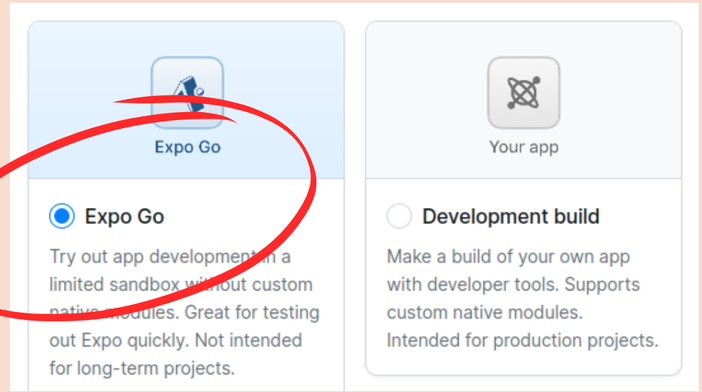
Configurez votre environnement

Configurons ensemble un environnement de développement local pour exécuter votre projet sur Android et iOS.

1 *Dans quel environnement faites vous du dev ?*



2 *Quel outils souhaitez vous utiliser pour le dev ?*



3 *Configurer un appareil Android avec Expo Go*

Scannez le code QR pour télécharger l'application depuis le Google Play Store, ou visitez la page Expo Go sur le Google Play Store.

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI (METRO BUNDLER)

Commencez à développer

Démarrer un serveur de développement :

Pour démarrer le serveur de développement, exécutez la commande suivante :

Ouvrez l'application sur votre appareil ""Mobile

Après avoir exécuté la commande ci-dessus, vous verrez un code QR dans votre terminal. Scannez ce code QR pour ouvrir l'application sur votre appareil.

Si vous utilisez un émulateur Android ou un simulateur iOS, vous pouvez appuyer respectivement sur a ou i pour ouvrir l'application.

https://reactnative.dev/docs/environment-setup

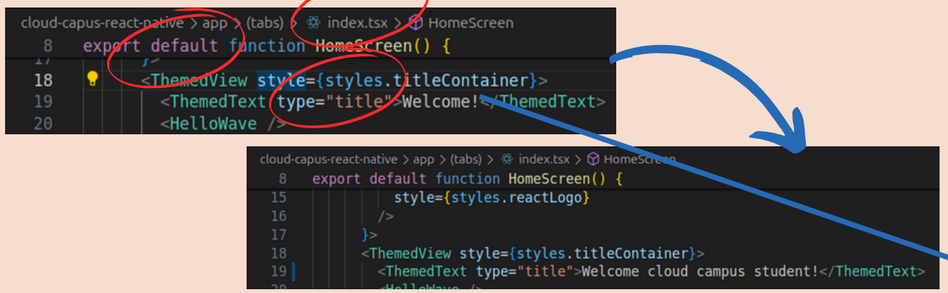
NPX ET CREATE EXPO APP

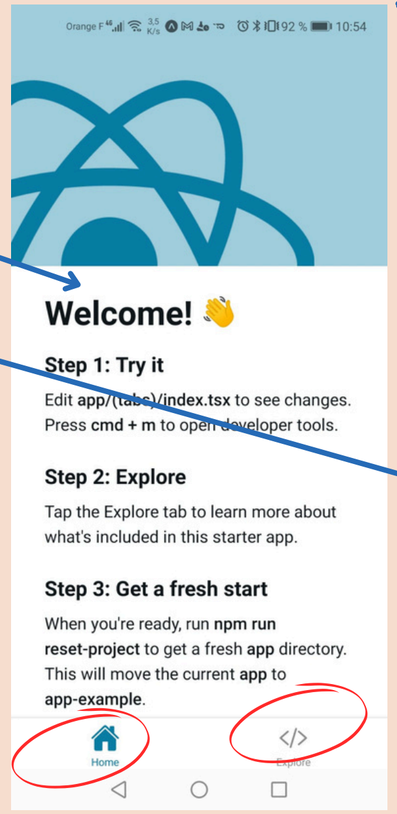
MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

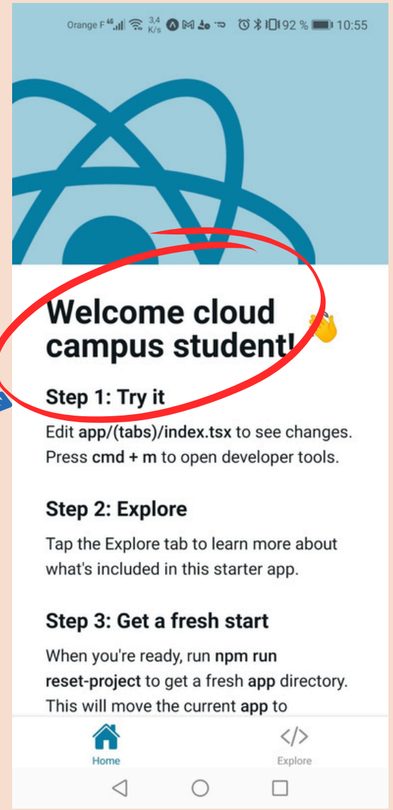
Commencez à développer

Effectuez votre premier changement

Ouvrez le fichier app/(tabs)/index.tsx dans votre éditeur de code et apportez une modification.







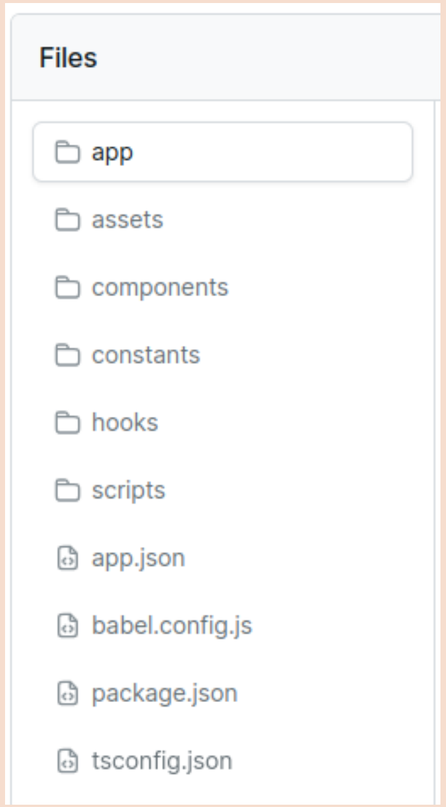
https://reactnative.dev/docs/environment-setup

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

Structure du fichier

Ci-dessous, vous pouvez vous familiariser avec la structure des fichiers du projet par défaut :



app *contient la navigation de l'application, basée sur des fichiers.*
L'application dispose de deux itinéraires définis par deux fichiers : ***app/(tabs)/index.tsx et app/(tabs)/explore.tsx.*** *Le fichier de mise en page dans* ***app/(tabs)/_layout.tsx*** *configure le navigateur d'onglets.*

assets *contient* ***adaptive-icon.png*** *utilisé pour Android et un* ***icon.png*** *utilisé pour iOS comme icônes d'application. Il contient également* ***splash.png*** *qui est une image pour l'écran de démarrage du projet et un* ***favicon.png*** *si l'application s'exécute dans un navigateur.*

components *contient des composants React Native, comme ThemedText.tsx, qui crée des éléments de texte qui utilisent le jeu de couleurs de l'application en modes clair et sombre.*

constants *Contient Colors.ts, qui contient une liste de valeurs de couleur dans toute l'application.*

hooks *contient des React Hooks, qui permettent de partager un comportement commun entre les composants. Par exemple, useThemeColor() est un hook qui renvoie une couleur basée sur le thème actuel.*

scripts *contient reset-project.js, qui peut être exécuté avec npx run reset-project. Ce script déplacera le répertoire de l'application vers app-example et créera un nouveau répertoire d'application avec un fichier index.tsx.*

app.json *contient les options de configuration du projet et est appelé configuration de l'application. Ces options modifient le comportement de votre projet lors du développement, de la création, de la soumission et de la mise à jour de votre application.*

babel.config.js *Applique le préréglage babel-preset-expo qui étend le préréglage React Native par défaut et ajoute la prise en charge des décorateurs, des packages Web d'arborescence et le chargement d'icônes de police avec des dépendances natives facultatives si elles sont installées.*

package.json *Le fichier package.json contient les dépendances, les scripts et les métadonnées du projet. Chaque fois qu'une nouvelle dépendance est ajoutée à votre projet, elle sera ajoutée à ce fichier.*

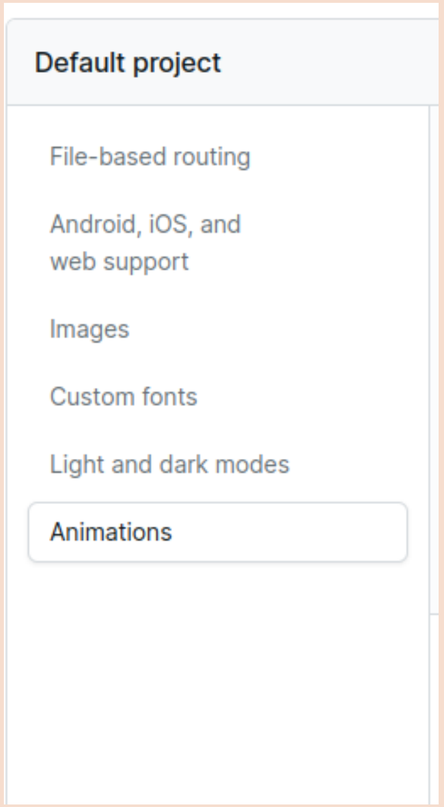
tsconfig.json *contient les règles que TypeScript utilisera pour appliquer la sécurité des types tout au long du projet.*

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

Fonctionnalités du template

Le modèle de projet par défaut présente les fonctionnalités suivantes :



File-based routing *L'application dispose de* ***deux écrans*** *:* ***app/(tabs)/index.tsx*** *et* ***app/(tabs)/explore.tsx.*** *Le fichier de mise en page dans* ***app/(tabs)/_layout.tsx*** *configure le navigateur d'onglets.*

Android, iOS, and web support *Vous pouvez ouvrir ce projet sur Android, iOS et le Web. Pour ouvrir la version Web, appuyez sur w dans le terminal après avoir exécuté le projet.*

Images *Pour les images statiques, vous pouvez utiliser les suffixes @2x et @3x pour fournir des images pour différentes densités d'écran.*

Custom fonts *Ouvrez app/_layout.tsx pour voir comment charger des polices personnalisées.*

Light and dark modes *Ce modèle prend en charge les modes clair et sombre. Le hook useColorScheme() vous permet d'inspecter le jeu de couleurs actuel de l'utilisateur, afin que vous puissiez ajuster les couleurs de l'interface utilisateur en conséquence.Apprendre encore plus*

Animations *Le composant composants/HelloWave.tsx utilise l'API Animated pour créer une animation de main agitante. Sur iOS, le composant/ParallaxScrollView.tsx crée un effet de parallaxe pour l'image d'en-tête.*

https://reactnative.dev/docs/environment-setup

NPX ET CREATE EXPO APP

MISE EN PLACE D'UN ENVIRONNEMENT DE DEV - EXPO CLI

Réinitialisez votre projet

Vous pouvez supprimer le code passe-partout et repartir à zéro avec un nouveau projet. Exécutez la commande suivante pour réinitialiser votre projet :Terminal

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/cloud-capus-react-native$ tree -L 1
.
├── app
├── app.json
├── assets
├── babel.config.js
├── components
├── constants
├── expo-env.d.ts
├── hooks
├── node_modules
├── package.json
├── package-lock.json
├── README.md
├── scripts
└── tsconfig.json

8 directories, 7 files
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/cloud-capus-react-native$ npm run reset-project

> cloud-capus-react-native@1.0.0 reset-project
> node ./scripts/reset-project.js

/app moved to /app-example.
New /app directory created.
app/index.tsx created.
app/_layout.tsx created.
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/cloud-capus-react-native$ tree -L 1
.
├── app
├── app-example
├── app.json
├── assets
├── babel.config.js
├── components
├── constants
├── expo-env.d.ts
├── hooks
├── node_modules
├── package.json
├── package-lock.json
├── README.md
├── scripts
└── tsconfig.json

9 directories, 7 files
```

cloud-capus-react-native > {} package.json > ...

```
{
  "name": "cloud-capus-react-native",
  "main": "expo-router/entry",
  "version": "1.0.0",
  ▷ Debug
  "scripts": {
    "start": "expo start",
    "reset-project": "node ./scripts/reset-project.js",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web",
    "test": "jest --watchAll",
    "lint": "expo lint"
  },
```

Il ne nous reste plus qu'a créer et développer des appli après avoir mis en place l'environnement de dev

https://reactnative.dev/docs/environment-setup

NPX ET CREATE EXPO APP

UTILISATION DE EXPO CLI DANS REACT NATIVE - AWESOMEPROJECT

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx create-expo-app AwesomeProject
Need to install the following packages:
  create-expo-app@2.3.5
Ok to proceed? (y) y
✔ Downloaded and extracted project files.
> npm install
npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.18.6: This proposal has been merged to the
```

```
added 1524 packages, and audited 1525 packages in 1m

138 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd AwesomeProject
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development
 without a Mac
- npm run web
```

NPX ET CREATE EXPO APP

UTILISATION DE EXPO CLI DANS REACT NATIVE - AWESOMEPROJECT

1 Start a development server

To start the development server, run the following command:

Terminal — Copy

```
› npx expo start
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/AwesomeProject$ npx expo start
Starting project at /home/balde/www/react_native/AwesomeProject
Starting Metro Bundler
The following packages should be updated for best compatibility with the installed expo version:
  expo@51.0.2 - expected version: ~51.0.4
  expo-font@12.0.4 - expected version: ~12.0.5
  expo-router@3.5.11 - expected version: ~3.5.12
Your project may not work correctly until you install the expected versions of the packages.

› Metro waiting on exp://10.16.185.152:8081
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Web is waiting on http://localhost:8081
```

2 Open the app on your device

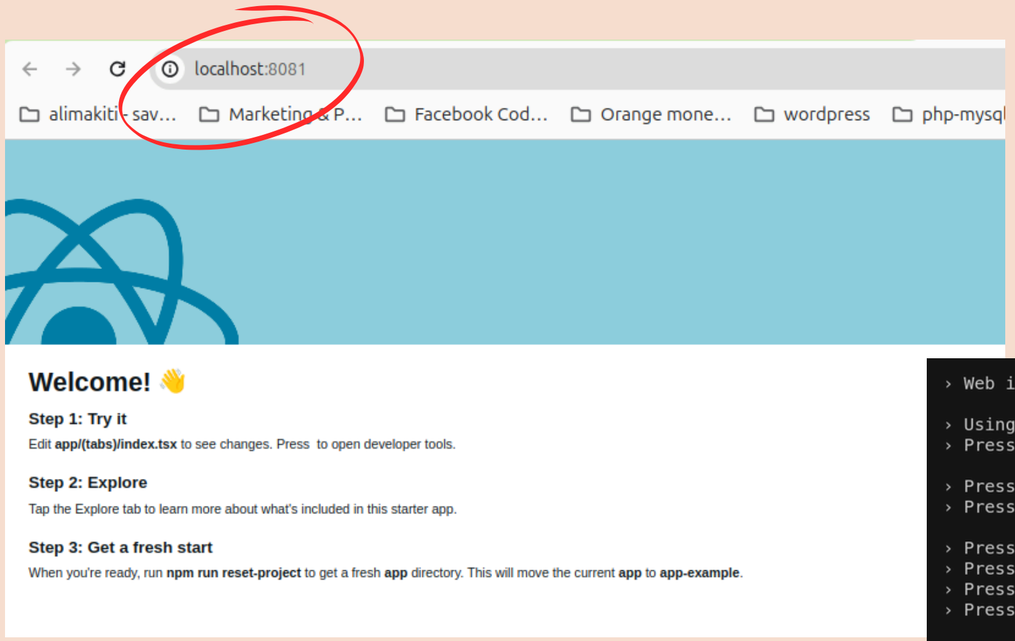
After running the command above, you will see a QR code in your terminal. Scan this QR code to open the app on your device.

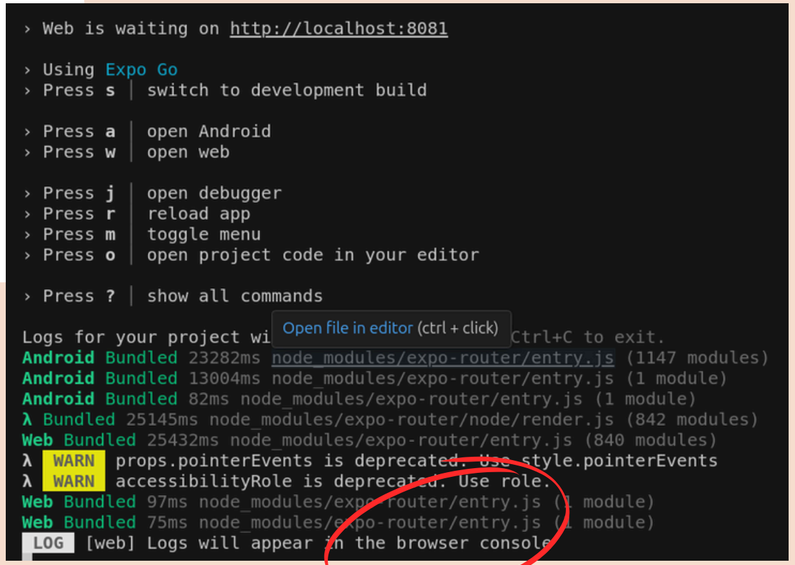
If you're using an Android Emulator or iOS Simulator, you can press `a` or `i` respectively to open the app.

▸ Having problems?

NPX ET CREATE EXPO APP

UTILISATION DE EXPO CLI DANS REACT NATIVE - AWESOMEPROJECT

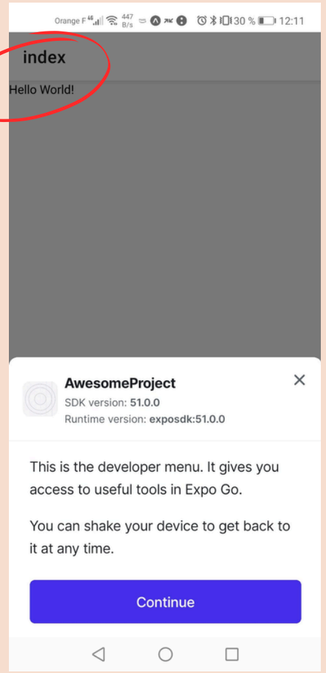


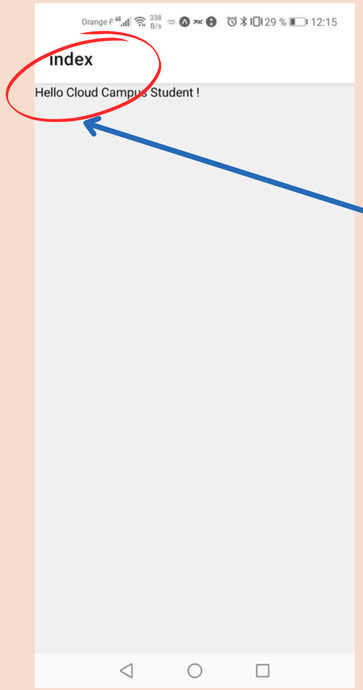


NPX ET CREATE EXPO APP

UTILISATION DE EXPO CLI DANS REACT NATIVE - AWESOMEPROJECT

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/AwesomeProject$ ls
app          app.json  babel.config.js  constants      hooks         package.json       README.md  tsconfig.json
app-example  assets    components       expo-env.d.ts  node_modules  package-lock.json  scripts
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/AwesomeProject$
```





home > balde > www > react_native > AwesomeProject > app > index.tsx > Index

```tsx
import {ScrollView, Text, View } from "react-native";

export default function Index() {
  return (
    <ScrollView contentInsetAdjustmentBehavior="automatic">
      <View>
        <Text>Hello World!</Text>
      </View>
    </ScrollView>
  );
}
```

home > balde > www > react_native > AwesomeProject > app > index.tsx > Index

```tsx
import {ScrollView, Text, View } from "react-native";

export default function Index() {
  return (
    <ScrollView contentInsetAdjustmentBehavior="automatic">
      <View>
        <Text>Hello Cloud Campus Student !</Text>
      </View>
    </ScrollView>
  );
}
```

NPX ET CREATE EXPO APP

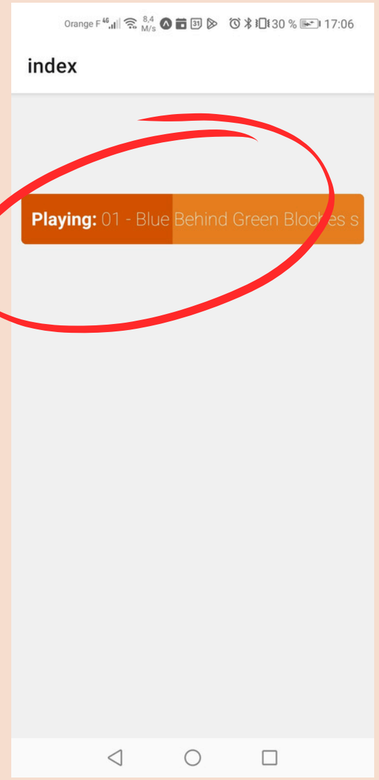
UTILISATION DE EXPO CLI DANS REACT NATIVE - AWESOMEPROJECT

home > balde > www > react_native > AwesomeProject > app > index.tsx > Index > styles

```tsx
import React from 'react';
import {StyleSheet, Text, View } from "react-native";

export default function Index() {
    const name = '01 - Blue Behind Green Bloches s';
    const styles = StyleSheet.create({
      container: {
        margin: 10,
        marginTop: 100,
        backgroundColor: '#e67e22',
        borderRadius: 5,
      },
      innerContainer: {
        backgroundColor: '#d35400',
        height: 50,
        width: 150,
        borderTopLeftRadius: 5,
        borderBottomLeftRadius: 5,
      },
      title: {
        fontSize: 18,
        fontWeight: '200',
        color: '#fff',
        position: 'absolute',
        backgroundColor: 'transparent',
        top: 12,
        left: 10,
      },
      subtitle: {
        fontWeight: 'bold',
      },
    });
```

home > balde > www > react_native > AwesomeProject > app > index.tsx > Index > styles

```tsx
export default function Index() {
    const styles = StyleSheet.create({
      },
    });

  return (
    <View style={styles.container}>
      <View style={styles.innerContainer} />
      <Text style={styles.title}>
        <Text style={styles.subtitle}>Playing:</Text> {name}
      </Text>
    </View>
  );
}
```



NPX ET CREATE EXPO APP

TEMPLATE BLANK (TYPESCRIPT - EXPO CLI - REACT NATIVE)

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx create-expo-app --template
✔ Choose a template: › Blank (TypeScript)
✔ What is your app named? … expo-template-app
✔ Downloaded and extracted project files.
> npm install
```

```
added 1192 packages, and audited 1193 packages in 36s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd expo-template-app
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS develo
- npm run web
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ cd expo-template-app/
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ ls
app.json  App.tsx  assets  babel.config.js  node_modules  package.json  package-lock.json  tsconfig.json
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npm start
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npm start

> expo-template-app@1.0.0 start
> expo start

Starting project at /home/balde/www/react_native/expo-template-app
Starting Metro Bundler
The following packages should be updated for best compatibility with the installed expo version:
  typescript@5.4.5 - expected version: ~5.3.3
Your project may not work correctly until you install the expected versions of the packages.

› Metro waiting on exp://10.16.185.152:8081
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
```

```
› Using Expo Go
› Press s | switch to development build

› Press a | open Android
› Press w | open web

› Press j | open debugger
› Press r | reload app
› Press m | toggle menu
› Press o | open project code in your editor

› Press ? | show all commands
```

NPX ET CREATE EXPO APP

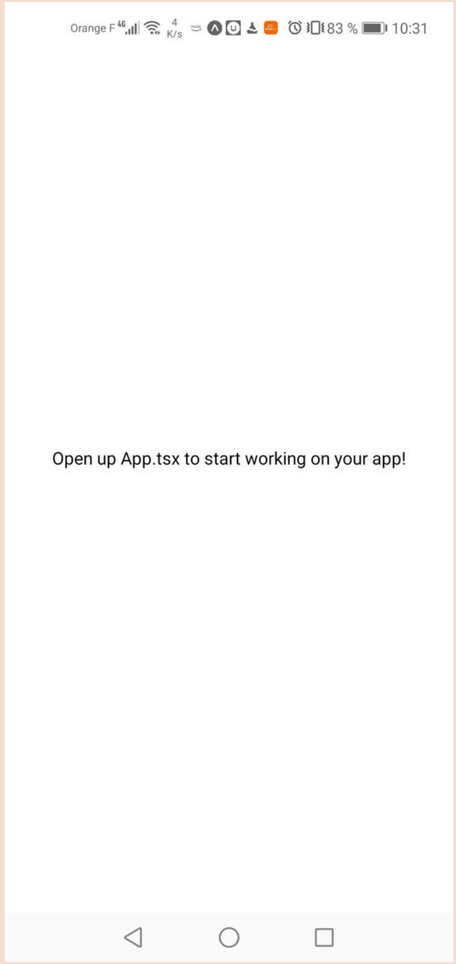
TEMPLATE BLANK (TYPESCRIPT - EXPO CLI - REACT NATIVE)

App.tsx

home > balde > www > react_native > expo-template-app > App.tsx > styles

```tsx
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text>Open up App.tsx to start working on your app!</Text>
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

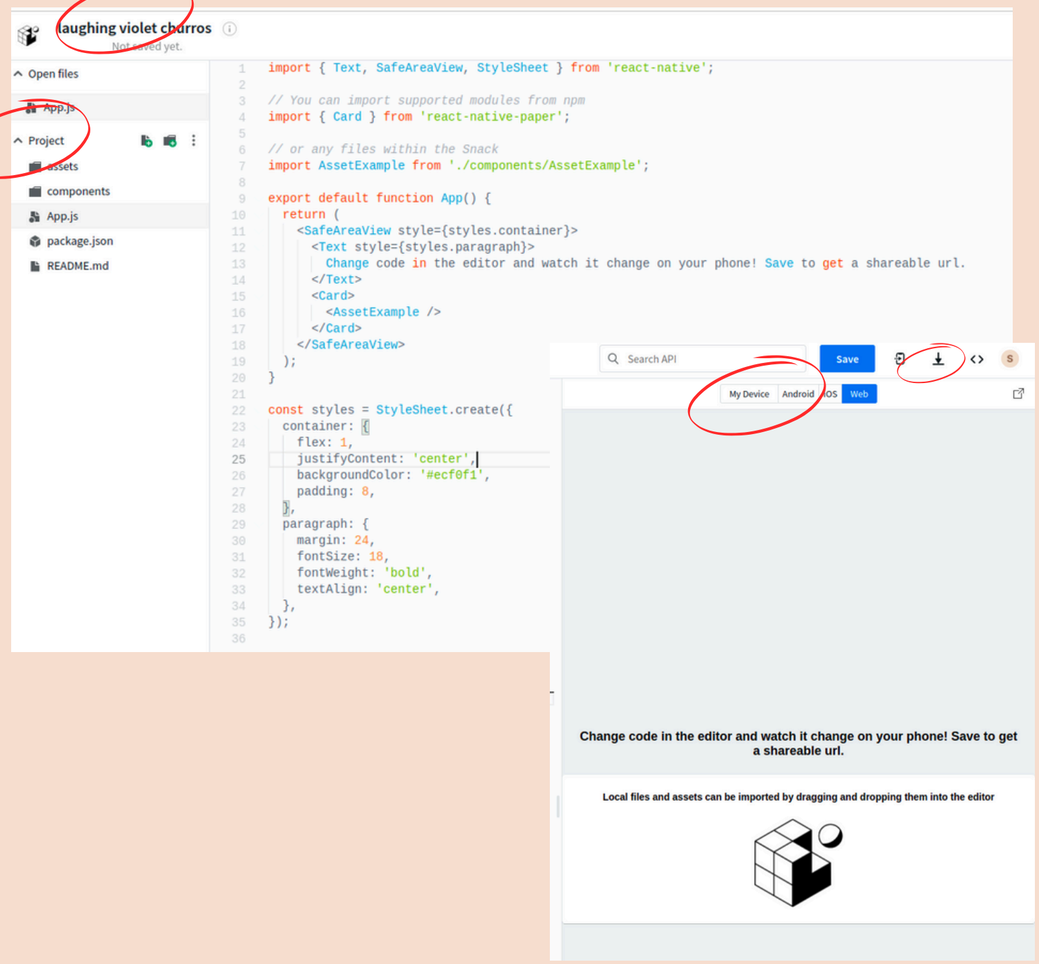
```tsx
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#fff",
    alignItems: "center",
    justifyContent: "center",
  },
  text: { fontWeight: "bold" },
});
```

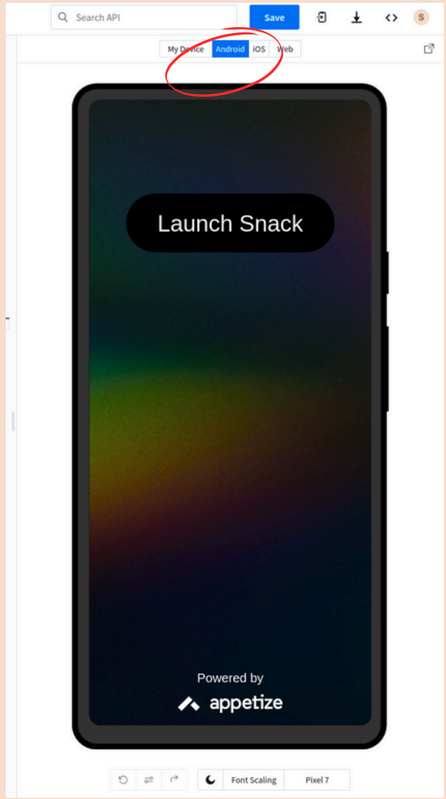


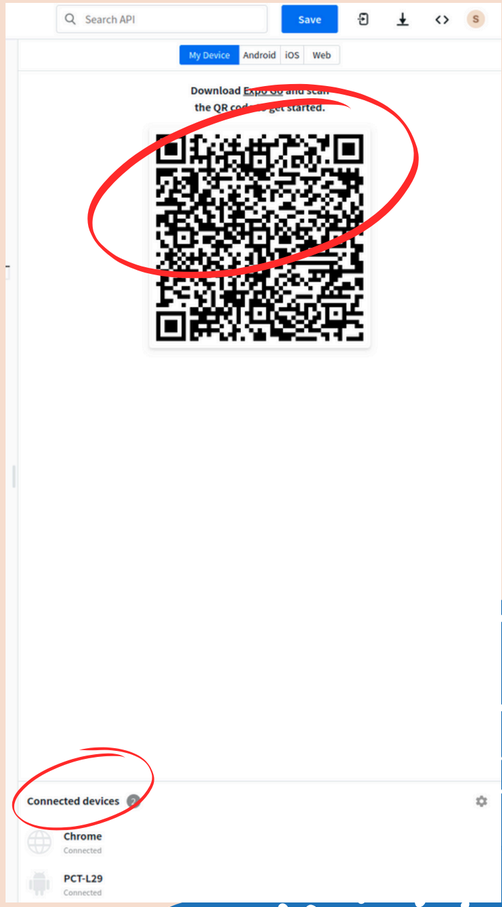
NPX ET CREATE EXPO APP

L'OUTILS SNACK EXPO - DÉVELOPPER DANS UN ENV CLOUD

Expo Snack *est un outil en ligne très pratique pour développer et tester des applications React Native directement dans votre navigateur sans avoir besoin de configurer un environnement de développement complet.*



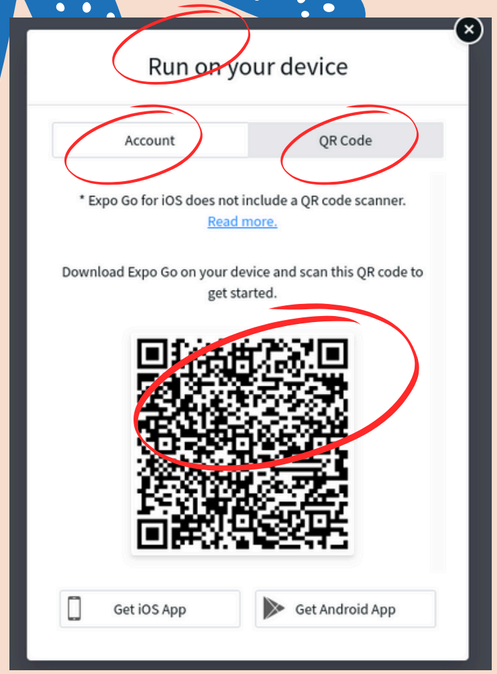


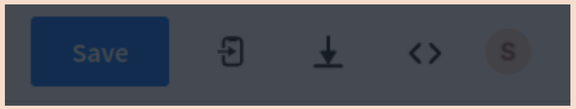


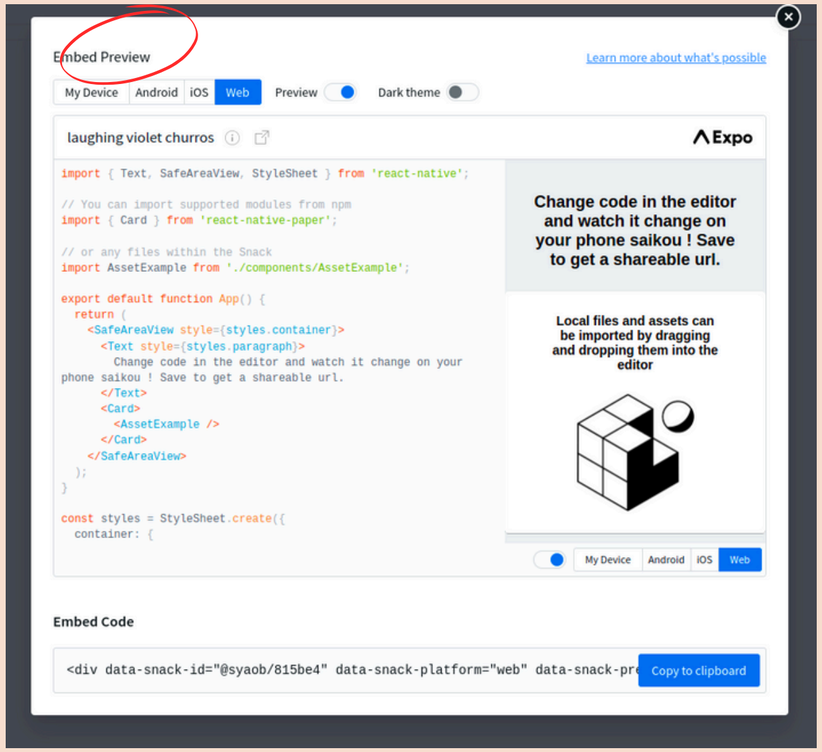
👉 ***https://snack.expo.dev/***

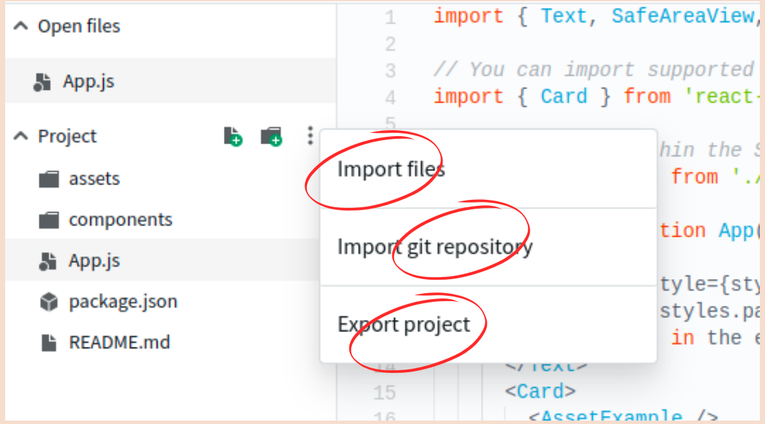
NPX ET CREATE EXPO APP

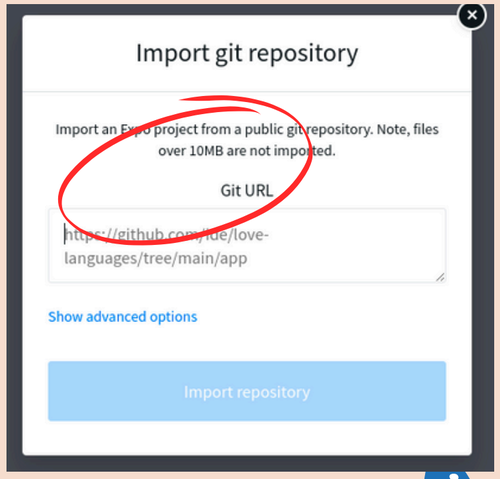
L'OUTILS SNACK EXPO





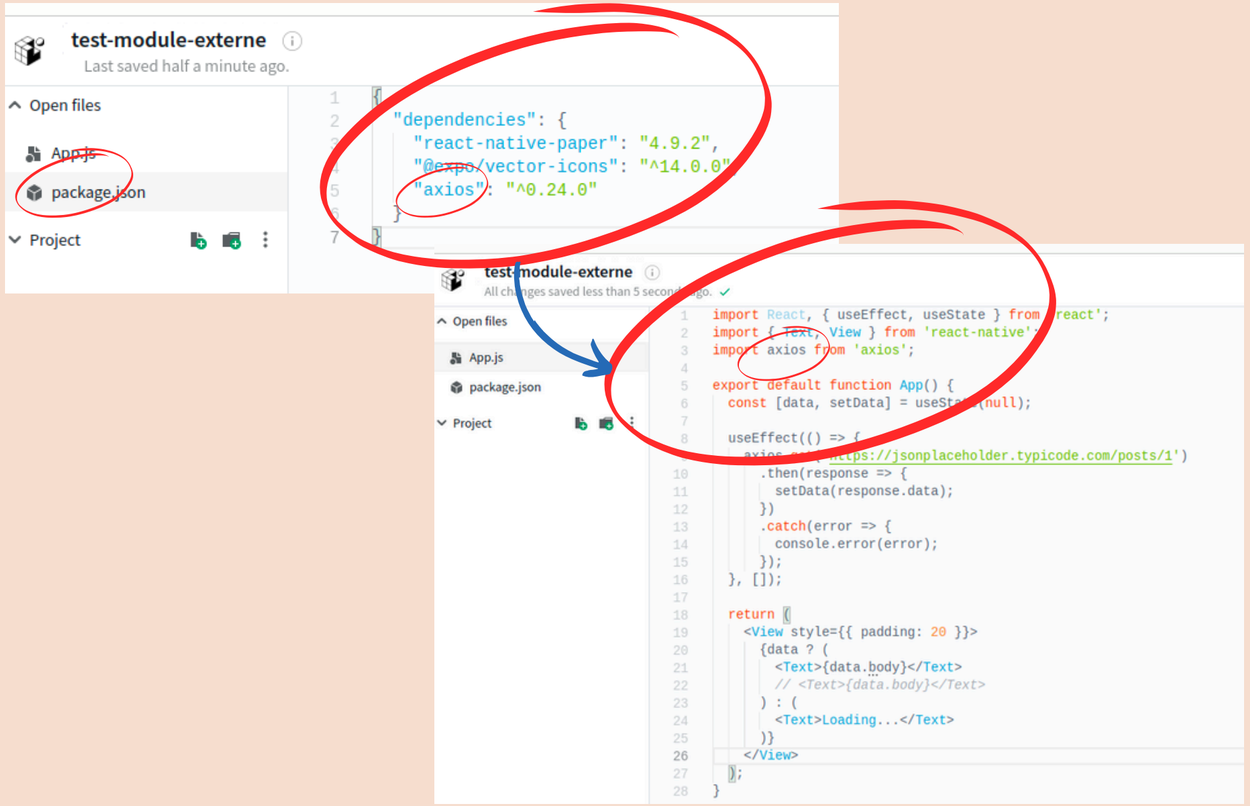






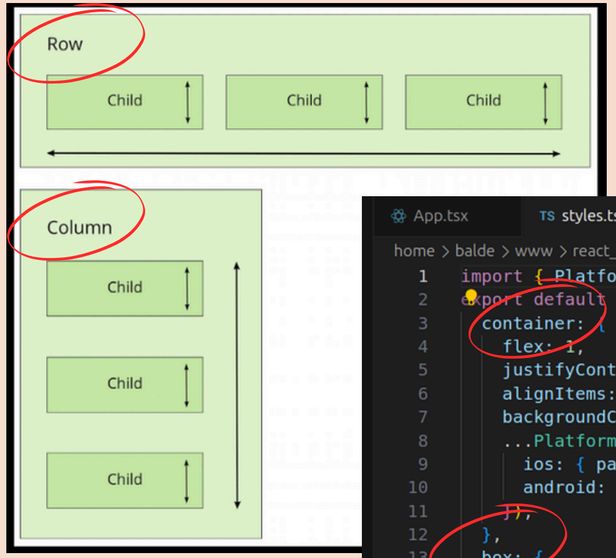
NPX ET CREATE EXPO APP

L'OUTILS SNACK EXPO



NPX ET CREATE EXPO APP

LE STYLE (SIMPLEMENT) DANS REACT NATIVE

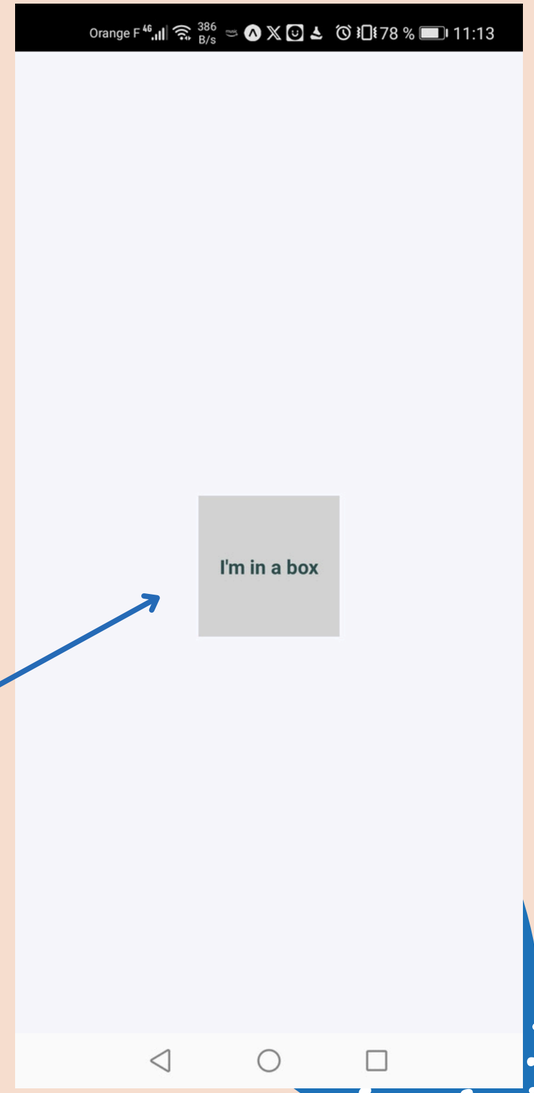


App.tsx styles.ts

home > balde > www > react_native > expo-template-app > styles.ts > ...

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
    ...Platform.select({
      ios: { paddingTop: 20 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },
  box: {
    width: 100,
    height: 100,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "lightgray",
  },
  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },
});
```

```tsx
import React from "react";
import { Text, View } from "react-native";
import styles from "./styles";
export default function App() {
  return (
    <View style={styles.container}>
      <View style={styles.box}>
        <Text style={styles.boxText}>I'm in a box</Text>
      </View>
    </View>
  );
}
```



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

MISE EN PAGE SIMPLE À TROIS COLONNES

App.tsx | styles.ts | package.json

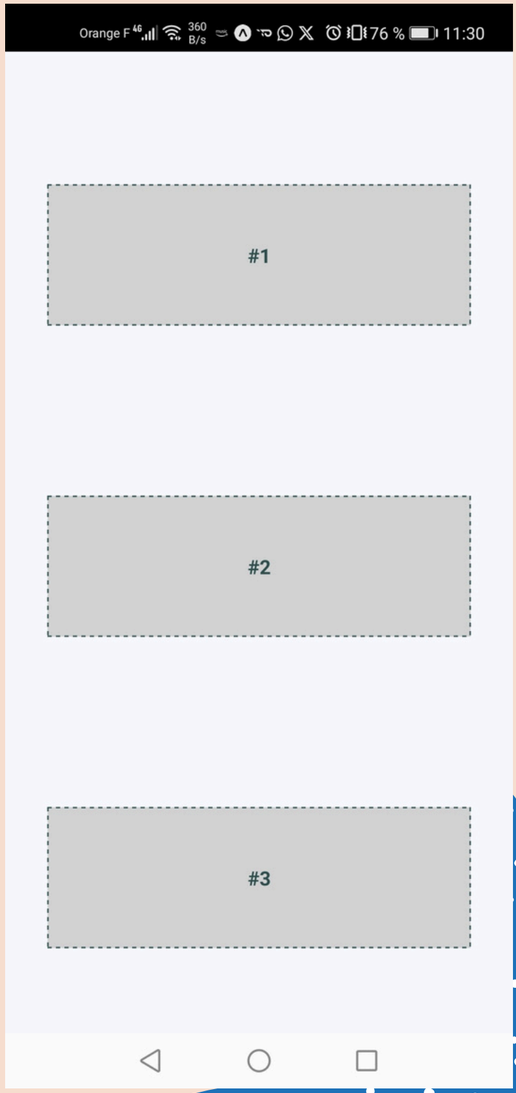
home > balde > www > react_native > expo-template-app > App.tsx > App

```tsx
import React from "react";
import { Text, View } from "react-native";
import styles from "./styles";
export default function App() {
  return (
    <View style={styles.container}>
      <View style={styles.box}>
        <Text style={styles.boxText}>#1</Text>
      </View>
      <View style={styles.box}>
        <Text style={styles.boxText}>#2</Text>
      </View>
      <View style={styles.box}>
        <Text style={styles.boxText}>#3</Text>
      </View>
    </View>
  );
}
```

App.tsx | styles.ts | package.json

home > balde > www > react_native > expo-template-app > styles.ts > default > box

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    alignItems: "center",
    justifyContent: "space-around",
    backgroundColor: "ghostwhite",
    ...Platform.select({
      ios: { paddingTop: 20 },
      android: { paddingTop: StatusBar.currentHeight }
    })
  },
  box: {
    width: 300,
    height: 100,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "lightgray",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray"
  },
  boxText: {
    color: "darkslategray",
    fontWeight: "bold"
  }
});
```



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

AMELIORER LA MISE EN PAGE SIMPLE À TROIS COLONNES

App.tsx | styles.ts | package.json

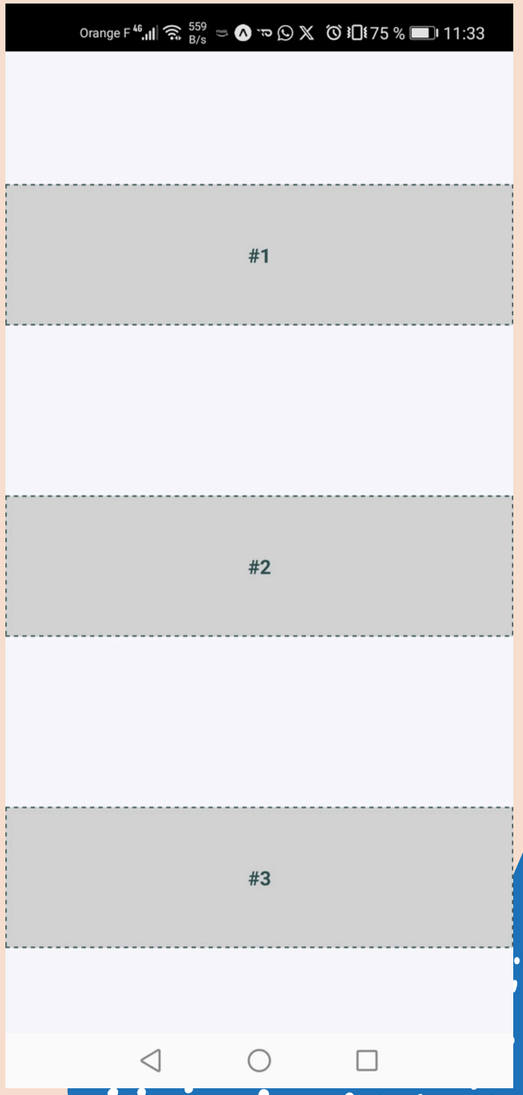
home > balde > www > react_native > expo-template-app > styles.ts > ...

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    backgroundColor: "ghostwhite",
    justifyContent: "space-around",
    ...Platform.select({
      ios: { paddingTop: 20 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },
  box: {
    height: 100,
    justifyContent: "center",
    alignSelf: "stretch",
    alignItems: "center",
    backgroundColor: "lightgray",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
  },
  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },
});

```

*Le changement clé ici est la propriété **alignSelf**. Cela indique aux éléments avec le style box de modifier leur largeur ou leur hauteur (en fonction du flexDirection de leur conteneur) pour remplir l'espace. De plus, le style de boîte ne définit plus de propriété de largeur car celle-ci sera désormais calculée à la volée.*

***justifyContent** contrôle l'alignement des éléments le long de l'axe principal (horizontal ou vertical selon flexDirection).*

***alignItems** contrôle l'alignement des éléments le long de l'axe secondaire (perpendiculaire à l'axe principal).*



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

MISE EN PAGE SIMPLE À TROIS COLONNES => OPTIMISATION DU CODE

App.tsx | Box.js | styles.ts

home > balde > www > react_native > expo-template-app > components > Box.js

```
1  import React from "react";
2  import { PropTypes } from "prop-types";
3  import { View, Text } from "react-native";
4  import styles from "../styles";
5  export default function Box({ children }) {
6    return (
7      <View style={styles.box}>
8        <Text style={styles.boxText}>{children}</Text>
9      </View>
10   );
11 }
12 Box.propTypes = {
13   children: PropTypes.node.isRequired,
14 };
```

App.tsx | Box.js | styles.ts

home > balde > www > react_native > expo-template-app > App.tsx > App

```
1  import React from "react";
2  import { Text, View } from "react-native";
3  import styles from "./styles";
4  import Box from "./components/Box";
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        {/* <View style={styles.box}>
9          <Text style={styles.boxText}>#1</Text>
10       </View> */}
11       <Box>
12         #1
13       </Box>
14       <View style={styles.box}>
15         <Text style={styles.boxText}>#2</Text>
16       </View>
17       <View style={styles.box}>
18         <Text style={styles.boxText}>#3</Text>
19       </View>
20     </View>
21   );
22 }
```

App.tsx | Box.js | styles.ts

home > balde > www > react_native > expo-template-app > App.tsx > App

```
1  import React from "react";
2  import { Text, View } from "react-native";
3  import styles from "./styles";
4  import Box from "./components/Box";
5  export default function App() {
6    return (
7      <View style={styles.container}>
8        <Box>
9          #1
10       </Box>
11       <Box>
12         #2
13       </Box>
14       <Box>
15         #3
16       </Box>
17     </View>
18   );
19 }
```

NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

LIGNES FLEXIBLES : ROWS

App.tsx | styles.ts | package.json

home > balde > www > react_native > expo-template-app > styles.ts > ...

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    backgroundColor: "ghostwhite",
    justifyContent: "space-around",
    ...Platform.select({
      ios: { paddingTop: 20 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },
  box: {
    height: 100,
    justifyContent: "center",
    alignSelf: "stretch",
    alignItems: "center",
    backgroundColor: "lightgray",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
  },
  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },
});
```

ne > balde > www > react_native > expo-template-app > styles.ts > default > box

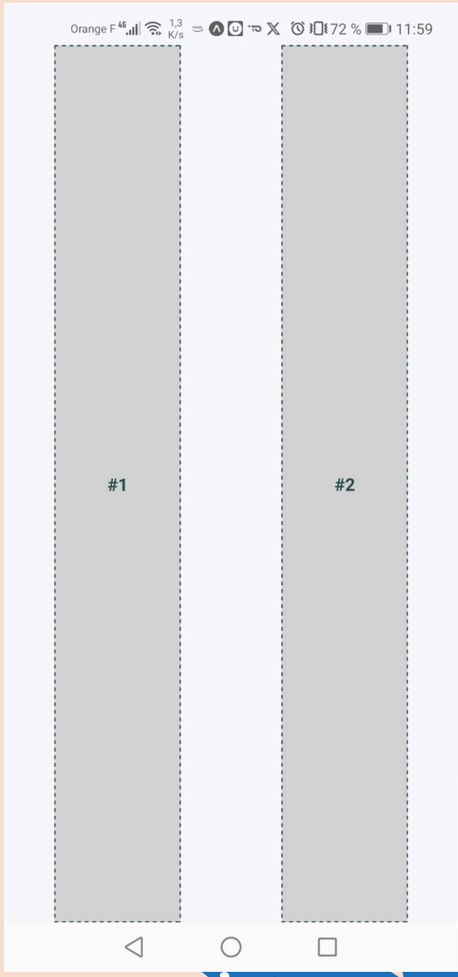
```ts
import { Platform, StyleSheet, StatusBar } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "row",
    backgroundColor: "ghostwhite",
    alignItems: "center",
    justifyContent: "space-around",
    ...Platform.select({
      ios: { paddingTop: 20 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },
  box: {
    width: 100,
    justifyContent: "center",
    alignSelf: "stretch",
    alignItems: "center",
    backgroundColor: "lightgray",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
  },
  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },
});
```

ne > balde > www > react_native > expo-template-app > App.tsx > ...

```tsx
import React from "react";
import { View, StatusBar } from "react-native";
import styles from "./styles";
import Box from "./components/Box";
const boxes = new Array(10).fill(null).map((v, i) => i + 1);
export default function App() {
  return (
    <View style={styles.container}>
      <StatusBar hidden={false} />
      {boxes.map((i) => (
        <Box key={i}>#{i}</Box>
      ))}
    </View>
  );
}
```

home > balde > www > react_native > expo-template-app > App.tsx >

```tsx
import React from "react";
import { Text, View } from "react-native";
import styles from "./styles";
import Box from "./components/Box";
export default function App() {
  return (
    <View style={styles.container}>
      <Box>
        #1
      </Box>
      <Box>
        #2
      </Box>
    </View>
  );
}
```



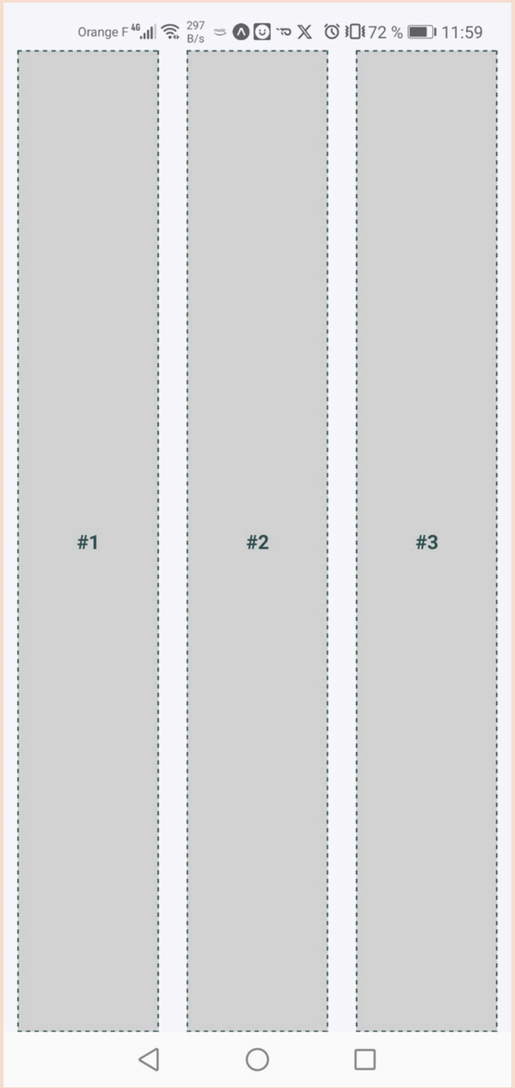
NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

LIGNES FLEXIBLES : ROWS

home > balde > www > react_native > expo-template-app > App.tsx >

```tsx
import React from "react";
import { Text, View } from "react-native";
import styles from "./styles";
import Box from "./components/Box";
export default function App() {
  return (
    <View style={styles.container}>
      <Box>
        #1
      </Box>
      <Box>
        #2
      </Box>
      <Box>
        #3
      </Box>
    </View>
  );
}
```



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

GRILLES FLEXIBLES

home > balde > www > react_native > expo-template-app > TS styles.ts > default > box

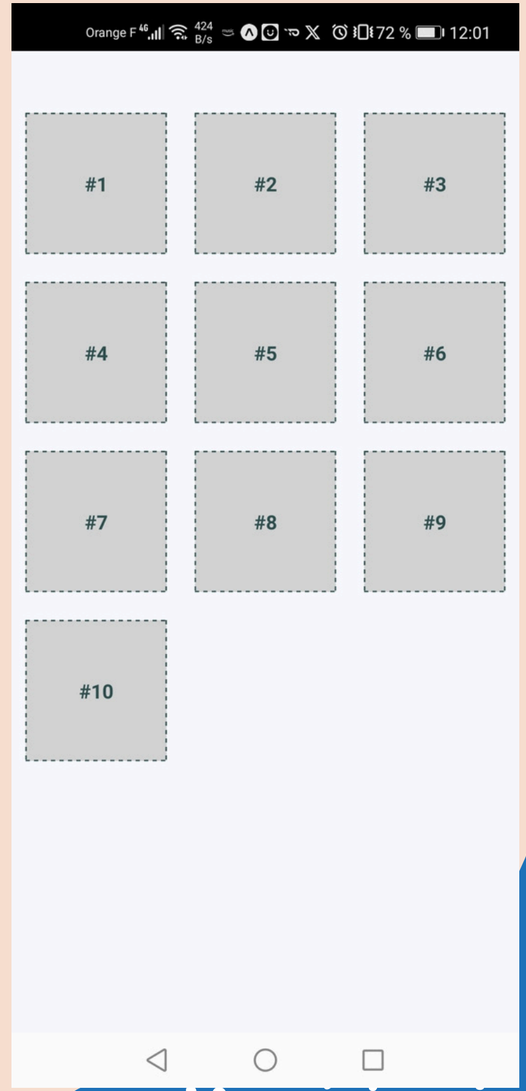
```ts
import { Platform, StyleSheet, StatusBar } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "row",
    backgroundColor: "ghostwhite",
    alignItems: "center",
    justifyContent: "space-around",
    ...Platform.select({
      ios: { paddingTop: 20 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },
  box: {
    width: 100,
    justifyContent: "center",
    alignSelf: "stretch",
    alignItems: "center",
    backgroundColor: "lightgray",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
  },
  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },
});
```

home > balde > www > react_native > expo-template-app > TS styles.ts > default

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "row",
    flexWrap: "wrap",
    backgroundColor: "ghostwhite",
    alignItems: "center",
    ...Platform.select({
      ios: { paddingTop: 40 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },

  box: {
    height: 100,
    width: 100,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "lightgray",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
    margin: 10,
  },

  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },
});
```

home > balde > www > react_native > expo-template-app > App.tsx > ...

```tsx
import React from "react";
import { View, StatusBar } from "react-native";
import styles from "./styles";
import Box from "./components/Box";
const boxes = new Array(10).fill(null).map((v, i) => i + 1);
export default function App() {
  return (
    <View style={styles.container}>
      <StatusBar hidden={false} />
      {boxes.map((i) => (
        <Box key={i}>#{i}</Box>
      ))}
    </View>
  );
}
```



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

LIGNES ET COLONNES FLEXIBLES - LE MIXTE

styles.ts

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    backgroundColor: "ghostwhite",
    alignItems: "center",
    justifyContent: "space-around",
    ...Platform.select({
      ios: { paddingTop: 40 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },

  box: {
    height: 100,
    width: 100,
    justifyContent: "center",
    alignItems: "center",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
    backgroundColor: "lightgray",
  },

  boxText: {
    color: "darkslategray",
    fontWeight: "bold",
  },

  row: {
    flex: 1,
    flexDirection: "row",
    justifyContent: "space-around",
    alignSelf: "stretch",
  },

  column: {
    flex: 1,
    flexDirection: "column",
    alignItems: "center",
    justifyContent: "space-around",
    alignSelf: "stretch",
  },
});
```

Row.tsx

```tsx
import React from "react";
import PropTypes from "prop-types";
import { View } from "react-native";
import styles from "../styles";

type Props = {
  children: React.ReactNode;
};

export default function Row({ children }: Props) {
  return <View style={styles.row}>{children}</View>;
}

Row.propTypes = {
  children: PropTypes.node.isRequired,
};
```

Column.tsx

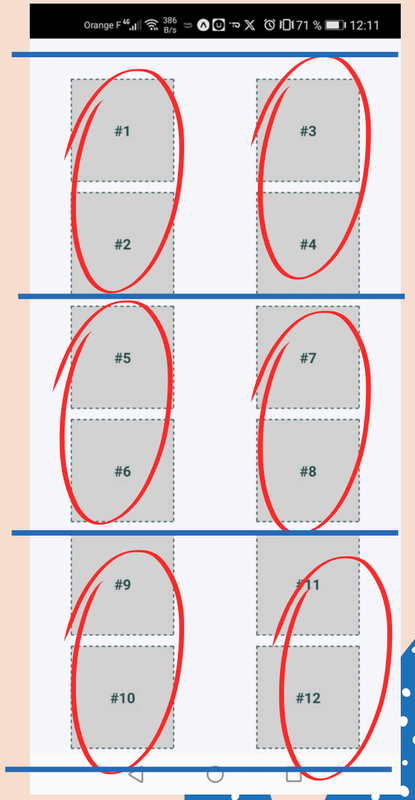
```tsx
import React from "react";
import PropTypes from "prop-types";
import { View } from "react-native";
import styles from "../styles";

type Props = {
  children: React.ReactNode;
};

export default function Column({ children }: Props) {
  return <View style={styles.column}>{children}</View>;
}
Column.propTypes = {
  children: PropTypes.node.isRequired,
};
```

Box.js

```jsx
import React from "react";
import { PropTypes } from "prop-types";
import { View, Text } from "react-native";
import styles from "../styles";
export default function Box({ children }) {
  return (
    <View style={styles.box}>
      <Text style={styles.boxText}>{children}</Text>
    </View>
  );
}
Box.propTypes = {
  children: PropTypes.node.isRequired,
};
```

App.tsx

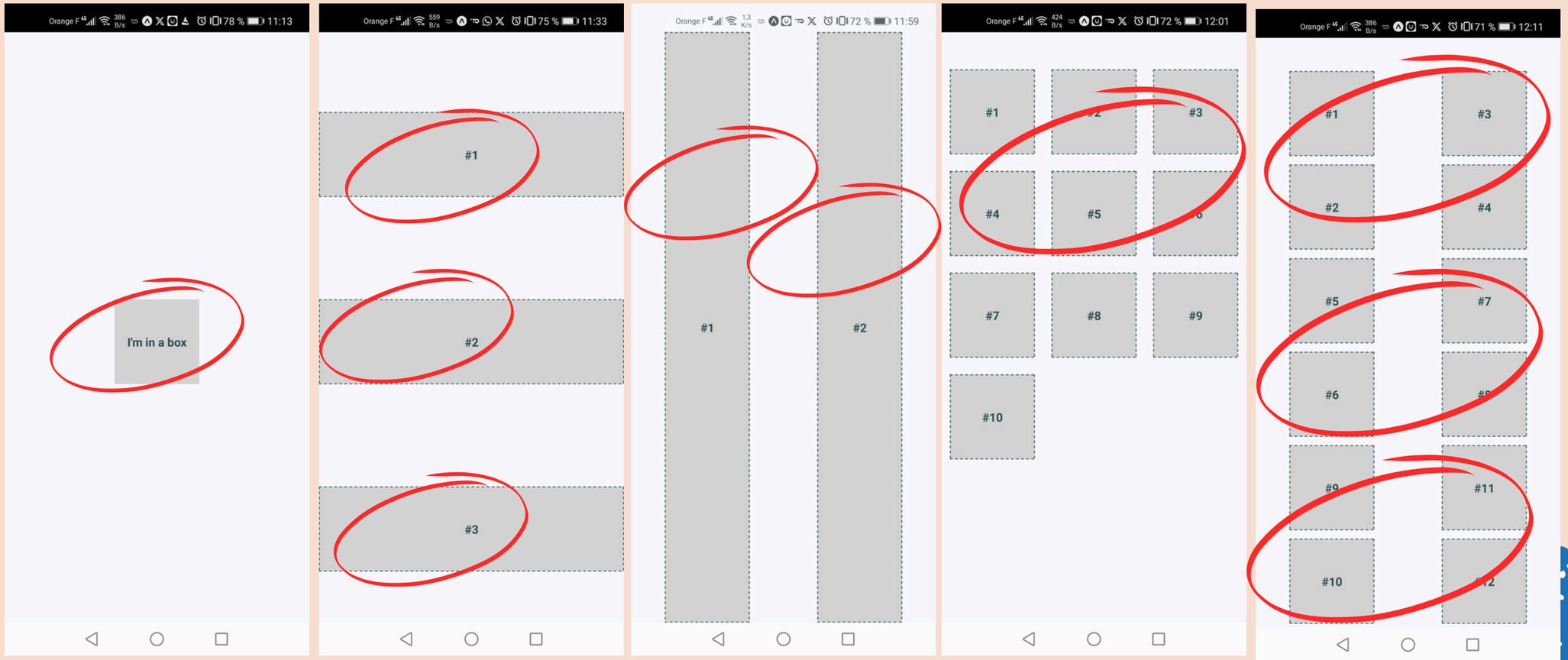
```tsx
import React from "react";
import { StatusBar } from "react-native";
import styles from "./styles";
import Row from "./components/Row";
import Column from "./components/Column";
import Box from "./components/Box";

export default function App() {
  return (
    <View style={styles.container}>
      <StatusBar hidden={false} />
      <Row>
        <Column>
          <Box>#1</Box>
          <Box>#2</Box>
        </Column>
        <Column>
          <Box>#3</Box>
          <Box>#4</Box>
        </Column>
      </Row>
      <Row>
        <Column>
          <Box>#5</Box>
          <Box>#6</Box>
        </Column>
        <Column>
          <Box>#7</Box>
          <Box>#8</Box>
        </Column>
      </Row>
      <Row>
        <Column>
          <Box>#9</Box>
          <Box>#10</Box>
        </Column>
        <Column>
          <Box>#11</Box>
          <Box>#12</Box>
        </Column>
      </Row>
    </View>
  );
}
```



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

LAYOUT PRÉSENTATION SYNTHÈSE



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

NAVIGATION ENTRE LES ÉCRANS

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npm install @react-navigation/native

up to date, audited 1211 packages in 3s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

react-navigation/native est un module clé de la bibliothèque React Navigation, qui est largement utilisée pour la navigation dans les applications React Native. Il fournit les primitives nécessaires pour créer et gérer la navigation dans une application mobile créée avec React Native

dépendances à react-navigation/native et react-navigation/native-stack

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npx expo install react-native-screens react-native-safe-area-context

› Installing 2 SDK 51.0.0 compatible native modules using npm
> npm install

up to date, audited 1211 packages in 3s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npm install @react-navigation/native-stack

up to date, audited 1211 packages in 3s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

home > balde > www > react_native > expo-template-app > TS router.ts > RootStackParamList

```ts
export type RootStackParamList = {
    Home: undefined;
    Settings: undefined;
};
```

react-navigation/native-stack est un module qui fournit un navigateur basé sur des piles pour React Navigation en utilisant les composants natifs pour une performance améliorée. Il est basé sur le package react-native-screens pour utiliser les transitions et les animations de navigation natives.

NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

NAVIGATION ENTRE LES ÉCRANS

home > balde > www > react_native > expo-template-app > components > Settings.tsx > ...

```tsx
import { View, Text, Button, StatusBar } from "react-native";
import styles from "../styles";

import { NativeStackScreenProps } from "@react-navigation/native-stack";
import { RootStackParamList } from "../router";

type Props = NativeStackScreenProps<RootStackParamList>;
export default function Settings({ navigation }: Props) {
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>Settings Screen</Text>
      <Button title="Home" onPress={() => navigation.navigate("Home")} />
    </View>
  );
}
```

Permet de définir les types pour les propriétés de navigation et les routes, facilitant ainsi la gestion des navigateurs de type stack natif dans React Native

gère l'état de la navigation de votre application et doit envelopper vos navigateurs.

home > balde > www > react_native > expo-template-app > TS router.ts > RootStackParamList

```tsx
export type RootStackParamList = {
    Home: undefined;
    Settings: undefined;
  };
```

home > balde > www > react_native > expo-template-app > components > Home.tsx > Props

```tsx
import React from "react";
import { View, Text, Button, StatusBar } from "react-native";
import styles from "../styles";
import { NativeStackScreenProps } from "@react-navigation/native-stack";
import { RootStackParamList } from "../router";

type Props = NativeStackScreenProps<RootStackParamList>;
export default function Home({ navigation }: Props) {
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>Home Screen</Text>
      <Button
        title="Settings"
        onPress={() => navigation.navigate("Settings")}
      />
    </View>
  );
}
```

home > balde > www > react_native > expo-template-app > App.tsx > ...

```tsx
import * as React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import Home from "./components/Home";
import Settings from "./components/Settings";
import { RootStackParamList } from "./router";

const Stack = createNativeStackNavigator<RootStackParamList>();
export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={Home} />
        <Stack.Screen name="Settings" component={Settings} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

fonction utilisée dans React Navigation pour créer un stack navigator natif. Ce stack navigator est optimisé pour fonctionner de manière performante et intégrée dans des environnements natifs comme iOS et Android.

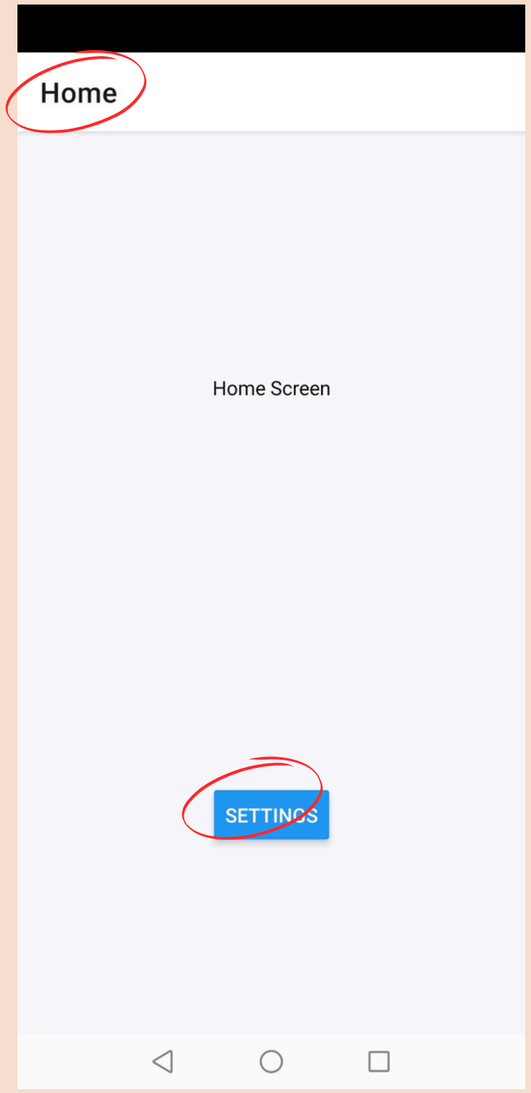
NPX ET CREATE EXPO APP

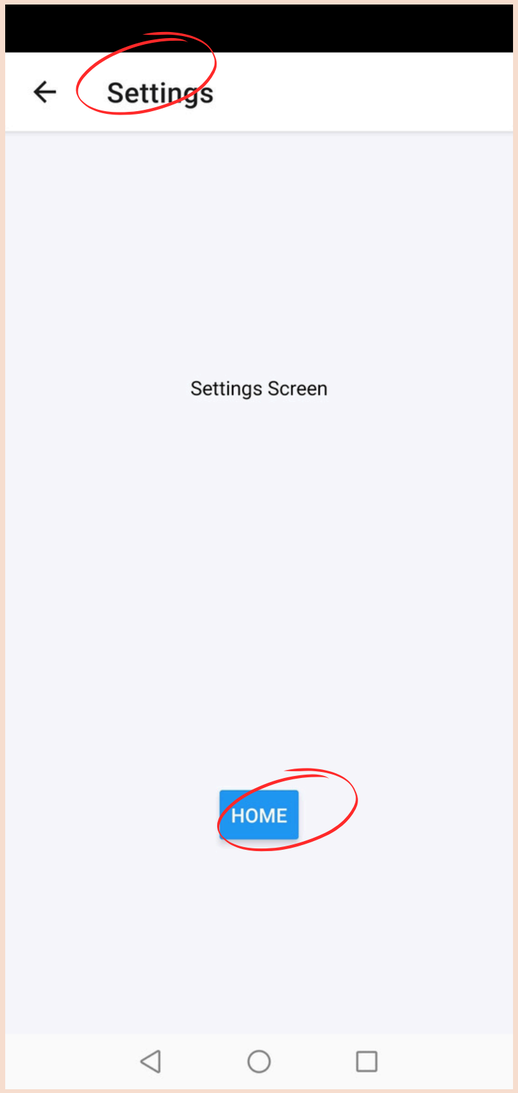
FLEXBOX & LAYOUT "MISE EN PAGE"

NAVIGATION ENTRE LES ÉCRANS

home > balde > www > react_native > expo-template-app > TS styles.ts > default > containe

```ts
import { Platform, StyleSheet, StatusBar } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    backgroundColor: "ghostwhite",
    alignItems: "center",
    justifyContent: "space-around",
    ...Platform.select({
      ios: { paddingTop: 40 },
      android: { paddingTop: StatusBar.currentHeight },
    }),
  },

  box: {
    height: 100,
    width: 100,
    justifyContent: "center",
    alignItems: "center",
    borderWidth: 1,
    borderStyle: "dashed",
    borderColor: "darkslategray",
    backgroundColor: "lightgray",
  },

```





NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

LES PARAMÈTRES DE ROUTE

App.tsx | Home.tsx | Details.tsx | router.ts

home > balde > www > react_native > expo-template-app > router.ts > ...

```ts
export type RootStackParamList = {
  Home: undefined;
  Details: { title: string };
};
```

Ce code définit un alias de type nommé Props qui spécifie le type de propriétés accessibles à un composant React dans un écran de navigation Native Stack nommé "Details" et “Home”.

App.tsx | Home.tsx | Details.tsx | router.ts

home > balde > www > react_native > expo-template-app > components > Home.tsx > Home

```tsx
import React from "react";
import { View, Text, Button, StatusBar } from "react-native";
import styles from "../styles";
import { NativeStackScreenProps } from "@react-navigation/native-stack";
import { RootStackParamList } from "../router";

type Props = NativeStackScreenProps<RootStackParamList, "Home">;
export default function Home({ navigation }: Props) {
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>Home Screen</Text>
      <Button
        title="First Item"
        onPress={() => navigation.navigate("Details", { title: "First Item" })}
      />
      <Button
        title="Second Item"
        onPress={() => navigation.navigate("Details", { title: "Second Item" })}
      />
      <Button
        title="Third Item"
        onPress={() => navigation.navigate("Details", { title: "Third Item" })}
      />
    </View>
  );
}
```

Nom du composant

les paramètres routes

App.tsx | Home.tsx | Details.tsx | router.ts

home > balde > www > react_native > expo-template-app > components > Details.tsx > default

```tsx
import { View, Text, StatusBar } from "react-native";
import styles from "../styles";

import { NativeStackScreenProps } from "@react-navigation/native-stack";
import { RootStackParamList } from "../router";

type Props = NativeStackScreenProps<RootStackParamList, "Details">;
export default function ({ route }: Props) {
  const { title } = route.params;
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>{title}</Text>
    </View>
  );
}
```

Nom du composant

NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

LES PARAMÈTRES DE ROUTE

App.tsx | Home.tsx | Details.tsx | router.ts

home > balde > www > react_native > expo-template-app > App.tsx > App

```tsx
import * as React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import Home from "./components/Home";
import Details from "./components/Details";
import Settings from "./components/Settings";
import { RootStackParamList } from "./router";

const Stack = createNativeStackNavigator<RootStackParamList>();
export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={Home} />
        <Stack.Screen name="Details" component={Details} />
        {/* <Stack.Screen name="Settings" component={Settings} /> */}
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

UTILISATION DE LA BARRE DE NAVIGATION

home > balde > www > react_native > expo-template-app > TS router.ts > RoutesParams

```ts
export type RoutesParams = {
  Home: undefined;
  Details: { title: string; stock: number; content: string };
};
```

home > balde > www > react_native > expo-template-app > components > Details.tsx > default > React.use

```tsx
import React from "react";
import { View, Text, StatusBar } from "react-native";
import styles from "../styles";

import { NativeStackScreenProps } from "@react-navigation/native-stack";
import { RoutesParams } from "../router";

type Props = NativeStackScreenProps<RoutesParams, "Details">;
export default function ({ route, navigation }: Props) {
  const { content, title, stock } = route.params;
  React.useEffect(() => {
    navigation.setOptions({ title });
  }, []);
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>{content} and stock is {stock}</Text>
    </View>
  );
}
```

home > balde > www > react_native > expo-template-app > components > Home.tsx > Home > <func

```tsx
import React from "react";
import { View, Text, Button, StatusBar } from "react-native";
import styles from "../styles";
import { NativeStackScreenProps } from "@react-navigation/native-stack";
import { RoutesParams } from "../router";

type Props = NativeStackScreenProps<RoutesParams, "Home">;
export default function Home({ navigation }: Props) {
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Button
        title="First Item"
        onPress={() =>
          navigation.navigate("Details", {
            title: "First Item",
            content: "First Item Content",
            stock: 1,
          })
        }
      />
      <Button
        title="Second Item"
        onPress={() =>
          navigation.navigate("Details", {
            title: "Second Item",
            content: "Second Item Content",
            stock: 0,
          })
        }
      />
      <Button
        title="Third Item"
        onPress={() =>
          navigation.navigate("Details", {
            title: "Third Item",
            content: "Third Item Content",
            stock: 200,
          })
        }
      />
    </View>
  );
}
```

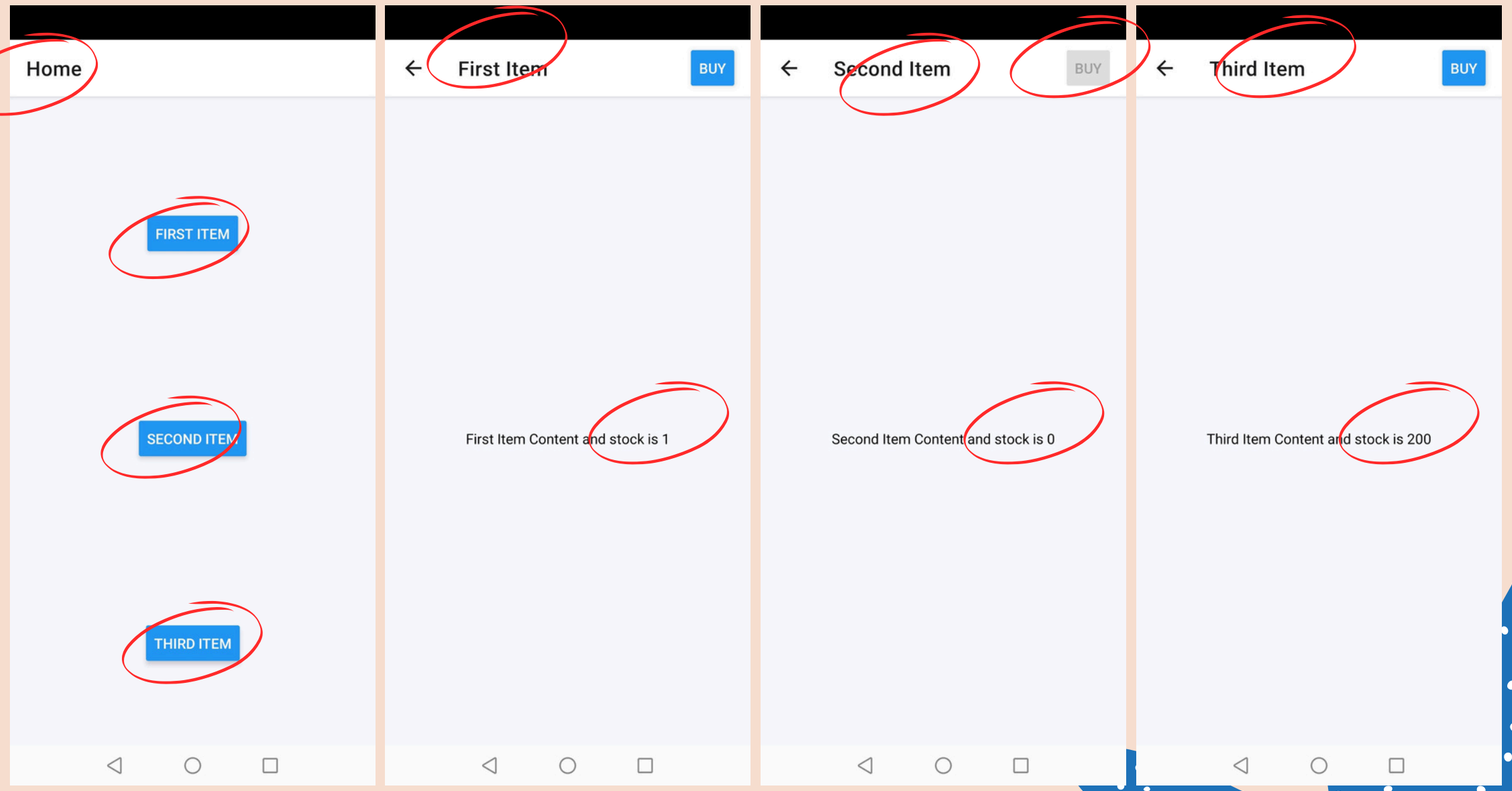
NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

UTILISATION DE LA BARRE DE NAVIGATION

home > balde > www > react_native > expo-template-app > App.tsx > App > <function> > headerRight

```tsx
import * as React from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import Home from "./components/Home";
import Details from "./components/Details";
// import Settings from "./components/Settings";
import { RoutesParams } from "./router";
import { Button } from "react-native";

const Stack = createNativeStackNavigator<RoutesParams>();
export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={Home} />
        <Stack.Screen name="Details" component={Details} options={({ route }) => ({
          headerRight: () => {
            return (
              <Button
                title="Buy"
                onPress={() => {}}
                disabled={route.params.stock === 0}
              />
            );
          },
        })}
        />
        {/* <Stack.Screen name="Settings" component={Settings} /> */}
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```



NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

NAVIGATION EN UTILISANT LES ONGLETS ET LES TABS

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npm install @react-navigation/bottom-tabs @react-navigation/drawer

added 13 packages, and audited 1224 packages in 6s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/expo-template-app$ npx expo install react-native-gesture-handler react-native-reanimated
› Installing 2 SDK 51.0.0 compatible native modules using npm
> npm install

changed 1 package, and audited 1224 packages in 4s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

home > balde > www > react_native > expo-template-app > babel.config.js

```
module.exports = function(api) {
  api.cache(true);
  return {
    presets: ['babel-preset-expo'],
    plugins: ["react-native-reanimated/plugin"],
  };
};
```

home > balde > www > react_native > expo-template-app > router.ts > Routes

```
export type Routes = {
    Home: undefined;
    News: undefined;
    Settings: undefined;
  };
```

App.tsx | package.json | babel.config.js | Home.tsx | Settings.tsx

home > balde > www > react_native > expo-template-app > components > News.tsx > News

```
import React from "react";
import { View, Text } from "react-native";
import styles from "../styles";

export default function News() {
  return (
    <View style={styles.container}>
      <Text>News Content</Text>
    </View>
  );
}
```

App.tsx | package.json | babel.config.js | Home.tsx | Settings.tsx

home > balde > www > react_native > expo-template-app > components > Settings.tsx > Settin

```
import React from "react";
import { View, Text } from "react-native";
import styles from "../styles";

export default function Settings() {
  return (
    <View style={styles.container}>
      <Text>Settings Content</Text>
    </View>
  );
}
```

App.tsx | package.json | babel.config.js | Home.tsx | Settings.tsx

home > balde > www > react_native > expo-template-app > components > Home.tsx > Home

```
import React from "react";
import { View, Text } from "react-native";
import styles from "../styles";

export default function Home() {
  return (
    <View style={styles.container}>
      <Text>Home Content</Text>
    </View>
  );
}
```

NPX ET CREATE EXPO APP

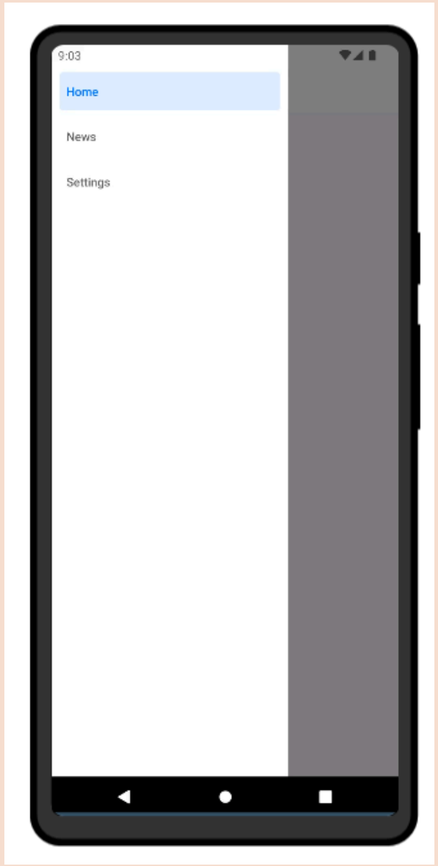
FLEXBOX & LAYOUT "MISE EN PAGE"

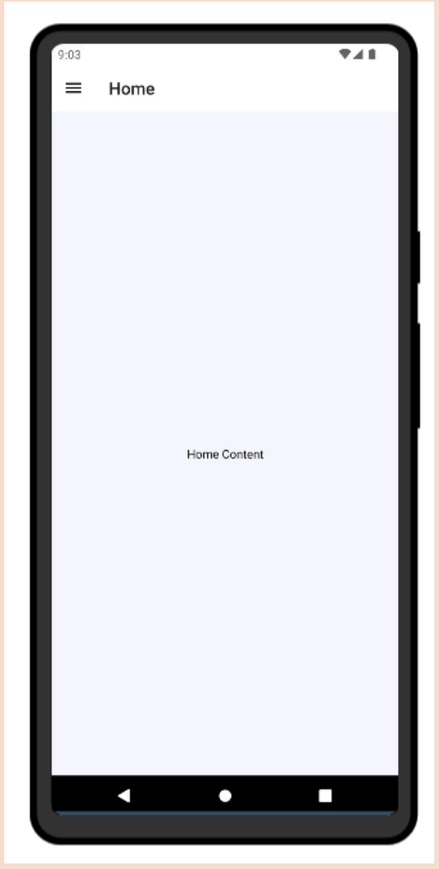
NAVIGATION EN UTILISANT LES ONGLETS ET LES TABS

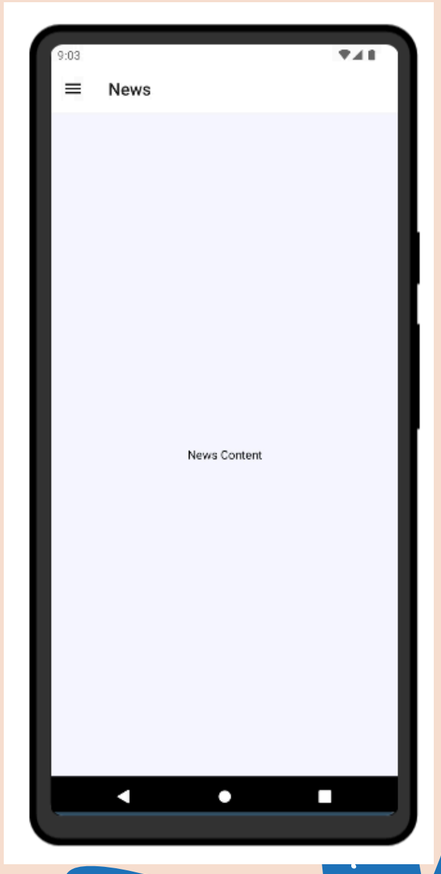
package.json | babel.config.js | App.tsx | Home.tsx | Settings.tsx | Ne

home > balde > www > react-native > expo-template-app > App.tsx > ...

```tsx
import { NavigationContainer } from "@react-navigation/native";
import { createDrawerNavigator } from "@react-navigation/drawer";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import { Platform } from "react-native";
import Home from "./components/Home";
import News from "./components/News";
import Settings from "./components/Settings";
import { Routes } from "./router";

const Tab = createBottomTabNavigator<Routes>();
const Drawer = createDrawerNavigator<Routes>();

export default function App() {
  return (
    <NavigationContainer>
      {Platform.OS === "ios" && (
        <Tab.Navigator>
          <Tab.Screen name="Home" component={Home} />
          <Tab.Screen name="News" component={News} />
          <Tab.Screen name="Settings" component={Settings} />
        </Tab.Navigator>
      )}

      {Platform.OS == "android" && (
        <Drawer.Navigator>
          <Drawer.Screen name="Home" component={Home} />
          <Drawer.Screen name="News" component={News} />
          <Drawer.Screen name="Settings" component={Settings} />
        </Drawer.Navigator>
      )}
    </NavigationContainer>
  );
}
```







NPX ET CREATE EXPO APP

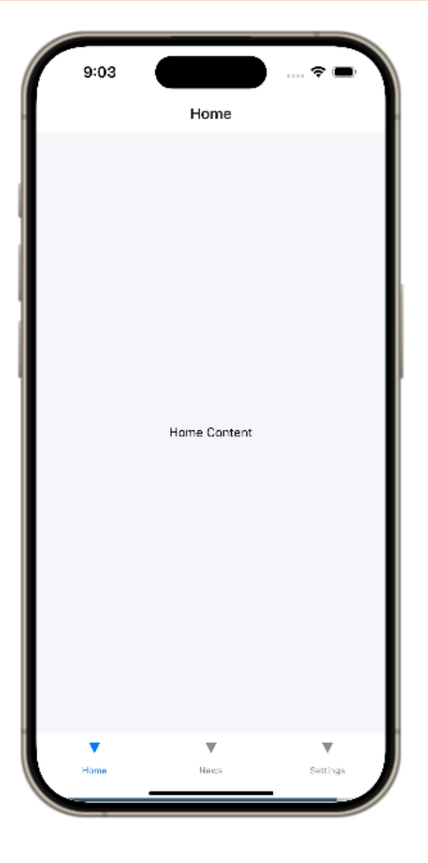
FLEXBOX & LAYOUT "MISE EN PAGE"

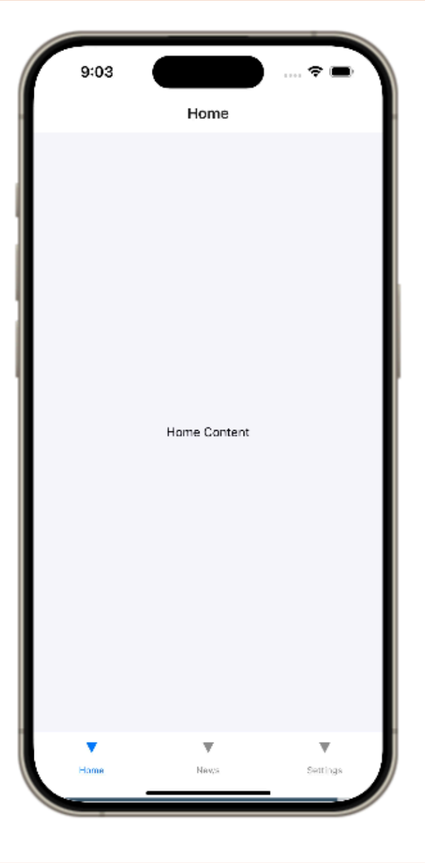
NAVIGATION EN UTILISANT LES ONGLETS ET LES TABS

package.json | babel.config.js | App.tsx | Home.tsx | Settings.tsx

home > balde > www > react_native > expo-template-app > App.tsx > App

```tsx
import { NavigationContainer } from "@react-navigation/native";
import { createDrawerNavigator } from "@react-navigation/drawer";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import { Platform } from "react-native";
import Home from "./components/Home";
import News from "./components/News";
import Settings from "./components/Settings";
import { Routes } from "./router";

const Tab = createBottomTabNavigator<Routes>();
const Drawer = createDrawerNavigator<Routes>();

export default function App() {
  return (
    <NavigationContainer>
      {Platform.OS === "android" && (
        <Tab.Navigator>
          <Tab.Screen name="Home" component={Home} />
          <Tab.Screen name="News" component={News} />
          <Tab.Screen name="Settings" component={Settings} />
        </Tab.Navigator>
      )}

      {Platform.OS == "ios" && (
        <Drawer.Navigator>
          <Drawer.Screen name="Home" component={Home} />
          <Drawer.Screen name="News" component={News} />
          <Drawer.Screen name="Settings" component={Settings} />
        </Drawer.Navigator>
      )}
    </NavigationContainer>
  );
}
```





NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

NAVIGATION EN UTILISANT LES LIENS

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx create-expo-app route-file-based --templa
te
? Choose a template: › - Use arrow-keys. Return to submit.
    Default
    Blank
    Blank (TypeScript)
❯   Navigation (TypeScript) - File-based routing with TypeScript enabled
    Blank (Bare)
```

```
✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd route-file-based
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/route-file-based$ ls app
 +html.tsx   _layout.tsx   modal.tsx   +not-found.tsx   '(tabs)'
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/route-file-based$
```

_layout.tsx | index.tsx | settings.tsx

home > balde > www > react_native > route-file-based > app > _layout.tsx > ...

```tsx
import { Stack } from "expo-router";
export default function RootLayout() {
  return <Stack />;
}
```

bibliothèque qui permet de gérer la navigation dans les applications Expo de manière simple et efficace en utilisant un système basé sur les routes de fichiers.

_layout.tsx | index.tsx | settings.tsx

home > balde > www > react_native > route-file-based > app > index.tsx > Home

```tsx
export default function Home() {
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>Home Screen</Text>

      <Link href="/settings" asChild>
        <Button title="Settings" />
      </Link>
    </View>
  );
}
```

_layout.tsx | index.tsx | settings.tsx

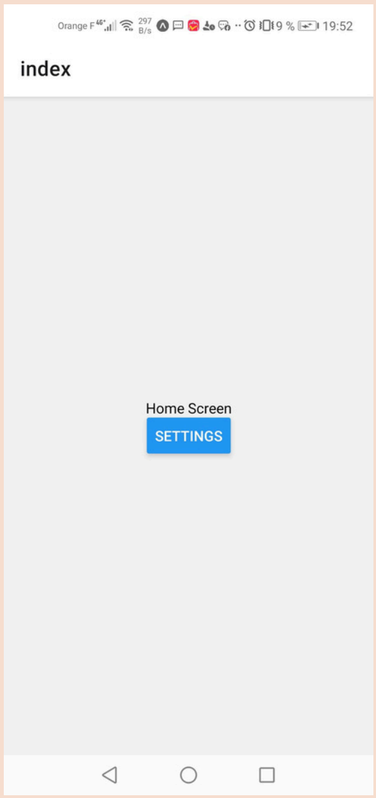
home > balde > www > react_native > route-file-based > app > settings.tsx > ...

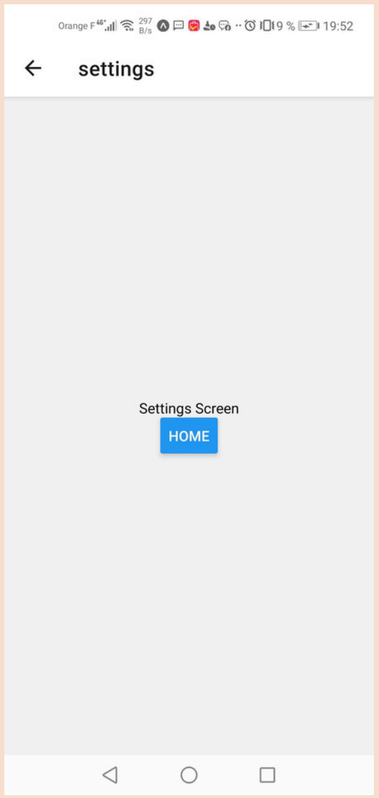
```tsx
import React from "react";
import { View, Text, Button, StatusBar } from "react-native";
import styles from "../styles";
import { Link } from "expo-router";

export default function Settings() {
  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>Settings Screen</Text>

      <Link href="/" asChild>
        <Button title="Home" />
      </Link>
    </View>
  );
}
```

NPX ET CREATE EXPO APP

FLEXBOX & LAYOUT "MISE EN PAGE"

NAVIGATION EN UTILISANT LES LIENS





NPX ET CREATE EXPO APP

RENDU DES LISTES D'ÉLÉMENTS

RENDU DES COLLECTIONS DE DONNÉES

App.tsx | styles.ts

home > balde > www > react_native > expo-template-app > App.tsx > ...

```tsx
import { Text, View, FlatList } from "react-native";
import styles from "./styles";

const data = new Array(100)
  .fill(null)
  .map((v, i) => ({ key: i.toString(), value: `Item ${i}` }));

export default function App() {
  return (
    <View style={styles.container}>
      <FlatList
        data={data}
        renderItem={({ item }) => <Text style={styles.item}>{item.value}</Text>}
      />
    </View>
  );
}
```

permet de rendre des listes défilables de données de manière performante

App.tsx | styles.ts

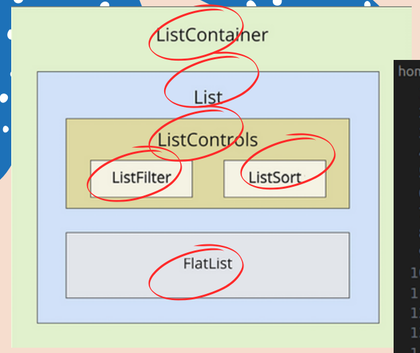
home > balde > www > react_native > expo-template-app > styles.ts > ...

```ts
import { StyleSheet } from "react-native";
export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    paddingTop: 40,
  },
  item: {
    margin: 5,
    padding: 5,
    color: "slategrey",
    backgroundColor: "ghostwhite",
    textAlign: "center",
  },
});
```

NPX ET CREATE EXPO APP

RENDU DES LISTES D'ÉLÉMENTS

TRIER ET FILTRER LES LISTES



home > balde > www > react_native > expo-template-app > components > ListContainer.tsx > ListContainer

```tsx
import React, { useState, useMemo } from "react";
import List from "./List";

function mapItems(items: string[]) {
  return items.map((value, i) => ({ key: i.toString(), value }));
}

const array = new Array(100).fill(null).map((v, i) => `Item ${i}`);

function filterAndSort(text: string, asc: boolean): string[] {
  return array
    .filter((i) => text.length === 0 || i.includes(text))
    .sort(
      asc
        ? (a, b) => (a > b ? 1 : a < b ? -1 : 0)
        : (b, a) => (b > a ? 1 : b < a ? -1 : 0)
    );
}

export default function ListContainer() {
  const [asc, setAsc] = useState(true);
  const [filter, setFilter] = useState("");

  const data = useMemo(() => {
    return filterAndSort(filter, asc);
  }, [filter, asc]);

  return (
    <List
      data={mapItems(data)}
      asc={asc}
      onFilter={(text) => {
        setFilter(text);
      }}
      onSort={() => {
        setAsc(!asc);
      }}
    />
  );
}
```

home > balde > www > react_native > expo-template-app > components > List.tsx > List

```tsx
import React from "react";
import { Text, FlatList } from "react-native";
import styles from "../styles";
import ListControls from "./ListControls";

type Props = {
  data: { key: string; value: string }[];
  onFilter: (text: string) => void;
  onSort: () => void;
  asc: boolean;
};

export default function List({ data, onFilter, onSort, asc }: Props) {
  return (
    <FlatList
      data={data}
      ListHeaderComponent={<ListControls {...{ onFilter, onSort, asc }} />}
      renderItem={({ item }) => <Text style={styles.item}>{item.value}</Text>}
    />
  );
}
```

home > balde > www > react_native > expo-template-app > TS styles.ts

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    paddingTop: 40,
  },

  item: {
    margin: 5,
    padding: 5,
    color: "slategrey",
    backgroundColor: "ghostwhite",
    textAlign: "center",
  },

  filter: {
    height: 40,
    width: 200,
  },

  controls: {
    flex: 1,
    flexDirection: "row",
    justifyContent: "space-between",
    alignItems: "center",
    padding: 10,
    backgroundColor: "white",
  },
});
```

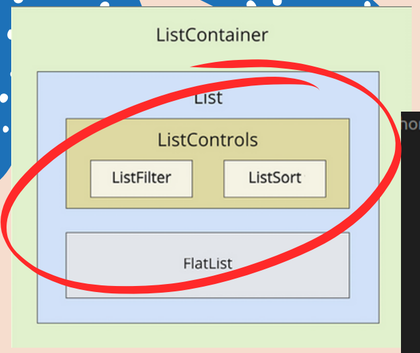
home > balde > www > react_native > expo-template-app > components > ListControls.tsx > ...

```tsx
import React from "react";
import { View } from "react-native";
import styles from "../styles";
import ListFilter from "./ListFilter";
import ListSort from "./ListSort";

type Props = {
  onFilter: (text: string) => void;
  onSort: () => void;
  asc: boolean;
};

export default function ListControls({ onFilter, onSort, asc }: Props) {
  return (
    <View style={styles.controls}>
      <ListFilter onFilter={onFilter} />
      <ListSort onSort={onSort} asc={asc} />
    </View>
  );
}
```

NPX ET CREATE EXPO APP

RENDU DES LISTES D'ÉLÉMENTS

TRIER ET FILTRER LES LISTES



home > balde > www > react_native > expo-template-app > components > ListFilter.tsx

```tsx
import React from "react";
import { View, TextInput } from "react-native";
import styles from "../styles";

type Props = {
  onFilter: (text: string) => void;
};

export default function ListFilter({ onFilter }: Props) {
  return (
    <View>
      <TextInput
        autoFocus
        placeholder="Search"
        style={styles.filter}
        onChangeText={onFilter}
      />
    </View>
  );
}
```

home > balde > www > react_native > expo-template-app > components > ListSort.tsx

```tsx
import React from "react";
import { Text } from "react-native";

const arrows = new Map([
  [true, "▼"],
  [false, "▲"],
]);

type Props = {
  onSort: () => void;
  asc: boolean;
};

export default function ListSort({ onSort, asc }: Props) {
  return <Text onPress={onSort}>{arrows.get(asc)}</Text>;
}
```

NPX ET CREATE EXPO APP

RENDU DES LISTES D'ÉLÉMENTS

RÉCUPÉRATION DES DONNÉES DE LA LISTE VIA API

home > balde > www > react_native > expo-template-app > TS api.ts > filterAndSort

```ts
const items = new Array(100).fill(null).map((v, i) => `Item ${i}`);

function filterAndSort(data: string[], text: string, asc: boolean) {
  return data
    .filter((i) => text.length === 0 || i.includes(text))
    .sort(
      asc
        ? (a, b) => (b > a ? -1 : a === b ? 0 : 1)
        : (a, b) => (a > b ? -1 : a === b ? 0 : 1)
    );
}

export function fetchItems(
  filter: string,
  asc: boolean
): Promise<{ json: () => Promise<{ items: string[] }> }> {
  return new Promise((resolve) => {
    resolve({
      json: () =>
        Promise.resolve({
          items: filterAndSort(items, filter, asc),
        }),
    });
  });
}
```

home > balde > www > react_native > expo-template-app > components > ListContainer.tsx

```tsx
import React, { useState, useEffect } from "react";
import { fetchItems } from "../api";
import List from "./List";

type MappedList = { key: string; value: string }[];

function mapItems(items: string[]): MappedList {
  return items.map((value, i) => ({ key: i.toString(), value }));
}

export default function ListContainer() {
  const [asc, setAsc] = useState(true);
  const [filter, setFilter] = useState("");
  const [data, setData] = useState<MappedList>([]);

  useEffect(() => {
    fetchItems(filter, asc)
      .then((resp) => resp.json())
      .then(({ items }) => {
        setData(mapItems(items));
      });
  }, []);

  return (
    <List
      data={data}
      asc={asc}
      onFilter={(text) => {
        fetchItems(text, asc)
          .then((resp) => resp.json())
          .then(({ items }) => {
            setFilter(text);
            setData(mapItems(items));
          });
      }}
      onSort={() => {
        fetchItems(filter, !asc)
          .then((resp) => resp.json())
          .then(({ items }) => {
            setAsc(!asc);
            setData(mapItems(items));
          });
      }}
    />
  );
}
```

NPX ET CREATE EXPO APP

RENDU DES LISTES D'ÉLÉMENTS

CHARGEMENT DIFFERÉ "LOADING" DE LA LISTE

home > balde > www > react_native > expo-template-app > TS api.ts > fetchItems

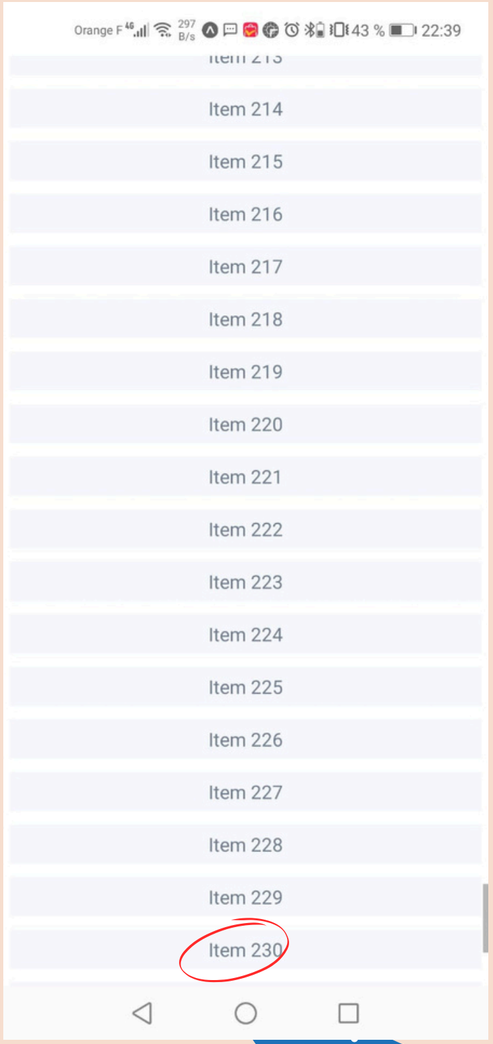
```ts
function* genItems() {
  let cnt = 0;

  while (true) {
    yield `Item ${cnt++}`;
  }
}

let items = genItems();

export function fetchItems({ refresh }: { refresh?: boolean }) {
  if (refresh) {
    items = genItems();
  }

  return Promise.resolve({
    json: () =>
      Promise.resolve({
        items: new Array(30).fill(null).map(() => items.next().value as string),
      }),
  });
}
```

home > balde > www > react_native > expo-template-app > components > List.tsx > List

```tsx
import React from "react";
import { Text, FlatList } from "react-native";
import styles from "../styles";

type Props = {
  data: { key: string; value: string }[];
  fetchItems: () => Promise<void>;
};

export default function List({
  data,
  fetchItems,
}: Props) {
  return (
    <FlatList
      data={data}
      renderItem={({ item }) => <Text style={styles.item}>{item.value}</Text>}
      onEndReached={fetchItems}
    />
  );
}
```

home > balde > www > react_native > expo-template-app > components > ListContainer.tsx > ListContaine

```tsx
import React, { useState, useEffect } from "react";
import * as api from "../api";
import List from "./List";

export default function ListContainer() {
  const [data, setData] = useState<{ key: string; value: string }[]>([]);
  const [isRefreshing, setIsRefreshing] = useState(false);

  function fetchItems() {
    return api
      .fetchItems({})
      .then((resp) => resp.json())
      .then(({ items }) => {
        setData([
          ...data,
          ...items.map((value) => ({
            key: value,
            value,
          })),
        ]);
      });
  }

  useEffect(() => {
    fetchItems();
  }, []);

  return (
    <List data={data} fetchItems={fetchItems}
    />
  );
}
```



NPX ET CREATE EXPO APP

RENDU DES LISTES D'ÉLÉMENTS

IMPLÉMENTATION DE "PULL TO REFRESH" => TIRER POUR ACTUALISER

home > balde > www > react_native > expo-template-app > TS api.ts > fetchItems

```ts
function* genItems() {
  let cnt = 0;

  while (true) {
    yield `Item ${cnt++}`;
  }
}

let items = genItems();

export function fetchItems({ refresh }: { refresh?: boolean }) {
  if (refresh) {
    items = genItems();
  }

  return Promise.resolve({
    json: () =>
      Promise.resolve({
        items: new Array(30).fill(null).map(() => items.next().value as string),
      }),
  });
}
```

home > balde > www > react_native > expo-template-app > components > List.tsx > List

```tsx
import React from "react";
import { Text, FlatList } from "react-native";
import styles from "../styles";

type Props = {
  data: { key: string; value: string }[];
  fetchItems: () => Promise<void>;
  refreshItems: () => Promise<void>;
  isRefreshing: boolean;
};

export default function List({
  data,
  fetchItems,
  refreshItems,
  isRefreshing,
}: Props) {
  return (
    <FlatList
      data={data}
      renderItem={({ item }) => <Text style={styles.item}>{item.value}</Text>}
      onEndReached={fetchItems}
      onRefresh={refreshItems}
      refreshing={isRefreshing}
    />
  );
}
```

home > balde > www > react_native > expo-template-app > components > ListContainer.tsx > ListContainer

```tsx
import React, { useState, useEffect } from "react";
import * as api from "../api";
import List from "./List";

export default function ListContainer() {
  const [data, setData] = useState<{ key: string; value: string }[]>([]);
  const [isRefreshing, setIsRefreshing] = useState(false);

  function fetchItems() {
    return api
      .fetchItems({})
      .then((resp) => resp.json())
      .then(({ items }) => {
        setData([
          ...data,
          ...items.map((value) => ({
            key: value,
            value,
          })),
        ]);
      });
  }

  function refreshItems() {
```

home > balde > www > react_native > expo-template-app > components > ListContainer.tsx

```tsx
export default function ListContainer() {
  }

  function refreshItems() {
    setIsRefreshing(true);
    return api
      .fetchItems({ refresh: true })
      .then((resp) => resp.json())
      .then(({ items }) => {
        setData(
          items.map((value) => ({
            key: value,
            value,
          }))
        );
      })
      .finally(() => {
        setIsRefreshing(false);
      });
  }

  useEffect(() => {
    fetchItems();
  }, []);

  return (
    <List data={data} fetchItems={fetchItems}
      refreshItems={refreshItems}
      isRefreshing={isRefreshing}
    />
  );
}
```

Comme nous avons fourni les props onRefresh et refreshing, notre composant FlatList active automatiquement le geste tirer pour actualiser. Le gestionnaire onRefresh sera appelé lorsque vous extrayez la liste et la propriété d'actualisation permettra au spinner de chargement de refléter l'état de chargement.

Pour appliquer les props définis dans le composant List, implémentons la fonction refreshItems avec l'état isRefreshing dans le composant ListContainer. Dans refreshItems, ainsi que dans la méthode fetchItems, nous obtenons les éléments de la liste mais les enregistrons en tant que nouvelle liste.

NPX ET CREATE EXPO APP

LA GEOLOCALISATION - CARTE GOOGLE MAPS

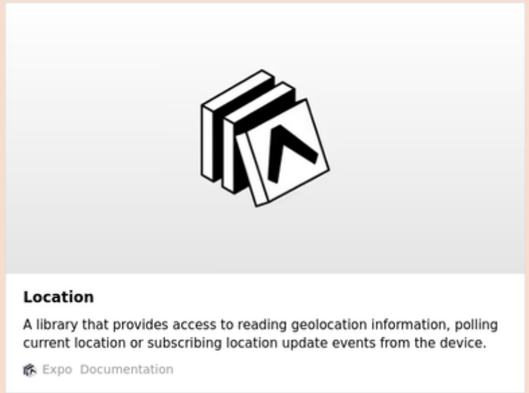
UTILISATION DE L'API DE GÉOLOCALISATION

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx create-expo-app geo-and-maps --template
✔ Choose a template: › Blank (TypeScript)
✔ Downloaded and extracted project files.
> npm install
```

```
To run your project, navigate to the directory and run one of the following npm commands.

- cd geo-and-maps
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ cd geo-and-maps/
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/geo-and-maps$ npx expo install expo-location
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

added 1 package, and audited 1194 packages in 4s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

expo-location est un module Expo qui fournit un accès facile à la localisation de l'appareil sur les appareils mobiles. Il permet à une application Expo d'obtenir des informations sur la position géographique de l'appareil, telles que la latitude, la longitude, la vitesse, l'altitude, etc. Cela permet de créer des applications qui nécessitent des fonctionnalités basées sur la localisation, comme la cartographie, la navigation, le suivi, etc.



https://docs.expo.dev/versions/latest/sdk/location/

Welcome | app.json A | package.json M

geo-and-maps > package.json > ...

```
{
  "name": "geo-and-maps",
  "version": "1.0.0",
  "main": "expo/AppEntry.js",
  Debug
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web"
  },
  "dependencies": {
    "expo": "~51.0.8",
    "expo-status-bar": "~1.12.1",
    "react": "18.2.0",
    "react-native": "0.74.1",
    "expo-location": "~17.0.1"
  },
  "devDependencies": {
    "@babel/core": "^7.20.0",
    "@types/react": "~18.2.45",
    "typescript": "^5.1.3"
  },
  "private": true
}
```

Welcome | app.json M | package.json M

geo-and-maps > app.json > {} expo

```
{
  "expo": {
    "name": "geo-and-maps",
    "slug": "geo-and-maps",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "userInterfaceStyle": "light",
    "splash": {
      "image": "./assets/splash.png",
      "resizeMode": "contain",
      "backgroundColor": "#ffffff"
    },
    "ios": {
      "supportsTablet": true
    },
    "android": {
      "adaptiveIcon": {
        "foregroundImage": "./assets/adaptive-icon.png",
        "backgroundColor": "#ffffff"
      }
    },
    "web": {
      "favicon": "./assets/favicon.png"
    },
    "plugins": [
      [
        "expo-location",
        {
          "locationAlwaysAndWhenInUsePermission": "Allow $(PRODUCT_NAME) to use your location."
        }
      ]
    ]
  }
}
```

NPX ET CREATE EXPO APP

LA GEOLOCALISATION - CARTE GOOGLE MAPS

UTILISATION DE L'API DE GÉOLOCALISATION

geo-and-maps > App.tsx > WhereAmI > useEffect() callback > setPosition > then() callback

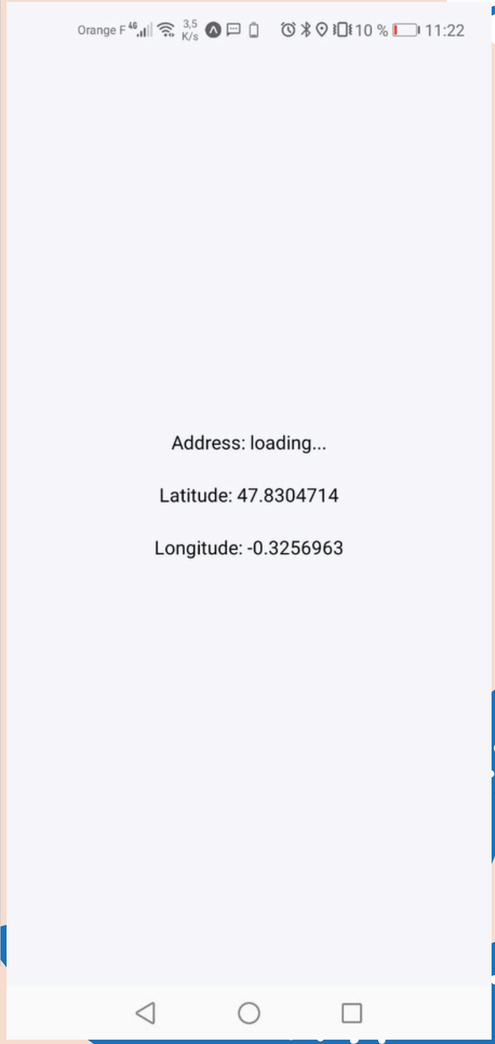
```tsx
import React, { useState, useEffect } from "react";
import { Text, View } from "react-native";
import * as Location from "expo-location";
import styles from "./styles";

const API_KEY = "";
const URL = `https://maps.google.com/maps/api/geocode/json?key=${API_KEY}&latlng=`;

export default function WhereAmI() {
  const [address, setAddress] = useState("loading...");
  const [longitude, setLongitude] = useState<number | undefined>();
  const [latitude, setLatitude] = useState<number | undefined>();

  useEffect(() => {
    function setPosition({
      coords: { latitude, longitude },
    }: Location.LocationObject) {
      setLongitude(longitude);
      setLatitude(latitude);

      fetch(`${URL}${latitude},${longitude}`)
        .then((resp) => resp.json())
        .then(({ results }) => {
          if (results.length > 0) {
            setAddress(results[0].formatted_address);
          }
        })
        .catch((error) => {
          console.log(error.message);
        });
    }

    let watcher: Location.LocationSubscription;

    (async () => {
      let { status } = await Location.requestForegroundPermissionsAsync();
      if (status !== "granted") {
        return;
      }

      let location = await Location.getCurrentPositionAsync({});
      setPosition(location);

      watcher = await Location.watchPositionAsync(
        { accuracy: Location.LocationAccuracy.Highest },
        setPosition
      );
    })();

    return () => {
```

geo-and-maps > App.tsx > WhereAmI > useEffect() callback > setPosition > then() ca

```tsx
export default function WhereAmI() {
  useEffect(() => {
    (async () => {
      watcher = await Location.watchPositionAsync(
        { accuracy: Location.LocationAccuracy.Highest },
        setPosition
      );
    })();

    return () => {
      watcher?.remove();
    };
  }, []);

  return (
    <View style={styles.container}>
      <Text style={styles.label}>Address: {address}</Text>
      <Text style={styles.label}>Latitude: {latitude}</Text>
      <Text style={styles.label}>Longitude: {longitude}</Text>
    </View>
  );
}
```

geo-and-maps > styles.ts > default > label

```tsx
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },
  label: {
    textAlign: "center",
    margin: 10,
  },

  address: {
    fontWeight: "bold",
  },
});
```



NPX ET CREATE EXPO APP

LA GEOLOCALISATION - CARTE GOOGLE MAPS

RENDU DE LA CARTE

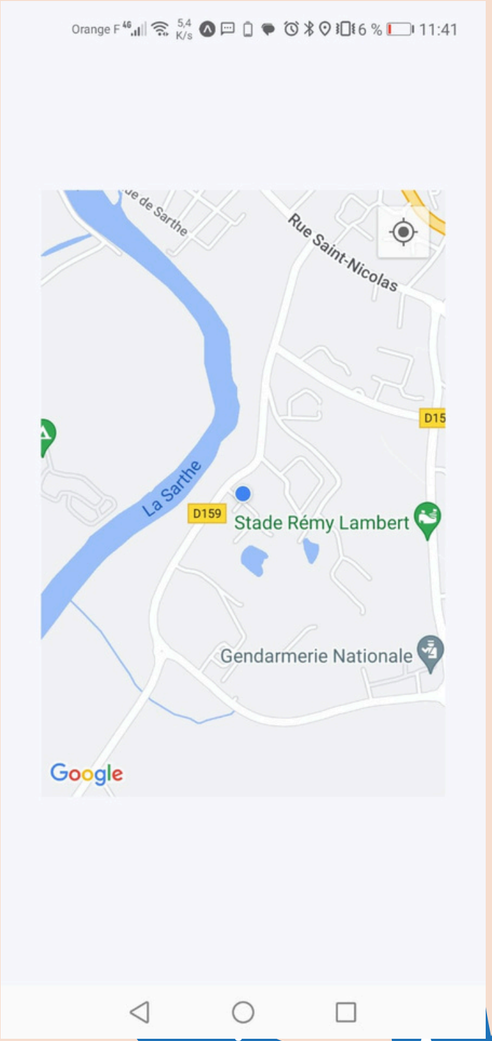
```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/geo-and-maps$ npm i react-native-maps

added 2 packages, and audited 1196 packages in 4s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

geo-and-maps > App.tsx > default

```tsx
import React from "react";
import { View, StatusBar } from "react-native";
import MapView from "react-native-maps";
import styles from "./styles";

StatusBar.setBarStyle("dark-content");

export default () => (
  <View style={styles.container}>
    <MapView style={styles.mapView} showsUserLocation followsUserLocation />
  </View>
);
```

geo-and-maps > TS styles.ts > default > mapView

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  mapView: {
    alignSelf: "stretch",
    height: 450,
    margin: 30,
  },
});
```



NPX ET CREATE EXPO APP

LA GEOLOCALISATION - CARTE GOOGLE MAPS

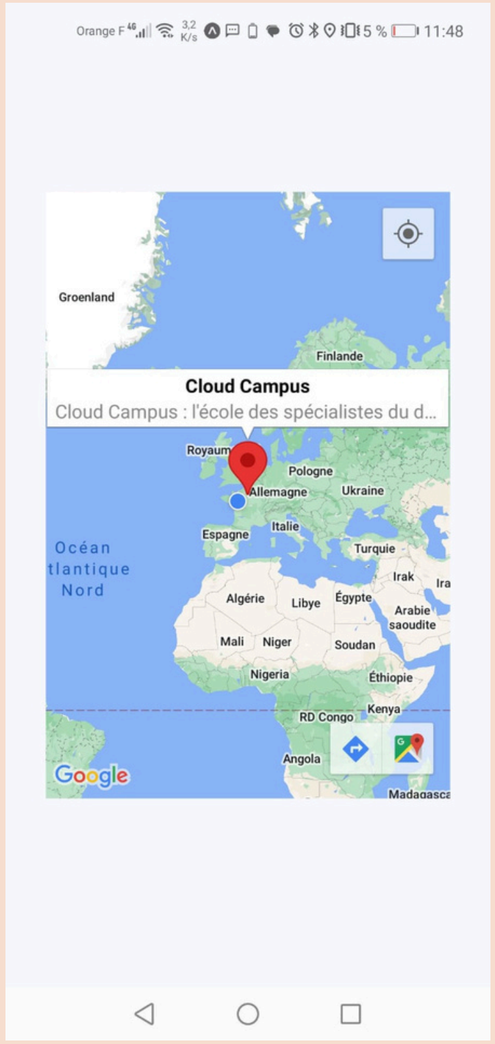
ANNOTER DES POINTS D'INTÉRÊT

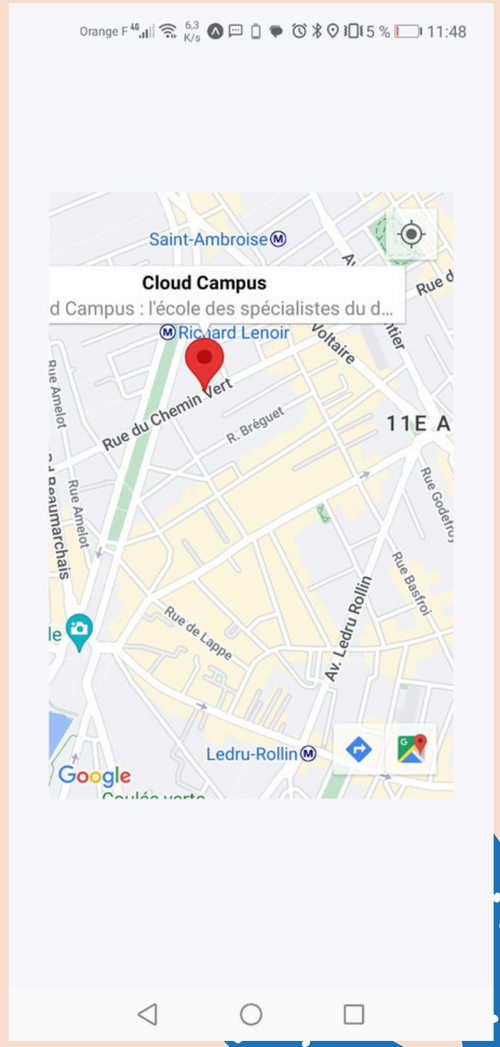
geo-and-maps > App.tsx > ...

```tsx
import React from "react";
import { View, StatusBar } from "react-native";
import MapView, {Marker} from "react-native-maps";
import styles from "./styles";

StatusBar.setBarStyle("dark-content");

export default () => (
  <View style={styles.container}>
    <MapView style={styles.mapView} showsUserLocation followsUserLocation >

    <Marker
          title="Cloud Campus"
          description="Cloud Campus : l'école des spécialistes du développement web"
          coordinate={{
            latitude: 48.8585802,
            longitude: 2.3730884,
          }}
        />
    </MapView>
  </View>
);
```

geo-and-maps > styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  mapView: {
    alignSelf: "stretch",
    height: 450,
    margin: 30,
  },
});
```





NPX ET CREATE EXPO APP

LA GEOLOCALISATION - CARTE GOOGLE MAPS

TRAÇAGE DES SUPERPOSITIONS

geo-and-maps > App.tsx > PlottingOverlays > onClickStout

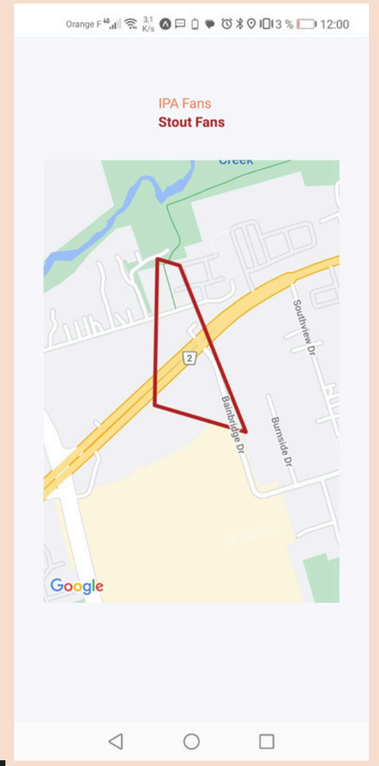
```tsx
import React, { useState } from "react";
import { View, Text, StatusBar } from "react-native";
import MapView, { Polygon } from "react-native-maps";
import styles from "./styles";

StatusBar.setBarStyle("dark-content");

type Overlay = {
  coordinates: { latitude: number; longitude: number }[];
  strokeColor: string;
  strokeWidth: number;
};

const ipaRegion: Overlay = {
  coordinates: [
    { latitude: 43.8486744, longitude: -79.0695283 },
    { latitude: 43.8537168, longitude: -79.0700046 },
    { latitude: 43.8518394, longitude: -79.0725697 },
    { latitude: 43.8481651, longitude: -79.0716377 },
    { latitude: 43.8486744, longitude: -79.0695283 },
  ],
  strokeColor: "coral",
  strokeWidth: 4,
};

const stoutRegion: Overlay = {
  coordinates: [
    { latitude: 43.8486744, longitude: -79.0693283 },
    { latitude: 43.8517168, longitude: -79.0710046 },
    { latitude: 43.8518394, longitude: -79.0715697 },
    { latitude: 43.8491651, longitude: -79.0716377 },
    { latitude: 43.8486744, longitude: -79.0693283 },
  ],
  strokeColor: "firebrick",
  strokeWidth: 4,
};

export default function PlottingOverlays() {
  const [ipaStyles, setIpaStyles] = useState<any>([
    styles.ipaText,
    styles.boldText,
  ]);
  const [stoutStyles, setStoutStyles] = useState<any>([styles.stoutText]);
  const [overlays, setOverlays] = useState<Overlay[]>([ipaRegion]);

  function onClickIpa() {
    setIpaStyles([...ipaStyles, styles.boldText]);
    setStoutStyles([stoutStyles[0]]);
    setOverlays([ipaRegion]);
  }

  function onClickStout() {
    setStoutStyles([...stoutStyles, styles.boldText]);
    setIpaStyles([ipaStyles[0]]);
    setOverlays([stoutRegion]);
  }
```

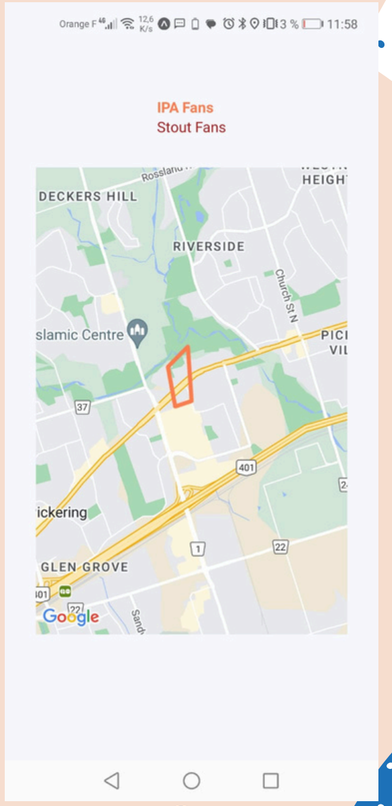
geo-and-maps > App.tsx > PlottingOverlays > onClickStout

```tsx
export default function PlottingOverlays() {
  function onClickStout() {
    setStoutStyles([...stoutStyles, styles.boldText]);
    setIpaStyles([ipaStyles[0]]);
    setOverlays([stoutRegion]);
  }

  return (
    <View style={styles.container}>
      <View>
        <Text style={ipaStyles} onPress={onClickIpa}>
          IPA Fans
        </Text>
        <Text style={stoutStyles} onPress={onClickStout}>
          Stout Fans
        </Text>
      </View>
      <MapView
        style={styles.mapView}
        showsPointsOfInterest={false}
        initialRegion={{
          latitude: 43.8486744,
          longitude: -79.0695283,
          latitudeDelta: 0.002,
          longitudeDelta: 0.04,
        }}
      >
        {overlays.map((v, i) => (
          <Polygon
            key={i}
            coordinates={v.coordinates}
            strokeColor={v.strokeColor}
            strokeWidth={v.strokeWidth}
          />
        ))}
      </MapView>
    </View>
  );
}
```

geo-and-maps > styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  mapView: {
    alignSelf: "stretch",
    height: 450,
    margin: 30,
  },

  ipaText: {
    color: "coral",
  },

  stoutText: {
    color: "firebrick",
  },

  boldText: {
    fontWeight: "bold",
  },
});
```





NPX ET CREATE EXPO APP

MANIPULATION DES INPUTS

RÉCUPÉRATION DE DONNÉES

App.tsx M | styles.ts U

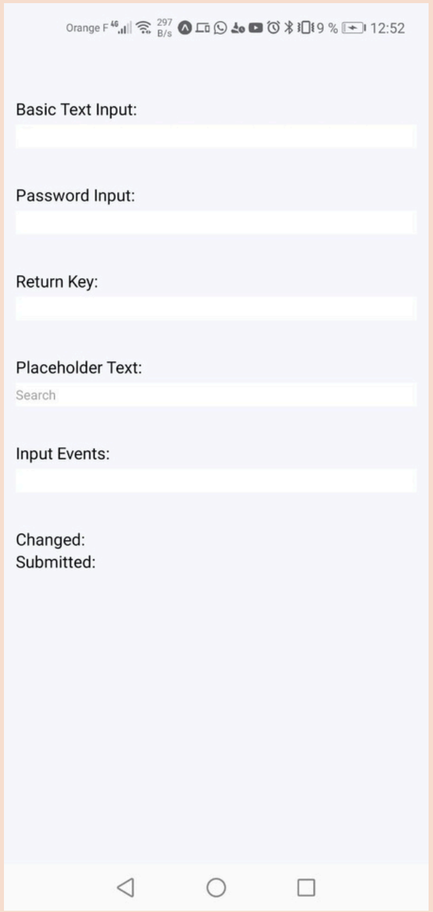
user-input > styles.ts > default > container

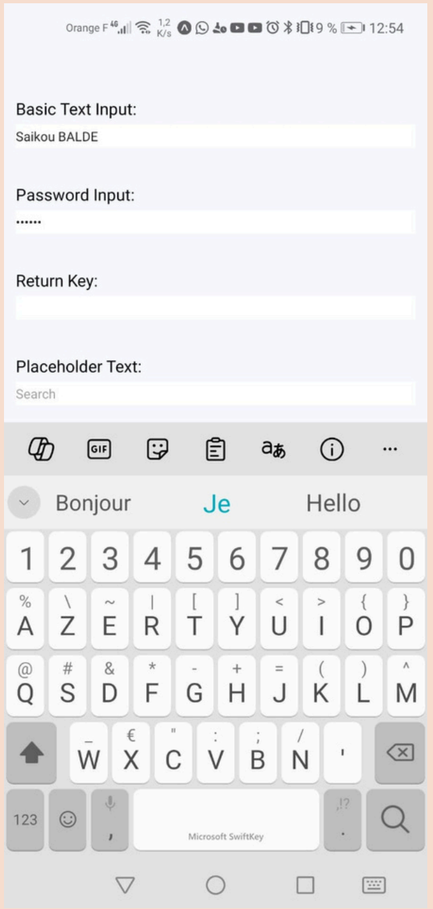
```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "ghostwhite",
    padding: 10,
    paddingTop: 80,
  },

  textInputContainer: {
    alignSelf: "stretch",
    marginBottom: 30,
  },

  textInputLabel: {
    marginBottom: 4,
  },

  textInput: {
    backgroundColor: "white",
    height: 20,
    fontSize: 11,
  },
});
```

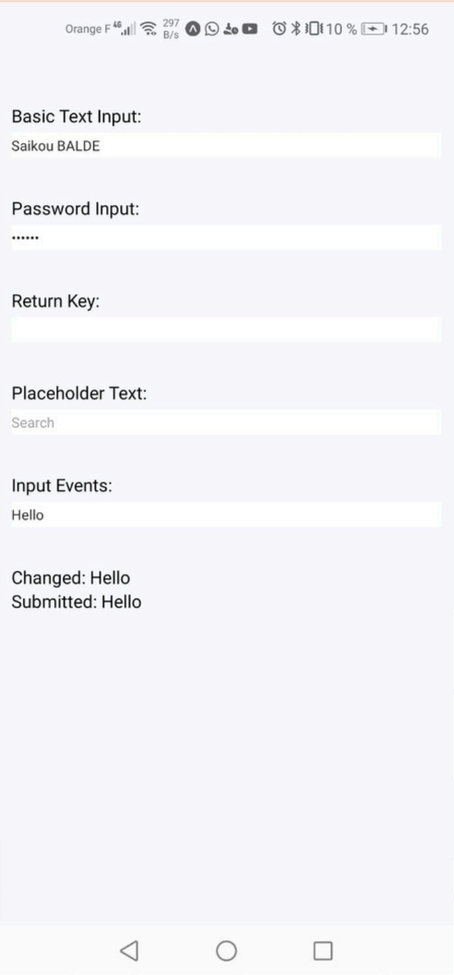
App.tsx M | styles.ts U

user-input > App.tsx > Input

```ts
import React, { useState } from "react";
import { Text, TextInput, TextInputProps, View } from "react-native";
import styles from "./styles";

type InputProps = TextInputProps & {
  label: string;
};

function Input(props: InputProps) {
  return (
    <View style={styles.textInputContainer}>
      <Text style={styles.textInputLabel}>{props.label}</Text>
      <TextInput style={styles.textInput} {...props} />
    </View>
  );
}

export default function CollectingTextInput() {
  const [changedText, setChangedText] = useState("");
  const [submittedText, setSubmittedText] = useState("");

  return (
    <View style={styles.container}>
      <Input label="Basic Text Input:" />
      <Input label="Password Input:" secureTextEntry />
      <Input label="Return Key:" returnKeyType="search" />
      <Input label="Placeholder Text:" placeholder="Search" />
      <Input
        label="Input Events:"
        onChangeText={(e) => {
          setChangedText(e);
        }}
        onSubmitEditing={(e) => {
          setSubmittedText(e.nativeEvent.text);
        }}
        onFocus={() => {
          setChangedText("");
          setSubmittedText("");
        }}
      />
      <Text>Changed: {changedText}</Text>
      <Text>Submitted: {submittedText}</Text>
    </View>
  );
}
```







NPX ET CREATE EXPO APP

MANIPULATION DES INPUTS

SÉLECTION PARMI UNE LISTE D`OPTIONS

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/user-input$ npx expo install @react-native-picker/picker
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

added 1 package, and audited 1194 packages in 4s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

user-input > TS styles.ts > [e] default > pickersBlock

```tsx
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: "column",
    backgroundColor: "ghostwhite",
    justifyContent: "center",
  },

  pickersBlock: {
    flex: 2,
    flexDirection: "row",
    justifyContent: "space-around",
    alignItems: "center",
  },

  pickerHeight: {
    height: 250,
  },

  pickerContainer: {
    flex: 1,
    flexDirection: "column",
    alignItems: "center",
    backgroundColor: "white",
    padding: 6,
    height: 240,
  },

  pickerLabel: {
    fontSize: 14,
    fontWeight: "bold",
  },

  picker: {
    width: 150,
    backgroundColor: "white",
  },

  selection: {
    flex: 1,
    textAlign: "center",
  },
});
```

user-input > Select.android.tsx > Select > props.items.map() callback

```tsx
import React from "react";
import { View, Text } from "react-native";
import { Picker } from "@react-native-picker/picker";
import styles from "./styles";
import { SelectProps } from "./SelectProps";

export default function Select(props: SelectProps) {
  return (
    <View>
      <Text style={styles.pickerLabel}>{props.label}</Text>
      <Picker style={styles.picker} {...props}>
        {props.items.map((i) => (
          <Picker.Item key={i.label} {...i} />
        ))}
      </Picker>
    </View>
  );
}
```

user-input > Select.ios.tsx > Select

```tsx
import React from "react";
import { View, Text } from "react-native";
import { Picker } from "@react-native-picker/picker";
import styles from "./styles";
import { SelectProps } from "./SelectProps";

export default function Select(props: SelectProps) {
  return (
    <View style={styles.pickerHeight}>
      <View style={styles.pickerContainer}>
        <Text style={styles.pickerLabel}>{props.label}</Text>
        <Picker style={styles.picker} {...props}>
          {props.items.map((i) => (
            <Picker.Item key={i.label} {...i} />
          ))}
        </Picker>
      </View>
    </View>
  );
}
```

user-input > App.tsx > SelectingOptions

```tsx
import React, { useState } from "react";
import { View, Text } from "react-native";
import styles from "./styles";
import Select from "./Select.android";

const sizes = [
  { label: "", value: null },
  { label: "S", value: "S" },
  { label: "M", value: "M" },
  { label: "L", value: "L" },
  { label: "XL", value: "XL" },
];

const garments = [
  { label: "", value: null, sizes: ["S", "M", "L", "XL"] },
  { label: "Socks", value: 1, sizes: ["S", "L"] },
  { label: "Shirt", value: 2, sizes: ["M", "XL"] },
  { label: "Pants", value: 3, sizes: ["S", "L"] },
  { label: "Hat", value: 4, sizes: ["M", "XL"] },
];

export default function SelectingOptions() {
  const [availableGarments, setAvailableGarments] = useState<typeof garments>(
    []
  );
  const [selectedSize, setSelectedSize] = useState<string | null>(null);
  const [selectedGarment, setSelectedGarment] = useState<number | null>(null);

  return (
    <View style={styles.container}>
      <View style={styles.pickersBlock}>
        <Select
          label="Size"
          items={sizes}
          selectedValue={selectedSize}
          onValueChange={(size: string) => {
            setSelectedSize(size);
            setSelectedGarment(null);
            setAvailableGarments(
              garments.filter((i) => i.sizes.includes(size))
            );
          }}
        />
        <Select
          label="Garment"
          items={availableGarments}
          selectedValue={selectedGarment}
          onValueChange={(garment: number) => {
```

user-input > App.tsx > SelectingOptions

```tsx
export default function SelectingOptions() {
          }}
        />
        <Select
          label="Garment"
          items={availableGarments}
          selectedValue={selectedGarment}
          onValueChange={(garment: number) => {
            setSelectedGarment(garment);
          }}
        />
      </View>
      <Text style={styles.selection}>
        {selectedSize &&
          selectedGarment &&
          `${selectedSize} ${
            garments.find((i) => i.value === selectedGarment)?.label
          }`}
      </Text>
    </View>
  );
}
```

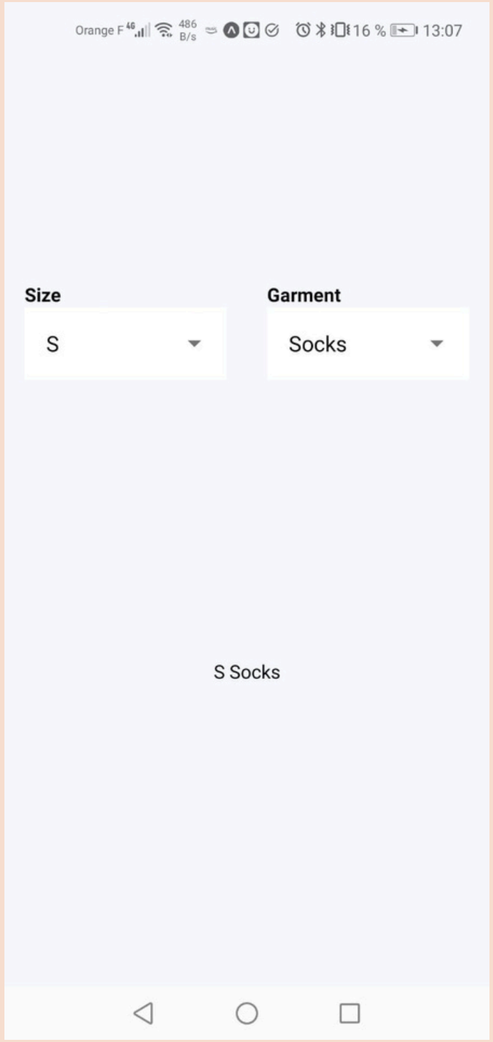
NPX ET CREATE EXPO APP

MANIPULATION DES INPUTS

SÉLECTION PARMI UNE LISTE D`OPTIONS

user-input > App.tsx > SelectingOptions

```tsx
import React, { useState } from "react";
import { View, Text } from "react-native";
import styles from "./styles";
import Select from "./Select.android";

const sizes = [
  { label: "", value: null },
  { label: "S", value: "S" },
  { label: "M", value: "M" },
  { label: "L", value: "L" },
  { label: "XL", value: "XL" },
];

const garments = [
  { label: "", value: null, sizes: ["S", "M", "L", "XL"] },
  { label: "Socks", value: 1, sizes: ["S", "L"] },
  { label: "Shirt", value: 2, sizes: ["M", "XL"] },
  { label: "Pants", value: 3, sizes: ["S", "L"] },
  { label: "Hat", value: 4, sizes: ["M", "XL"] },
];

export default function SelectingOptions() {
  const [availableGarments, setAvailableGarments] = useState<typeof garments>(
    []
  );
  const [selectedSize, setSelectedSize] = useState<string | null>(null);
  const [selectedGarment, setSelectedGarment] = useState<number | null>(null);

  return (
    <View style={styles.container}>
      <View style={styles.pickersBlock}>
        <Select
          label="Size"
          items={sizes}
          selectedValue={selectedSize}
          onValueChange={(size: string) => {
            setSelectedSize(size);
            setSelectedGarment(null);
            setAvailableGarments(
              garments.filter((i) => i.sizes.includes(size))
            );
          }}
        />
        <Select
          label="Garment"
          items={availableGarments}
          selectedValue={selectedGarment}
          onValueChange={(garment: number) => {
            setSelectedGarment(garment);
          }}
        />
      </View>
      <Text style={styles.selection}>
        {selectedSize &&
          selectedGarment &&
          `${selectedSize} ${
            garments.find((i) => i.value === selectedGarment)?.label
          }`}
      </Text>
    </View>
  );
}
```



NPX ET CREATE EXPO APP

MANIPULATION DES INPUTS

BASCULER ENTRE ACTIVÉ ET DÉSACTIVÉ

user-input > TS styles.ts > default

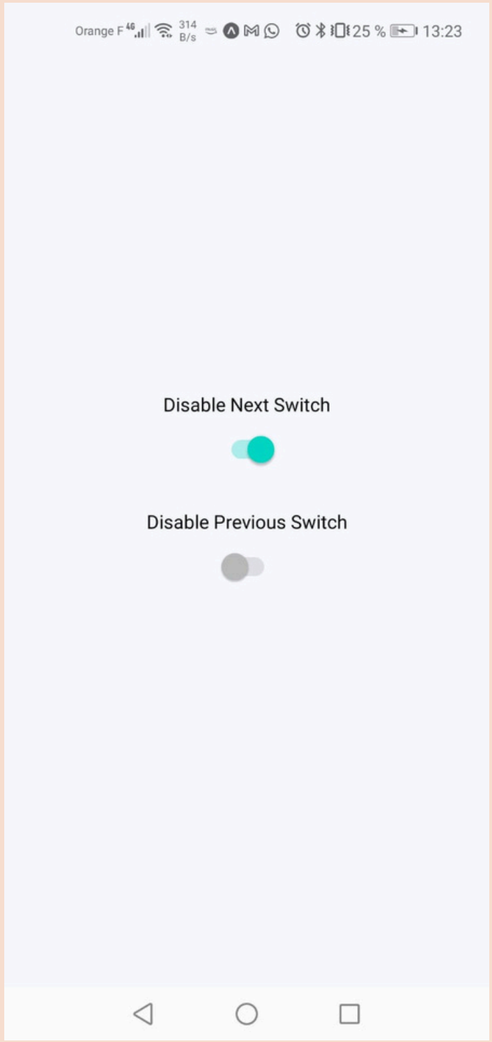
```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  customSwitch: {
    alignItems: "center",
    margin: 10,
  },
});
```

user-input > Switch.tsx > CustomSwitch

```tsx
import React from "react";
import { View, Text, Switch, SwitchProps } from "react-native";
import styles from "./styles";

type CustomSwitchProps = SwitchProps & {
  label: string;
};

export default function CustomSwitch(props: CustomSwitchProps) {
  return (
    <View style={styles.customSwitch}>
      <Text>{props.label}</Text>
      <Switch {...props} />
    </View>
  );
}
```

user-input > App.tsx > TogglingOnAndOff

```tsx
import React, { useState } from "react";
import { View } from "react-native";
import styles from "./styles";
import Switch from "./Switch";

export default function TogglingOnAndOff() {
  const [first, setFirst] = useState(false);
  const [second, setSecond] = useState(false);

  return (
    <View style={styles.container}>
      <Switch
        label="Disable Next Switch"
        value={first}
        disabled={second}
        onValueChange={setFirst}
      />
      <Switch
        label="Disable Previous Switch"
        value={second}
        disabled={first}
        onValueChange={setSecond}
      />
    </View>
  );
}
```



NPX ET CREATE EXPO APP

MANIPULATION DES INPUTS

COLLECTE DE LA DATE/HEURE

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/user-input$ npx expo install @react-native-community/datetimepicker
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

added 1 package, and audited 1195 packages in 4s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

user-input > styles.ts > default > datePickerLabel

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  datePickerContainer: {
    width: 260,
    margin: 20,
  },

  datePickerLabel: {
    fontSize: 18,
    fontWeight: "bold",
  },
});
```

user-input > TimePickerProps.ts > TimePickerProps

```ts
export type TimePickerProps = {
    label: string;
    value: Date;
    onChange: (date: Date) => void;
};
```

user-input > DatePickerProps.ts > DatePickerProps

```ts
export type DatePickerProps = {
    label: string;
    value: Date;
    onChange: (date: Date) => void;
};
```

user-input > TimePicker.android.tsx > TimePicker

```tsx
import React from "react";
import { Text, View } from "react-native";
import { DateTimePickerAndroid } from "@react-native-community/datetimepicker";
import styles from "./styles";
import { TimePickerProps } from "./TimePickerProps";

export default function TimePicker({
  label,
  value,
  onChange,
}: TimePickerProps) {
  return (
    <View style={styles.datePickerContainer}>
      <Text style={styles.datePickerLabel}>{label}</Text>
      <Text
        onPress={() =>
          DateTimePickerAndroid.open({
            value: value,
            mode: "time",
            is24Hour: true,
            onChange: (event, date) => {
              if (event.type === "set" && date) {
                onChange(date);
              }
            },
          })
        }
      >
        {value.toLocaleTimeString()}
      </Text>
    </View>
  );
}
```

user-input > DatePicker.android.tsx > DatePicker > <function>

```tsx
import React from "react";
import { Text, View } from "react-native";
import { DateTimePickerAndroid } from "@react-native-community/datetimepicker";
import styles from "./styles";
import { DatePickerProps } from "./DatePickerProps";

export default function DatePicker({
  label,
  value,
  onChange,
}: DatePickerProps) {
  return (
    <View style={styles.datePickerContainer}>
      <Text style={styles.datePickerLabel}>{label}</Text>
      <Text
        onPress={() => {
          DateTimePickerAndroid.open({
            value: value,
            mode: "date",
            onChange: (event, date) => {
              if (event.type === "set" && date) {
                onChange(date);
              }
            },
          });
        }}
      >
        {value.toLocaleDateString()}
      </Text>
    </View>
  );
}
```

user-input > App.tsx > CollectingDateTimeInput

```tsx
import React, { useState } from "react";
import { View } from "react-native";
import DatePicker from "./DatePicker";
import TimePicker from "./TimePicker";
import styles from "./styles";

export default function CollectingDateTimeInput() {
  const [date, setDate] = useState(new Date());
  const [time, setTime] = useState(new Date());

  return (
    <View style={styles.container}>
      <DatePicker
        label="Pick a date, any date:"
        value={date}
        onChange={setDate}
      />
      <TimePicker
        label="Pick a time, any time:"
        value={time}
        onChange={setTime}
      />
    </View>
  );
}
```

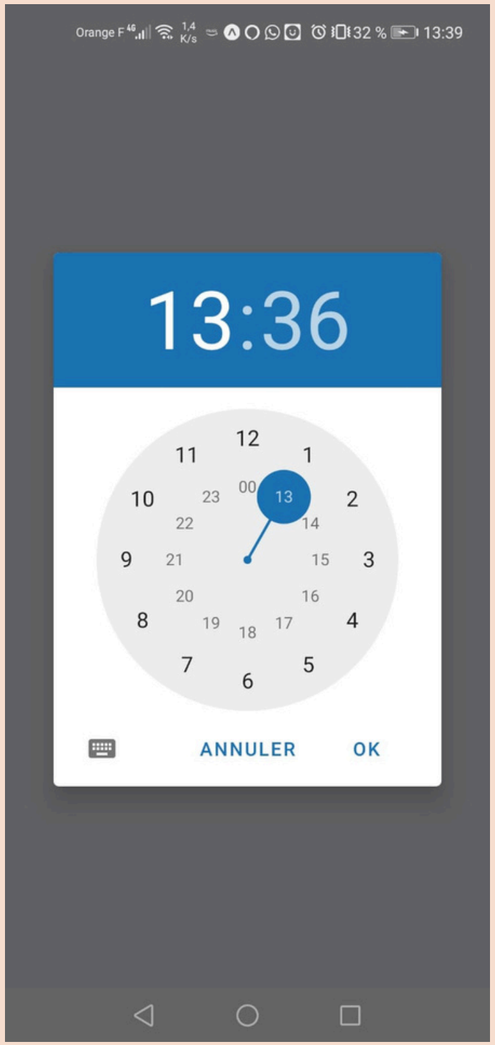
NPX ET CREATE EXPO APP

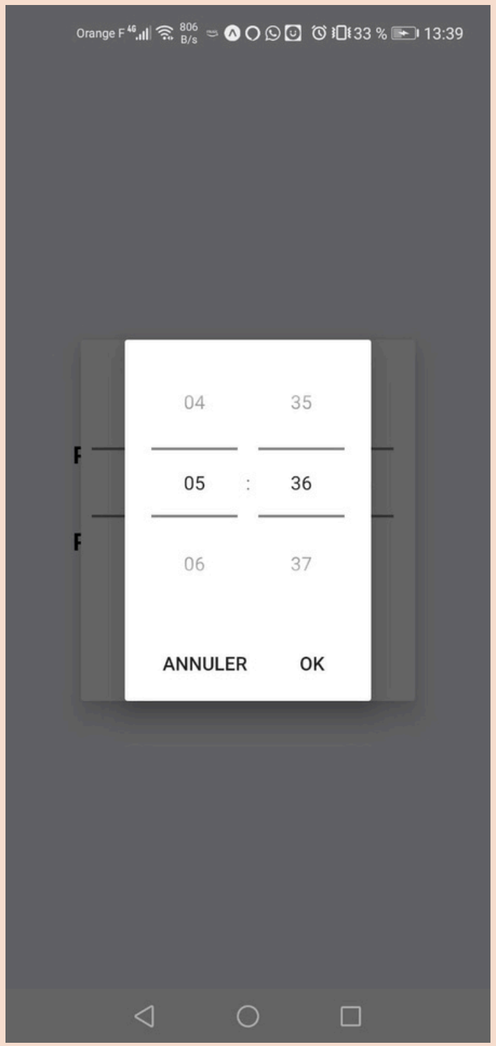
MANIPULATION DES INPUTS

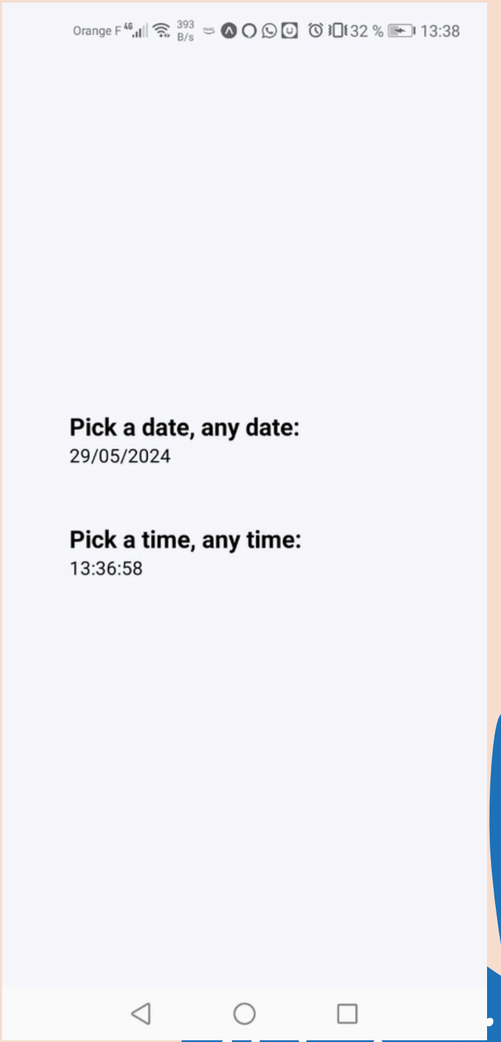
COLLECTE DE LA DATE/HEURE

user-input > App.tsx > ...

```tsx
import React, { useState } from "react";
import { View } from "react-native";
import DatePicker from "./DatePicker.android";
import TimePicker from "./TimePicker.android";
import styles from "./styles";

export default function CollectingDateTimeInput() {
  const [date, setDate] = useState(new Date());
  const [time, setTime] = useState(new Date());

  return (
    <View style={styles.container}>
      <DatePicker
        label="Pick a date, any date:"
        value={date}
        onChange={setDate}
      />
      <TimePicker
        label="Pick a time, any time:"
        value={time}
        onChange={setTime}
      />
    </View>
  );
}
```







NPX ET CREATE EXPO APP

RÉPONDRE AUX GESTES DES UTILISATEURS

FAIRE DÉFILER AVEC LES DOIGTS

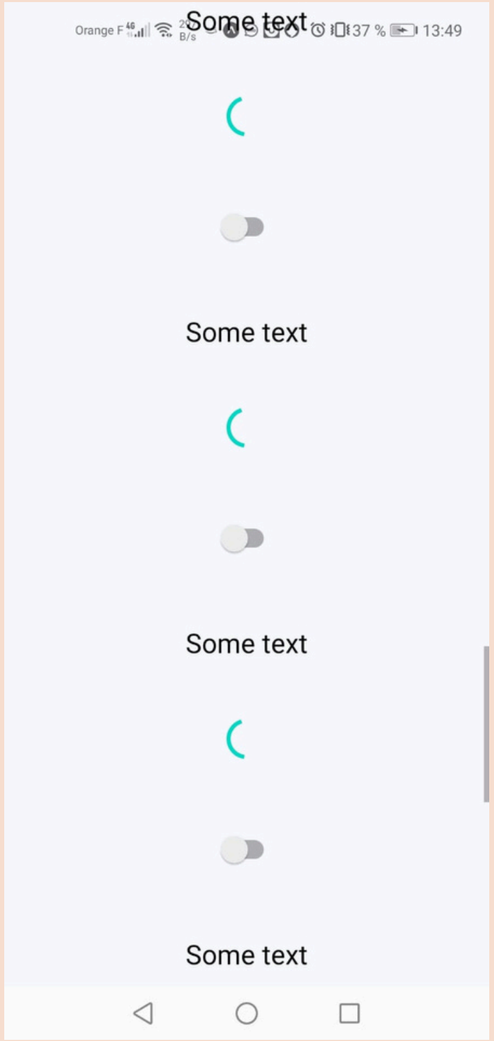
```
✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd user-gesture
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
```

user-gesture > TS styles.ts > default > container

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  text: {
    fontSize: 20,
    textAlign: "center",
  },

  scroll: {
    height: 1,
    alignSelf: "stretch",
  },

  scrollItem: {
    margin: 20,
    alignSelf: "center",
  },
});
```

user-gesture > App.tsx > ...

```tsx
import React from "react";
import {
  Text,
  ScrollView,
  ActivityIndicator,
  Switch,
  View,
} from "react-native";

import styles from "./styles";

export default function App() {
  return (
    <View style={styles.container}>
      <ScrollView style={styles.scroll}>
        {new Array(20).fill(null).map((v, i) => (
          <View key={i}>
            <Text style={[styles.scrollItem, styles.text]}>Some text</Text>
            <ActivityIndicator style={styles.scrollItem} size="large" />
            <Switch style={styles.scrollItem} />
          </View>
        ))}
      </ScrollView>
    </View>
  );
}
```



NPX ET CREATE EXPO APP

RÉPONDRE AUX GESTES DES UTILISATEURS

EVÈNEMENT TOUCH TACTILES

user-gesture > styles.ts > default > container

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  button: {
    padding: 10,
    margin: 5,
    backgroundColor: "azure",
    borderWidth: 1,
    borderRadius: 4,
    borderColor: "slategrey",
  },

  buttonText: {
    color: "slategrey",
  },
});
```

user-gesture > PressableButton.tsx > PressableButton

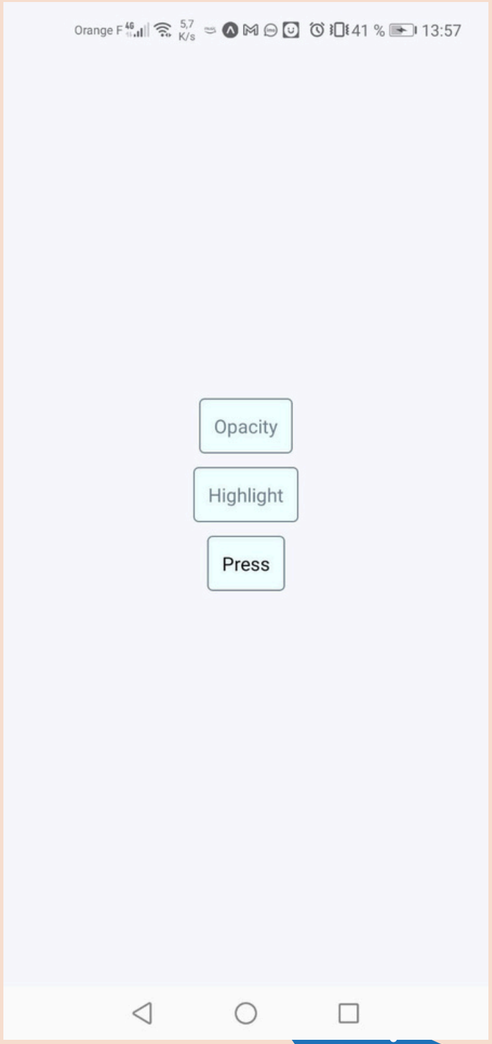
```tsx
import React, { useState } from "react";
import { Pressable, Text } from "react-native";
import styles from "./styles";

const PressableButton = () => {
  const [text, setText] = useState("Not Pressed");

  return (
    <Pressable
      onPressIn={() => setText("Pressed")}
      onPressOut={() => setText("Press")}
      onLongPress={() => {
        setText("Long Pressed");
      }}
      delayLongPress={500}
      style={({ pressed }) => [
        {
          opacity: pressed ? 0.5 : 1,
        },
        styles.button,
      ]}
    >
      <Text>{text}</Text>
    </Pressable>
  );
};

export default PressableButton;
```

user-gesture > App.tsx > App

```tsx
import React from "react";
import { View } from "react-native";
import styles from "./styles";
import { OpacityButton, HighlightButton } from "./Button";
import PressableButton from "./PressableButton";

export default function App() {
  return (
    <View style={styles.container}>
      <OpacityButton onPress={() => {}} label="Opacity" />
      <HighlightButton onPress={() => {}} label="Highlight" />
      <PressableButton />
    </View>
  );
}
```

user-gesture > Button.tsx > ...

```tsx
import React from "react";
import { Text, TouchableOpacity, TouchableHighlight } from "react-native";
import styles from "./styles";

type ButtonProps = {
  label: string;
  onPress: () => void;
};

export const OpacityButton = ({ label, onPress }: ButtonProps) => {
  return (
    <TouchableOpacity
      style={styles.button}
      onPress={onPress}
      activeOpacity={0.5}
    >
      <Text style={styles.buttonText}>{label}</Text>
    </TouchableOpacity>
  );
};

export const HighlightButton = ({ label, onPress }: ButtonProps) => {
  return (
    <TouchableHighlight
      style={styles.button}
      underlayColor="rgba(112,128,144,0.3)"
      onPress={onPress}
    >
      <Text style={styles.buttonText}>{label}</Text>
    </TouchableHighlight>
  );
};
```



NPX ET CREATE EXPO APP

RÉPONDRE AUX GESTES DES UTILISATEURS

EVÈNEMENT TOUCH TACTILES

NPX ET CREATE EXPO APP

RÉPONDRE AUX GESTES DES UTILISATEURS

UTILISATION DE COMPOSANTS GLISSANTS ET ANNULABLES

user-gesture > TS styles.ts > default > swipeBlank

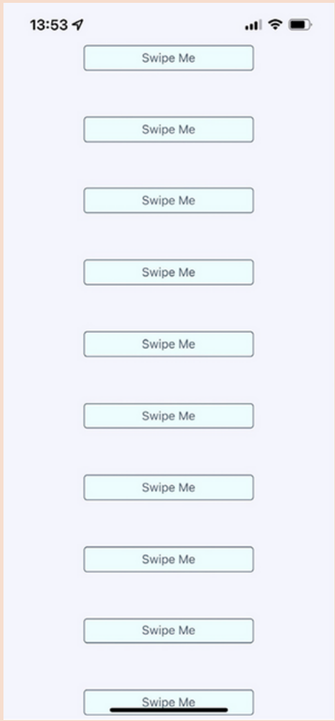
```ts
export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "space-between",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  swipeContainer: {
    flex: 1,
    flexDirection: "row",
    width: 200,
    height: 30,
    marginTop: 50,
  },

  swipeItem: {
    height: 30,
    backgroundColor: "azure",
    justifyContent: "center",
    borderWidth: 1,
    borderRadius: 4,
    borderColor: "slategrey",
  },

  swipeItemText: {
    textAlign: "center",
    color: "slategrey",
  },

  swipeBlank: {
    height: 30,
  },
});
```

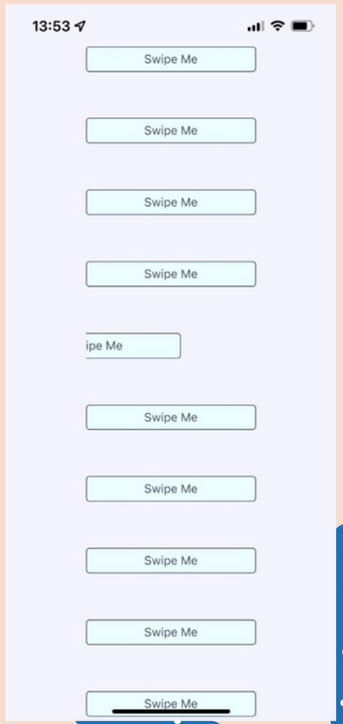
user-gesture > Swipeable.tsx > ...

```tsx
import React from "react";
import {
  View,
  ScrollView,
  Text,
  NativeSyntheticEvent,
  NativeScrollEvent,
} from "react-native";
import styles from "./styles";

type SwipeableProps = {
  name: string;
  width: number;
  onSwipe: () => void;
};

export default function Swipeable({ name, width, onSwipe }: SwipeableProps) {
  function onScroll(e: NativeSyntheticEvent<NativeScrollEvent>) {
    console.log(e.nativeEvent.contentOffset.x);
    e.nativeEvent.contentOffset.x >= width && onSwipe();
  }

  return (
    <View style={styles.swipeContainer}>
      <ScrollView
        horizontal
        snapToInterval={width}
        showsHorizontalScrollIndicator={false}
        scrollEventThrottle={10}
        onScroll={onScroll}
      >
        <View style={[styles.swipeItem, { width }]}>
          <Text style={styles.swipeItemText}>{name}</Text>
        </View>

        <View style={[styles.swipeBlank, { width }]} />
      </ScrollView>
    </View>
  );
}
```

user-gesture > App.tsx > SwipableAndCancellable

```tsx
import React, { useState } from "react";
import { View } from "react-native";
import styles from "./styles";
import Swipeable from "./Swipeable";

export default function SwipableAndCancellable() {
  const [items, setItems] = useState(
    new Array(10).fill(null).map((v, id) => ({ id, name: "Swipe Me" }))
  );

  function onSwipe(id: number) {
    return () => {
      setItems(items.filter((item) => item.id !== id));
    };
  }

  return (
    <View style={styles.container}>
      {items.map((item) => (
        <Swipeable
          key={item.id}
          onSwipe={onSwipe(item.id)}
          name={item.name}
          width={200}
        />
      ))}
    </View>
  );
}
```





NPX ET CREATE EXPO APP

LA BARRE DE PROGRESSION

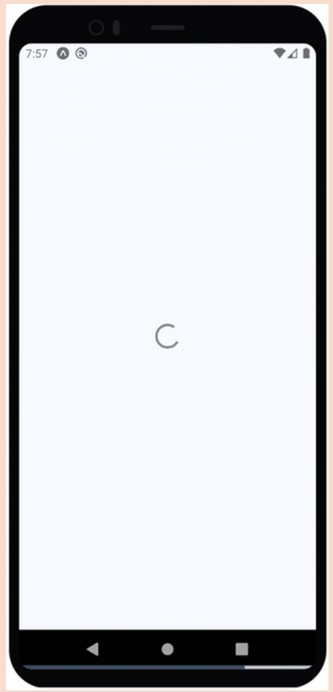
INDIQUER LES PROGRÈS

progress-bar > TS styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },
});
```

progress-bar > App.tsx > ...

```tsx
import React from "react";
import { View, ActivityIndicator } from "react-native";
import styles from "./styles";
export default function App() {
  return (
    <View style={styles.container}>
      <ActivityIndicator size="large" />
    </View>
  );
}
```



NPX ET CREATE EXPO APP

LA BARRE DE PROGRESSION

MESURER L'EVOLUTION DE LA BARRE DE PROGRESSION

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/progress-bar$ npm i react-native-progress

added 14 packages, and audited 1207 packages in 7s

139 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

progress-bar > TS styles.ts > [o] default

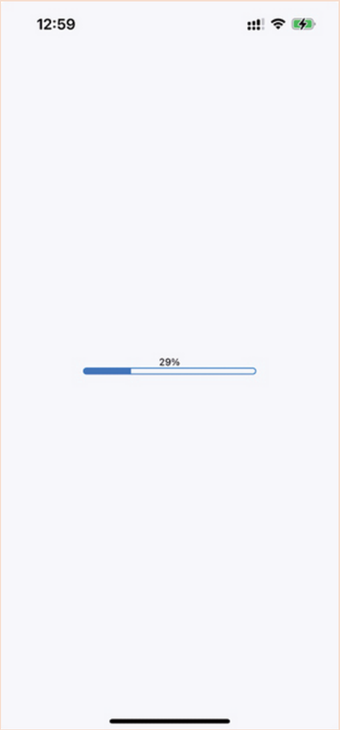
```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  progress: {
    width: 200,
  },

  progressText: {
    fontSize: 11,
    textAlign: "center",
  },
});
```

progress-bar > ProgressBar.tsx > ProgressBar

```tsx
import React from "react";
import { View, Text } from "react-native";
import styles from "./styles";
import * as Progress from "react-native-progress";

type ProgressBarProps = {
  progress: number;
};

export default function ProgressBar({ progress }: ProgressBarProps) {
  return (
    <View style={styles.progress}>
      <Text style={styles.progressText}>{Math.round(progress * 100)}%</Text>
      <Progress.Bar width={200} useNativeDriver={true} progress={progress} />
    </View>
  );
}
```

progress-bar > App.tsx > MeasuringProgress > useEffect() callback > upda

```tsx
import React, { useState, useEffect } from "react";
import { View } from "react-native";
import styles from "./styles";
import ProgressBar from "./ProgressBar";

export default function MeasuringProgress() {
  const [progress, setProgress] = useState(0);

  useEffect(() => {
    let timeoutRef: NodeJS.Timeout | null = null;

    function updateProgress() {
      setProgress((currentProgress) => {
        if (currentProgress < 1) {
          return currentProgress + 0.01;
        } else {
          return 0;
        }
      });

      timeoutRef = setTimeout(updateProgress, 100);
    }

    updateProgress();

    return () => {
      timeoutRef && clearTimeout(timeoutRef);
    };
  }, []);

  return (
    <View style={styles.container}>
      <ProgressBar progress={progress} />
    </View>
  );
}
```



NPX ET CREATE EXPO APP

LA BARRE DE PROGRESSION

BARRE DE PROGRESSION PAR ÉTAPE

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/progress-bar$ npm i @react-navigation/native @react-navigation/native-stack

added 18 packages, and audited 1225 packages in 7s

140 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

progress-bar > TS styles.ts > default > container

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  nav: {
    backgroundColor: "lightgray",
  },

  title: {
    flex: 1,
    fontSize: 16,
    fontWeight: "bold",
  },

  content: {
    fontSize: 23,
    fontWeight: "bold",
  },

  progress: {
    justifyContent: "center",
    alignItems: "center",
    padding: 1,
  },
});
```

progress-bar > TS router.ts > Routes

```ts
export type Routes = {
    First: { progress: number };
    Second: { progress: number };
    Third: { progress: number };
    Fourth: { progress: number };
  };
```

progress-bar > First.tsx > First

```tsx
import React from "react";
import { View, Text } from "react-native";
import styles from "./styles";

export default function First() {
  return (
    <View style={styles.container}>
      <Text style={styles.content}>First Content</Text>
    </View>
  );
}
```

progress-bar > Second.tsx > Second

```tsx
import React from "react";
import { View, Text } from "react-native";
import styles from "./styles";
import ProgressBar from "./ProgressBar";

export default function Second() {
  return (
    <View style={styles.container}>
      <Text style={styles.content}>Second Content</Text>
    </View>
  );
}
```

progress-bar > Third.tsx > Third

```tsx
import React from "react";
import { View, Text } from "react-native";
import styles from "./styles";
import ProgressBar from "./ProgressBar";

export default function Third() {
  return (
    <View style={styles.container}>
      <Text style={styles.content}>Third Content</Text>
    </View>
  );
}
```

progress-bar > Fourth.tsx > Fourth

```tsx
import React from "react";
import { View, Text } from "react-native";
import styles from "./styles";

export default function Fourth() {
  return (
    <View style={styles.container}>
      <Text style={styles.content}>Fourth Content</Text>
    </View>
  );
}
```

progress-bar > App.tsx > routes

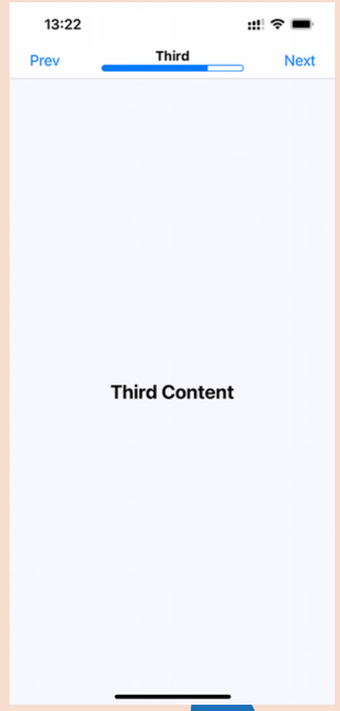
```tsx
import React from "react";
import { View, Text, Button } from "react-native";
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import First from "./First";
import Second from "./Second";
import Third from "./Third";
import Fourth from "./Fourth";
import ProgressBar from "./ProgressBar";
import styles from "./styles";
import { Routes } from "./router";

const Stack = createNativeStackNavigator<Routes>();

const routes = [
  { name: "First", component: First },
  { name: "Second", component: Second },
  { name: "Third", component: Third },
  { name: "Fourth", component: Fourth },
] as const;

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator
        screenOptions={({ route, navigation }) => {
          const currentRouteIndex = routes
            .map((r) => r.name)
            .indexOf(route.name);
          const prevRouteName = routes[currentRouteIndex - 1]?.name;
          const nextRouteName = routes[currentRouteIndex + 1]?.name;

          return {
            headerBackVisible: false,
            headerTitleAlign: "center",
            headerTitle: () => (
              <View style={styles.progress}>
                <Text style={styles.title}>{route.name}</Text>
                <ProgressBar progress={route.params.progress} />
              </View>
```

progress-bar > App.tsx > ...

```tsx
function App() {
        screenOptions={({ route, navigation }) => {
          const currentRouteIndex = routes
            .map((r) => r.name)
            .indexOf(route.name);
          const prevRouteName = routes[currentRouteIndex - 1]?.name;
          const nextRouteName = routes[currentRouteIndex + 1]?.name;

          return {
            headerBackVisible: false,
            headerTitleAlign: "center",
            headerTitle: () => (
              <View style={styles.progress}>
                <Text style={styles.title}>{route.name}</Text>
                <ProgressBar progress={route.params.progress} />
              </View>
            ),
            headerLeft: () => (
              <Button
                title="Prev"
                disabled={currentRouteIndex === 0}
                onPress={() => navigation.navigate(prevRouteName)}
              />
            ),
            headerRight: () => (
              <Button
                title="Next"
                disabled={currentRouteIndex === 3}
                onPress={() => navigation.navigate(nextRouteName)}
              />
            ),
          };
        }}
      >
        {routes.map((routeProps, index) => (
          <Stack.Screen
            key={routeProps.name}
            {...routeProps}
            initialParams={{ progress: (index + 1) / routes.length }}
          />
        ))}
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

NPX ET CREATE EXPO APP

LA BARRE DE PROGRESSION



NPX ET CREATE EXPO APP

AFFICHAGE DES ÉCRANS MODAUX

OBTENIR LA CONFIRMATION DE L'UTILISATEUR

```
✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd modal-screen
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
```

App.tsx M | ConfirmationModal.tsx U | styles.ts U

modal-screen > styles.ts > default > modalInner

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  text: {
    color: "slategrey",
  },

  modalContainer: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },

  modalInner: {
    backgroundColor: "azure",
    padding: 20,
    borderWidth: 1,
    borderColor: "lightsteelblue",
    borderRadius: 2,
    alignItems: "center",
  },

  modalText: {
    fontSize: 16,
    margin: 5,
    color: "slategrey",
  },

  modalButton: {
    fontWeight: "bold",
    margin: 5,
    color: "slategrey",
  },
});
```

App.tsx M | ConfirmationModal.tsx U | styles.ts U

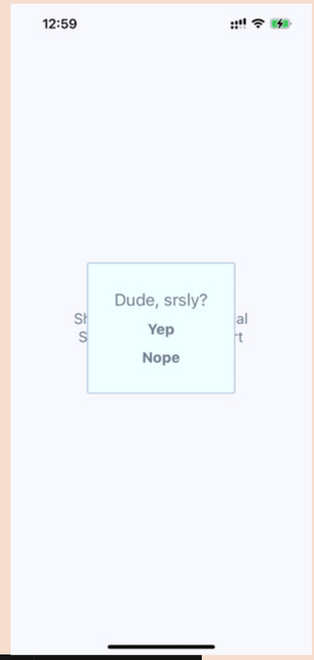
modal-screen > App.tsx > App > toggleAlert

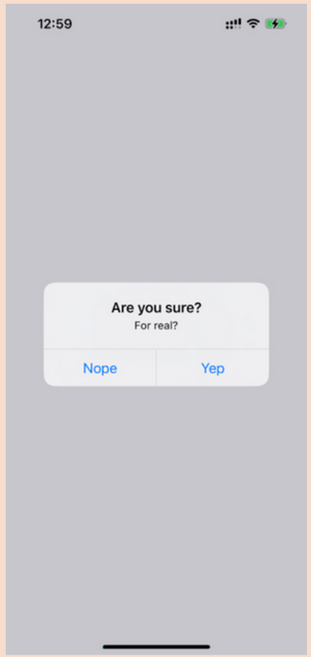
```tsx
import React, { useState } from "react";
import { View, Text, Alert } from "react-native";
import ConfirmationModal from "./ConfirmationModal";

import styles from "./styles";

export default function App() {
  const [modalVisible, setModalVisible] = useState(false);

  function toggleModal() {
    setModalVisible(!modalVisible);
  }

  function toggleAlert() {
    Alert.alert("Are you sure?", "For real?", [
      { text: "Nope", onPress: () => {} },
      { text: "Yep", onPress: () => {} },
    ]);
  }

  return (
    <View style={styles.container}>
      <ConfirmationModal
        animationType="fade"
        visible={modalVisible}
        onPressConfirm={toggleModal}
        onPressCancel={toggleModal}
      />
      <Text style={styles.text} onPress={toggleModal}>
        Show Confirmation Modal
      </Text>
      <Text style={styles.text} onPress={toggleAlert}>
        Show Confirmation Alert
      </Text>
    </View>
  );
}
```

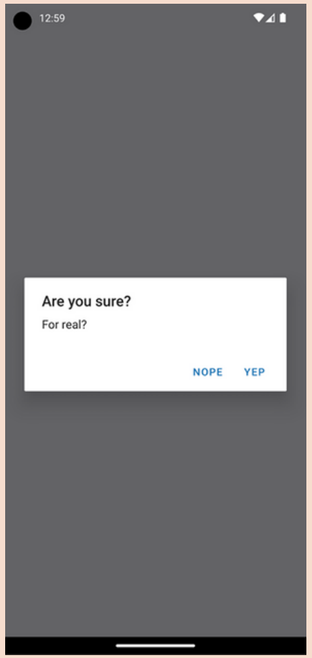
App.tsx M | ConfirmationModal.tsx U | styles.ts U

modal-screen > ConfirmationModal.tsx > ConfirmationModal

```tsx
import React from "react";
import { View, Text, Modal, ModalProps } from "react-native";
import styles from "./styles";

type Props = ModalProps & {
  onPressConfirm: () => void;
  onPressCancel: () => void;
};

export default function ConfirmationModal({
  onPressConfirm,
  onPressCancel,
  ...modalProps
}: Props) {
  return (
    <Modal transparent onRequestClose={() => {}} {...modalProps}>
      <View style={styles.modalContainer}>
        <View style={styles.modalInner}>
          <Text style={styles.modalText}>Dude, srsly?</Text>
          <Text style={styles.modalButton} onPress={onPressConfirm}>
            Yep
          </Text>
          <Text style={styles.modalButton} onPress={onPressCancel}>
            Nope
          </Text>
        </View>
      </View>
    </Modal>
  );
}
```







NPX ET CREATE EXPO APP

AFFICHAGE DES ÉCRANS MODAUX

CONFIRMATION D'ERREUR

App.tsx M | ErrorModal.tsx U | styles.ts U

modal-screen > styles.ts > default

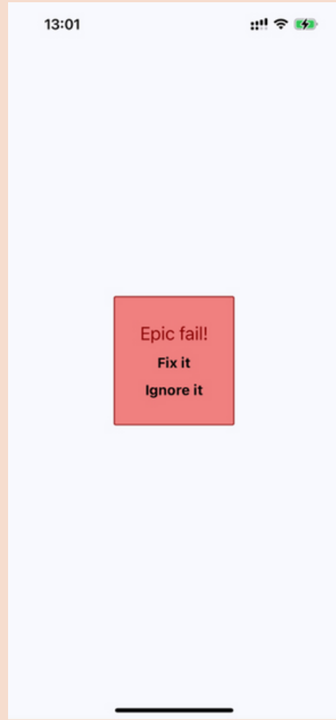
```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  text: {
    color: "slategrey",
  },

  modalContainer: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },

  modalInner: {
    backgroundColor: "azure",
    padding: 20,
    borderWidth: 1,
    borderColor: "lightsteelblue",
    borderRadius: 2,
    alignItems: "center",
  },

  modalInnerError: {
    backgroundColor: "lightcoral",
    borderColor: "darkred",
  },

  modalText: {
    fontSize: 16,
    margin: 5,
    color: "slategrey",
  },

  modalTextError: {
    fontSize: 18,
    color: "darkred",
  },

  modalButton: {
    fontWeight: "bold",
    margin: 5,
    color: "slategrey",
  },

  modalButtonError: {
    color: "black",
  },
});
```

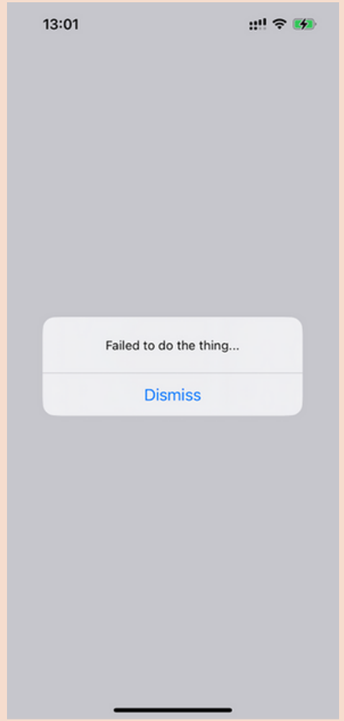
modal-screen > ErrorModal.tsx > ErrorModal

```tsx
import React from "react";
import { View, Text, Modal, ModalProps } from "react-native";
import styles from "./styles";

const innerViewStyle = [styles.modalInner, styles.modalInnerError];
const textStyle = [styles.modalText, styles.modalTextError];
const buttonStyle = [styles.modalButton, styles.modalButtonError];

type Props = ModalProps & {
  onPressConfirm: () => void;
  onPressCancel: () => void;
};

export default function ErrorModal({
  onPressConfirm,
  onPressCancel,
  ...modalProps
}: Props) {
  return (
    <Modal transparent onRequestClose={() => {}} {...modalProps}>
      <View style={styles.modalContainer}>
        <View style={innerViewStyle}>
          <Text style={textStyle}>Epic fail!</Text>
          <Text style={buttonStyle} onPress={onPressConfirm}>
            Fix it
          </Text>
          <Text style={buttonStyle} onPress={onPressCancel}>
            Ignore it
          </Text>
        </View>
      </View>
    </Modal>
  );
}
```

modal-screen > App.tsx > App

```tsx
import React, { useState } from "react";
import { View, Text, Alert } from "react-native";
import styles from "./styles";
import ErrorModal from "./ErrorModal";

export default function App() {
  const [modalVisible, setModalVisible] = useState(false);

  function toggleModal() {
    setModalVisible(!modalVisible);
  }

  function toggleAlert() {
    Alert.alert("", "Failed to do the thing...", [
      {
        text: "Dismiss",
      },
    ]);
  }

  return (
    <View style={styles.container}>
      <ErrorModal
        animationType="fade"
        visible={modalVisible}
        onPressConfirm={toggleModal}
        onPressCancel={toggleModal}
      />

      <Text style={styles.text} onPress={toggleModal}>
        Show Error Modal
      </Text>
      <Text style={styles.text} onPress={toggleAlert}>
        Show Error Alert
      </Text>
    </View>
  );
}
```





NPX ET CREATE EXPO APP

AFFICHAGE DES ÉCRANS MODAUX

NOTIFICATIONS PASSIVES

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/modal-screen$ npm i react-native-root-siblings react-native

added 5 packages, and audited 1198 packages in 8s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

App.tsx M | styles.ts U

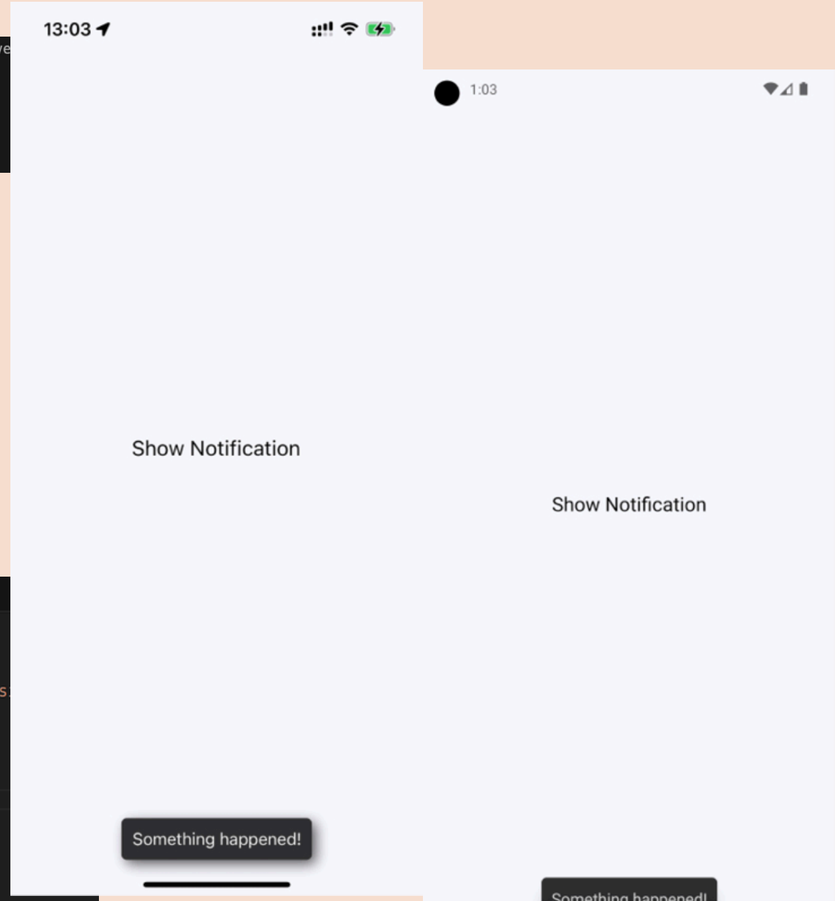
modal-screen > styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  notificationContainer: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },

  notificationInner: {
    backgroundColor: "azure",
    padding: 20,
    borderWidth: 1,
    borderColor: "lightsteelblue",
    borderRadius: 2,
    alignItems: "center",
  },
});
```

App.tsx M | styles.ts U

modal-screen > App.tsx > PassiveNotifications

```tsx
import React from "react";
import { Text, View } from "react-native";
import { RootSiblingParent } from "react-native-root-s
import styles from "./styles";
import Toast from "react-native-root-toast";

export default function PassiveNotifications() {
  return (
    <RootSiblingParent>
      <View style={styles.container}>
        <Text
          onPress={() => {
            Toast.show("Something happened!", {
              duration: Toast.durations.LONG,
            });
          }}
        >
          Show Notification
        </Text>
      </View>
    </RootSiblingParent>
  );
}
```



NPX ET CREATE EXPO APP

AFFICHAGE DES ÉCRANS MODAUX

MODAUX D`ACTIVITÉ

App.tsx 1, M | Activity.tsx U | styles.ts U

modal-screen > styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  modalContainer: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "rgba(0, 0, 0, 0.2)",
  },
});
```

App.tsx 1, M | Activity.tsx U | styles.ts U

modal-screen > Activity.tsx > Activity

```tsx
import React from "react";
import { View, Modal, ActivityIndicator } from "react-native";
import styles from "./styles";

type ActivityProps = {
  visible: boolean;
  size?: "small" | "large";
};

export default function Activity({ visible, size = "large" }: ActivityProps) {
  return (
    <Modal visible={visible} transparent>
      <View style={styles.modalContainer}>
        <ActivityIndicator size={size} />
      </View>
    </Modal>
  );
}
```

App.tsx M | Activity.tsx U | styles.ts U

modal-screen > App.tsx > App > onPress

```tsx
import React, { useState } from "react";
import { Text, View } from "react-native";
import styles from "./styles";
import Activity from "./Activity";

export default function App() {
  const [fetching, setFetching] = useState(false);
  const [promise, setPromise] = useState(Promise.resolve());

  function onPress() {
    setPromise(
      new Promise((resolve) => setTimeout(resolve, 3000)).then(() => {
        setFetching(false);
      })
    );
    setFetching(true);
  }

  return (
    <View style={styles.container}>
      <Activity visible={fetching} />
      <Text onPress={onPress}>Fetch Stuff...</Text>
    </View>
  );
}
```

NPX ET CREATE EXPO APP

UTILISER DES ANIMATIONS

UTILISER LES ANIMATIONS REACT NATIVE RÉANIMÉ

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/modal-screen$ npx expo install react-native-reanimated
 › Installing 1 SDK 51.0.0 compatible native module using npm
 > npm install

 added 1 package, and audited 1199 packages in 5s

 131 packages are looking for funding
   run `npm fund` for details

 found 0 vulnerabilities
```

modal-screen > babel.config.js > ...

```
1  module.exports = function(api) {
2    api.cache(true);
3    return {
4      presets: ['babel-preset-expo'],
5      plugins: ['react-native-reanimated/plugin'],
6    };
7  };
8
```

NPX ET CREATE EXPO APP

UTILISER DES ANIMATIONS

ANIMATION DES COMPOSANTS DE MISE EN PAGE

modal-screen > styles.ts > styles > container

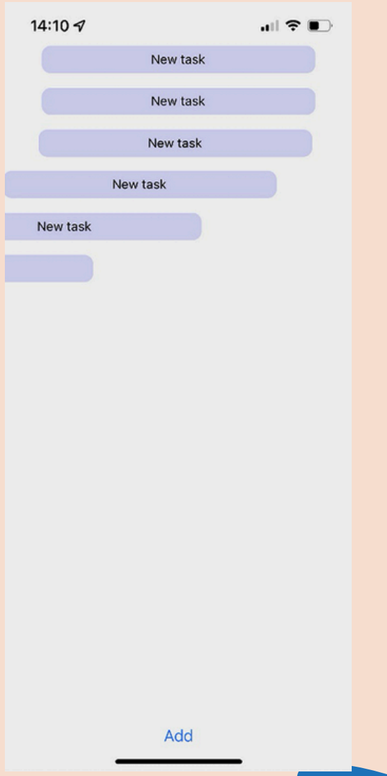
```ts
import { StyleSheet } from "react-native";

export const styles = StyleSheet.create({
  container: {
    paddingTop: 40,
    paddingBottom: 20,
    flexDirection: "column",
    alignItems: "center",
    flex: 1,
    backgroundColor: "#eee",
  },

  todoItem: {
    width: 300,
    height: 30,
    backgroundColor: "#cccccc",
    borderRadius: 10,
    alignItems: "center",
    justifyContent: "center",
    marginVertical: 8,
  },
});
```

modal-screen > TodoItem.tsx > ...

```tsx
import { Text, TouchableOpacity } from "react-native";
import Animated, { SlideInLeft, SlideOutRight } from "react-native-reanimated";
import { styles } from "./styles";

export type Todo = {
  id: string;
  title: string;
};

type TodoItemProps = Todo & {
  onPress: (id: string) => void;
};

export const TodoItem = ({ id, title, onPress }: TodoItemProps) => {
  return (
    <Animated.View entering={SlideInLeft} exiting={SlideOutRight}>
      <TouchableOpacity onPress={() => onPress(id)} style={styles.todoItem}>
        <Text>{title}</Text>
      </TouchableOpacity>
    </Animated.View>
  );
};
```

modal-screen > App.tsx > App > addTask

```tsx
import { useState } from "react";
import { Button, View } from "react-native";
import { TodoItem, Todo } from "./TodoItem";
import { styles } from "./styles";

export default function App() {
  const [todoList, setTodoList] = useState<Todo[]>([]);

  const addTask = () => {
    setTodoList([
      ...todoList,
      { id: String(new Date().getTime()), title: "New task" },
    ]);
  };

  const deleteTask = (id: string) => {
    setTodoList(todoList.filter((todo) => todo.id !== id));
  };

  return (
    <View style={styles.container}>
      <View style={{ flex: 1 }}>
        {todoList.map(({ id, title }) => (
          <TodoItem key={id} id={id} title={title} onPress={deleteTask} />
        ))}
      </View>
      <Button onPress={addTask} title="Add" />
    </View>
  );
}
```



NPX ET CREATE EXPO APP

UTILISER DES ANIMATIONS

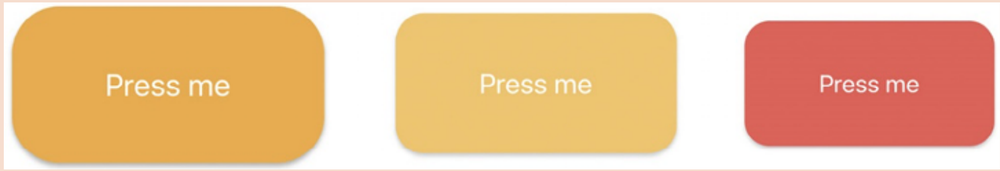
ANIMATION DES STYLES DE COMPOSANTS

modal-screen > styles.ts > styles > buttonContainer

```ts
import { StyleSheet } from "react-native";

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    height: "100%",
  },
  buttonContainer: {
    height: 100,
    width: 200,
    backgroundColor: "orange",
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.3,
    shadowRadius: 1.4,
  },
  button: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
  buttonText: {
    color: "#fff",
    fontSize: 20,
  },
});

export default styles;
```

modal-screen > App.tsx > App

```tsx
const App = () => {
  const animatedStyles = useAnimatedStyle(() => {
  }, []);

  const onPressIn = () => {
    radius.value = withSpring(20);
    opacity.value = withSpring(0.7);
    scale.value = withSpring(0.9);
  };

  const onLongPress = () => {
    scale.value = withSpring(0.8);
    color.value = withSpring(1);
  };

  const onPressOut = () => {
    radius.value = withSpring(30);
    opacity.value = withSpring(1);
    scale.value = withSpring(1, { damping: 50 });
    color.value = withSpring(0);
  };

  return (
    <View style={styles.container}>
      <Animated.View style={[styles.buttonContainer, animatedStyles]}>
        <Pressable
          onPressIn={onPressIn}
          onPressOut={onPressOut}
          onLongPress={onLongPress}
          style={styles.button}
        >
          <Text style={styles.buttonText}>Press me</Text>
        </Pressable>
      </Animated.View>
    </View>
  );
};

export default App;
```



NPX ET CREATE EXPO APP

CONTRÔLE DE L'AFFICHAGE DES IMAGES

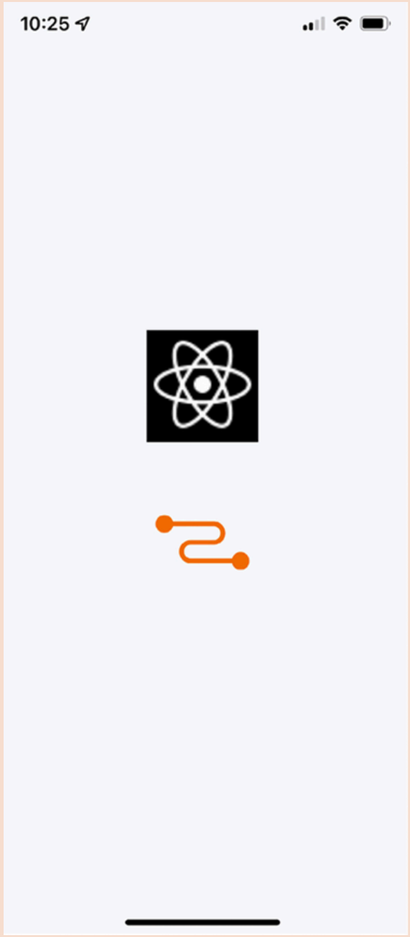
CHARGEMENT D'IMAGES

image-controle > TS styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  image: {
    width: 100,
    height: 100,
    margin: 20,
  },
});
```

image-controle > App.tsx > App

```tsx
import React from "react";
import { View, Image } from "react-native";
import styles from "./styles";

const reactLogo = "https://reactnative.dev/docs/assets/favicon.png";
const relayLogo = require("./assets/relay.png");

export default function App() {
  return (
    <View style={styles.container}>
      <Image style={styles.image} source={{ uri: reactLogo }} />
      <Image style={styles.image} source={relayLogo} />
    </View>
  );
}
```



NPX ET CREATE EXPO APP

CONTRÔLE DE L'AFFICHAGE DES IMAGES

REDIMENSIONNER LES IMAGES

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/image-controle$ npm i @react-native-community/slider

added 1 package, and audited 1194 packages in 5s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/image-controle$
```

App.tsx | styles.ts

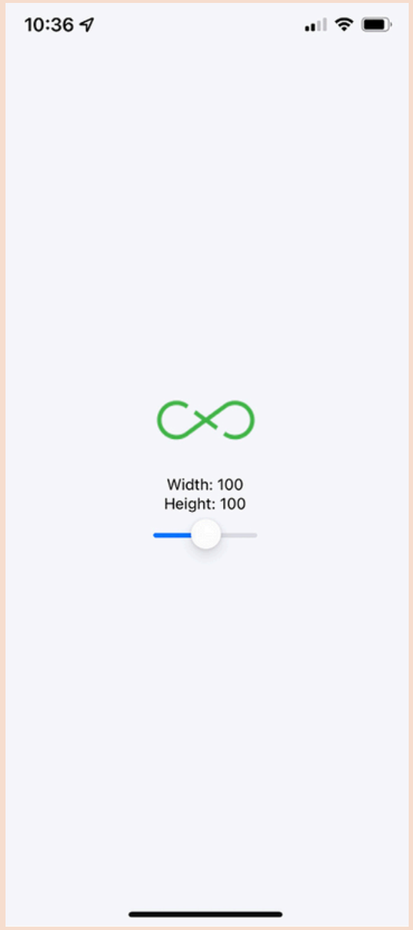
image-controle > styles.ts > default

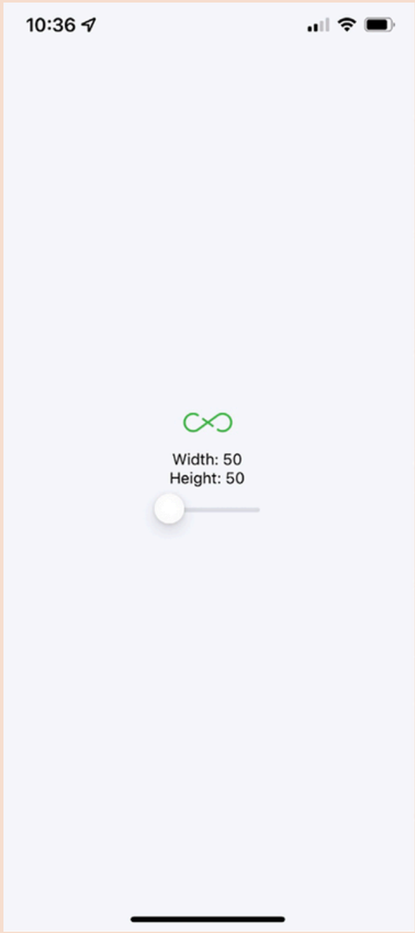
```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  slider: {
    width: 100,
  },
});
```

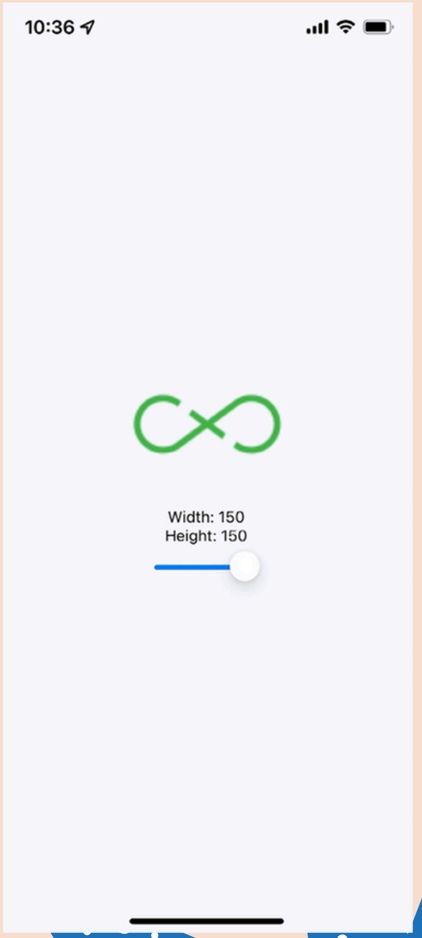
App.tsx | styles.ts

image-controle > App.tsx > App

```tsx
import React, { useState } from "react";
import { View, Text, Image } from "react-native";
import Slider from "@react-native-community/slider";
import styles from "./styles";

export default function App() {
  const source = require("./assets/flux.png");
  const [width, setWidth] = useState(100);
  const [height, setHeight] = useState(100);

  return (
    <View style={styles.container}>
      <Image source={source} style={{ width, height }} />
      <Text>Width: {width}</Text>
      <Text>Height: {height}</Text>
      <Slider
        style={styles.slider}
        minimumValue={50}
        maximumValue={150}
        value={width}
        onValueChange={(value) => {
          setWidth(value);
          setHeight(value);
        }}
      />
    </View>
  );
}
```







NPX ET CREATE EXPO APP

CONTRÔLE DE L'AFFICHAGE DES IMAGES

CHARGEMENT D'IMAGE EN DIFFÉRÉ

App.tsx | Button.tsx | LazyImage.tsx | styl

image-controle > styles.ts > default > button

```tsx
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  button: {
    padding: 10,
    margin: 5,
    backgroundColor: "azure",
    borderWidth: 1,
    borderRadius: 4,
    borderColor: "slategrey",
  },

  buttonText: {
    color: "slategrey",
  },
});
```

App.tsx | Button.tsx | LazyImage.tsx | styles.ts

image-controle > LazyImage.tsx > LazyImage

```tsx
import React, { useState } from "react";
import {
  View,
  Image,
  ImageProps,
  ImageStyle,
  StyleProp,
  ImageSourcePropType,
} from "react-native";

const placeholder = require("./assets/placeholder.png");

type PlaceholderProps = {
  loaded: boolean;
  style: StyleProp<ImageStyle>;
};

function Placeholder({ loaded, style }: PlaceholderProps) {
  if (loaded) {
    return null;
  } else {
    return <Image style={style} source={placeholder} />;
  }
}

type Props = {
  style: StyleProp<ImageStyle>;
  resizeMode: ImageProps["resizeMode"];
  source: ImageSourcePropType | null;
};

export default function LazyImage({ style, resizeMode, source }: Props) {
  const [loaded, setLoaded] = useState(false);

  return (
    <View style={style}>
      {!!source ? (
        <Image
          source={source}
          resizeMode={resizeMode}
          style={style}
          onLoad={() => {
            setLoaded(true);
          }}
        />
      ) : (
        <Placeholder loaded={loaded} style={style} />
      )}
    </View>
  );
}
```

App.tsx | Button.tsx | LazyImage.tsx | styles.ts

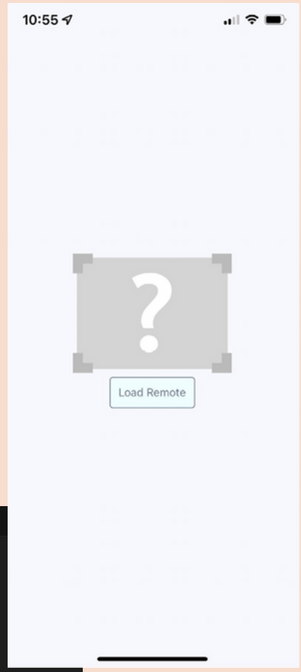
image-controle > Button.tsx > Button

```tsx
import React from "react";
import { Text, TouchableOpacity } from "react-native";
import styles from "./styles";

type Props = {
  label: string;
  onPress: () => void;
};

export default function Button({ label, onPress }: Props) {
  return (
    <TouchableOpacity style={styles.button} onPress={onPress}>
      <Text style={styles.buttonText}>{label}</Text>
    </TouchableOpacity>
  );
}
```

App.tsx | Button.tsx | LazyImage.tsx | styles.ts

image-controle > App.tsx > LazyLoading

```tsx
import React, { useState } from "react";
import { ImageSourcePropType, View } from "react-native";
import styles from "./styles";
import LazyImage from "./LazyImage";
import Button from "./Button";

const remote = "https://reactnative.dev/docs/assets/favicon.png";

export default function LazyLoading() {
  const [source, setSource] = useState<ImageSourcePropType | null>(null);

  return (
    <View style={styles.container}>
      <LazyImage
        style={{ width: 200, height: 150 }}
        resizeMode="contain"
        source={source}
      />
      <Button
        label="Load Remote"
        onPress={() => {
          setSource({ uri: remote });
        }}
      />
    </View>
  );
}
```



NPX ET CREATE EXPO APP

CONTRÔLE DE L'AFFICHAGE DES IMAGES

ICÔNES DE RENDU

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/image-controle$ npm i @react-native-picker/picker

added 1 package, and audited 1195 packages in 5s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/image-controle$ npx expo install @expo/vector-icons
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

up to date, audited 1195 packages in 3s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

App.tsx 1 | icon-names.ts | styles.ts

image-controle > styles.ts > default

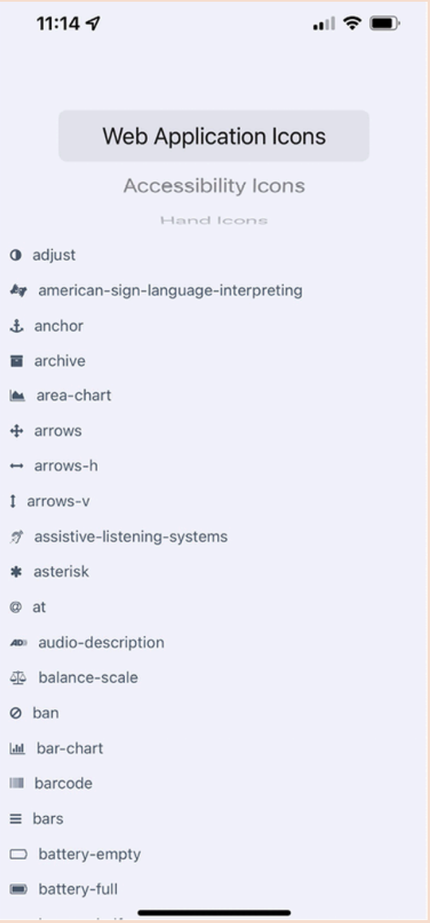
```tsx
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    alignItems: "center",
    backgroundColor: "ghostwhite",
  },

  picker: {
    height: 200,
    width: 300,
    marginTop: 20,
  },

  icons: {
    alignSelf: "stretch",
  },

  item: {
    flex: 1,
    flexDirection: "row",
    alignItems: "center",
  },

  itemIcon: {
    padding: 10,
    color: "slategrey",
  },

  itemText: {
    color: "slategrey",
  },
});
```

App.tsx 1 | icon-names.ts | styles.ts

image-controle > icon-names.ts > iconNames > web_app_icons

```ts
export const iconNames = {
    web_app_icons: [
      "adjust",
      "american-sign-language-interpreting",
      "anchor",
      "archive",
      "area-chart",
      "arrows",
      "arrows-h",
      "arrows-v",
      "assistive-listening-systems",
      "asterisk",
      "at",
      "audio-description",
      "balance-scale",
      "ban",
      "bar-chart",
      "barcode",
      "bars",
      "battery-empty",
      "battery-full",
      "battery-half",
      "battery-quarter",
      "battery-three-quarters",
      "bed",
      "beer",
      "bell",
    ],
    ally: [
      "american-sign-language-interpreting",
      "assistive-listening-systems",
      "audio-description",
      "blind",
      "braille",
      "cc",
```

```ts
    ally: [
      "american-sign-language-interpreting",
      "assistive-listening-systems",
      "audio-description",
      "blind",
      "braille",
      "cc",
      "deaf",
      "low-vision",
      "question-circle-o",
      "sign-language",
      "tty",
      "universal-access",
      "volume-control-phone",
      "wheelchair",
      "wheelchair-alt",
    ],
    hand: [
      "hand-lizard-o",
      "hand-o-down",
      "hand-o-left",
      "hand-o-right",
      "hand-o-up",
      "hand-paper-o",
      "hand-peace-o",
      "hand-pointer-o",
      "hand-rock-o",
      "hand-scissors-o",
      "hand-spock-o",
      "thumbs-down",
      "thumbs-o-down",
      "thumbs-o-up",
      "thumbs-up",
    ],
  } as const;

  export type IconsType = keyof typeof iconNames;
  export type IconName = (typeof iconNames)[IconsType][number];
```



NPX ET CREATE EXPO APP

SE DÉCONNECTER

DÉTECTION DE L'ÉTAT DU RÉSEAU

```
✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd offline
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/offline$ npm i @react-native-community/netinfo

added 1 package, and audited 1194 packages in 6s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

App.tsx M | styles.ts U

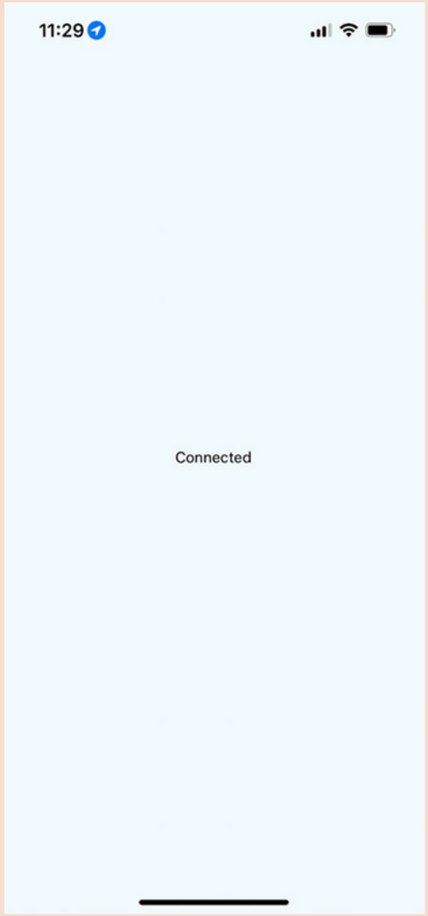
offline > styles.ts > default > welcome

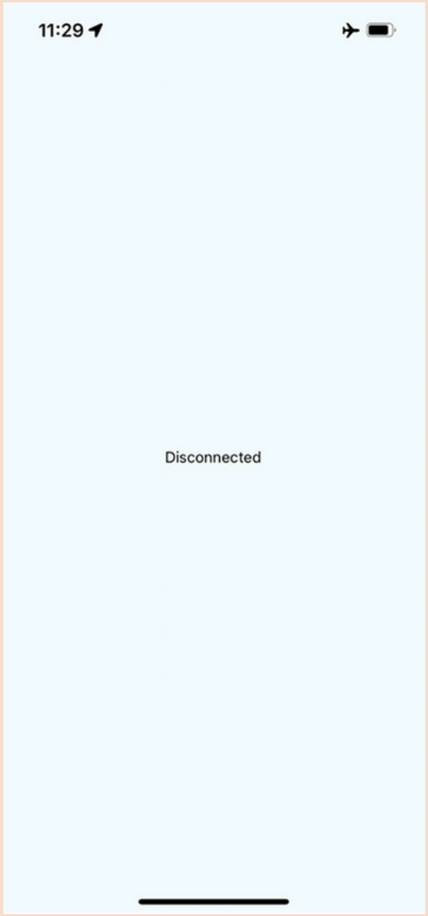
```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "#F5FCFF",
  },
  welcome: {
    fontSize: 20,
    textAlign: "center",
    margin: 10,
  },
  instructions: {
    textAlign: "center",
    color: "#333333",
    marginBottom: 5,
  },
});
```

App.tsx M | styles.ts U

offline > App.tsx > ...

```ts
import React, { useState, useEffect } from "react";
import { Text, View } from "react-native";
import NetInfo, { NetInfoState } from "@react-native-community/netinfo";
import styles from "./styles";

const connectedMap = {
  none: "Disconnected",
  unknown: "Disconnected",
  cellular: "Connected",
  wifi: "Connected",
  bluetooth: "Connected",
  ethernet: "Connected",
  wimax: "Connected",
  vpn: "Connected",
  other: "Connected",
} as const;

export default function App() {
  const [connected, setConnected] = useState("");

  useEffect(() => {
    function onNetworkChange(connection: NetInfoState) {
      const type = connection.type;
      setConnected(connectedMap[type]);
    }

    const unsubscribe = NetInfo.addEventListener(onNetworkChange);

    return () => {
      unsubscribe();
    };
  }, []);

  return (
    <View style={styles.container}>
      <Text>{connected}</Text>
    </View>
  );
}
```





NPX ET CREATE EXPO APP

SE DÉCONNECTER

STOCKAGE DES DONNÉES D'APPLICATION

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/offline$ npx expo install @react-native-async-storage/async-storage
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

added 3 packages, and audited 1197 packages in 5s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

App.tsx M | Button.tsx U | styles.ts U

offline > styles.ts > default > button

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "ghostwhite",
    paddingTop: 50,
  },

  input: {
    height: 35,
    backgroundColor: "white",
    marginBottom: 10,
    width: 100,
  },

  list: {
    height: 450,
  },

  controls: {
    flex: 1,
    flexDirection: "row",
    alignSelf: "stretch",
    justifyContent: "space-between",
    padding: 20,
  },

  button: {
    height: 45,
    width: 100,
    padding: 10,
    margin: 1,
    backgroundColor: "azure",
    borderWidth: 1,
    borderRadius: 4,
    borderColor: "slategrey",
    justifyContent: "center",
    alignItems: "center",
  },

  buttonText: {
    color: "slategrey",
  },
});
```

App.tsx M | Button.tsx U | styles.ts U

offline > App.tsx > App

```tsx
import React, { useState, useEffect } from "react";
import { Text, TextInput, View, FlatList } from "react-native";
import AsyncStorage from "@react-native-async-storage/async-storage";
import styles from "./styles";
import { Button } from "./Button";
import { KeyValuePair } from "@react-native-async-storage/async-storage/lib/typescript/types";

export default function App() {
  const [key, setKey] = useState("");
  const [value, setValue] = useState("");
  const [source, setSource] = useState<KeyValuePair[]>([]);

  function setItem() {
    return AsyncStorage.setItem(key, value)
      .then(() => {
        setKey("");
        setValue("");
      })
      .then(loadItems);
  }

  function clearItems() {
    return AsyncStorage.clear().then(loadItems);
  }

  async function loadItems() {
    const keys = await AsyncStorage.getAllKeys();
    const values = await AsyncStorage.multiGet(keys);
    setSource([...values]);
  }

  useEffect(() => {
    loadItems();
  }, []);

  return (
    <View style={styles.container}>
      <Text>Key:</Text>
      <TextInput
        style={styles.input}
        value={key}
        onChangeText={(v) => {
          setKey(v);
        }}
      />
      <Text>Value:</Text>
      <TextInput
        style={styles.input}
        value={value}
```

App.tsx M | Button.tsx U | styles.ts U

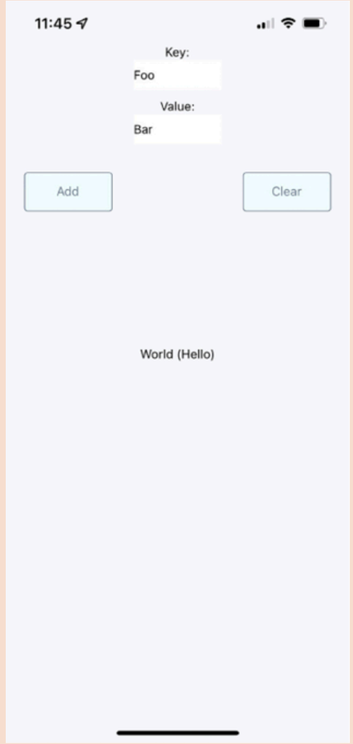
offline > App.tsx > App

```tsx
export default function App() {
        }}
      />
      <Text>Value:</Text>
      <TextInput
        style={styles.input}
        value={value}
        onChangeText={(v) => {
          setValue(v);
        }}
      />
      <View style={styles.controls}>
        <Button label="Add" onPress={setItem} />
        <Button label="Clear" onPress={clearItems} />
      </View>
      <View style={styles.list}>
        <FlatList
          data={source.map(([key, value]) => ({
            key: key.toString(),
            value,
          }))}
          renderItem={({ item: { value, key } }) => (
            <Text>
              {value} ({key})
            </Text>
          )}
        />
      </View>
    </View>
  );
}
```

App.tsx M | Button.tsx U | styles.ts U

offline > Button.tsx > Button

```tsx
import React from "react";
import { Text, TouchableOpacity } from "react-native";
import styles from "./styles";

type ButtonProps = {
  label: string;
  onPress: () => void;
};

export const Button = ({ label, onPress }: ButtonProps) => {
  return (
    <TouchableOpacity
      style={styles.button}
      onPress={onPress}
      activeOpacity={0.5}
    >
      <Text style={styles.buttonText}>{label}</Text>
    </TouchableOpacity>
  );
};
```



NPX ET CREATE EXPO APP

SE DÉCONNECTER

SYNCHRONISATION DES DONNÉES D`APPLICATION

App.tsx M | store.ts U | styles.ts U

offline > styles.ts > default

```ts
import { StyleSheet } from "react-native";

export default StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
    backgroundColor: "#F5FCFF",
  },
  welcome: {
    fontSize: 20,
    textAlign: "center",
    margin: 10,
  },
  instructions: {
    textAlign: "center",
    color: "#333333",
    marginBottom: 5,
  },
});
```

App.tsx M | store.ts U | styles.ts U

offline > store.ts > get > <function>

```ts
import AsyncStorage from "@react-native-async-storage/async-storage";
import NetInfo from "@react-native-community/netinfo";

const fakeNetworkData = {
  first: false,
  second: false,
  third: false,
};

export type Key = keyof typeof fakeNetworkData;

let connected = false;
const unsynced: string[] = [];

export function set(key: Key, value: boolean) {
  return new Promise((resolve, reject) => {
    if (connected) {
      fakeNetworkData[key] = value;
      resolve(true);
    } else {
      AsyncStorage.setItem(key, value.toString()).then(
        () => {
          unsynced.push(key);
          resolve(false);
        },
        (err) => reject(err)
      );
    }
  });
}

export function get(key?: Key): Promise<boolean | typeof fakeNetworkData> {
  return new Promise((resolve, reject) => {
    if (connected) {
      resolve(key ? fakeNetworkData[key] : fakeNetworkData);
    } else if (key) {
      AsyncStorage.getItem(key)
        .then((item) => resolve(item === "true"))
        .catch((err) => reject(err));
    } else {
```

App.tsx M | store.ts U | styles.ts U

offline > store.ts > get > <function>

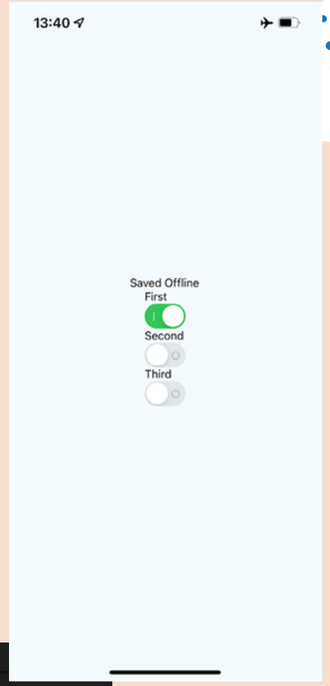
```ts
export function get(key?: Key): Promise<boolean | typeof fakeNetworkData> {
  return new Promise((resolve, reject) => {
        .then((item) => resolve(item === "true"))
        .catch((err) => reject(err));
    } else {
      AsyncStorage.getAllKeys()
        .then((keys) =>
          AsyncStorage.multiGet(keys).then((items) =>
            resolve(Object.fromEntries(items) as any)
          )
        )
        .catch((err) => reject(err));
    }
  });
}

NetInfo.fetch().then(
  (connection) => {
    connected = ["wifi", "unknown"].includes(connection.type);
  },
  () => {
    connected = false;
  }
);

NetInfo.addEventListener((connection) => {
  connected = ["wifi", "unknown"].includes(connection.type);

  if (connected && unsynced.length) {
    AsyncStorage.multiGet(unsynced).then((items) => {
      items.forEach(([key, val]) => set(key as Key, val === "true"));
      unsynced.length = 0;
    });
  }
```

App.tsx M | store.ts U | styles.ts U

offline > App.tsx > App

```tsx
import React, { useState, useEffect } from "react";
import { Text, View, Switch } from "react-native";
import NetInfo from "@react-native-community/netinfo";
import styles from "./styles";
import { set, get, Key } from "./store";

export default function App() {
  const [message, setMessage] = useState<string | null>(null);
  const [first, setFirst] = useState(false);
  const [second, setSecond] = useState(false);
  const [third, setThird] = useState(false);
  const setters = new Map([
    ["first", setFirst],
    ["second", setSecond],
    ["third", setThird],
  ]);

  function save(key: Key) {
    return (value: boolean) => {
      set(key, value).then(
        (connected) => {
          setters.get(key)?.(value);
          setMessage(connected ? null : "Saved Offline");
        },
        (err) => {
          setMessage(err);
        }
      );
    };
  }

  useEffect(() => {
    NetInfo.fetch().then(() =>
```

```tsx
export default function App() {

  useEffect(() => {
    NetInfo.fetch().then(() =>
      get().then(
        (items) => {
          for (let [key, value] of Object.entries(items)) {
            setters.get(key)?.(value);
          }
        },
        (err) => {
          setMessage(err);
        }
      )
    );
  }, []);

  return (
    <View style={styles.container}>
      <Text>{message}</Text>
      <View>
        <Text>First</Text>
        <Switch value={first} onValueChange={save("first")} />
      </View>
      <View>
        <Text>Second</Text>
        <Switch value={second} onValueChange={save("second")} />
      </View>
      <View>
        <Text>Third</Text>
        <Switch value={third} onValueChange={save("third")} />
      </View>
    </View>
  );
}
```



LA GESTION DES ETATS - REACT NATIVE

STATE "ETATS" ET PROPS "PROPRIÉTÉS"

QU'EST-CE QUE L'ÉTAT ET EN QUOI EST-IL DIFFÉRENT DES PROPS ?

```
import React from "react";
import { View, Text, Pressable } from "react-native";
export const ManagedButton = () => {
  return (
    <View>
      <Text>this text will display the current status</Text>
      <Pressable onPress="">
        <Text>Press here to check/uncheck</Text>
      </Pressable>
    </View>
  );
};
```

```
import React, { useState } from "react";
import { View, Text, Pressable } from "react-native";
export const ManagedButton = () => {
    const [checkedState, setCHeckedState] =      useState("unchecked");
  return (
    <View>
        <Text>this text will display the current status,          which is: {checkedState}</Text>
      <Pressable onPress="">
        <Text>Press here to check/uncheck</Text>
      </Pressable>
    </View>
  );
};
```

```
<Pressable onPress={setCheckedState("checked")}>
```

LA GESTION DES ETATS - REACT NATIVE

STATE "ETATS" ET PROPS "PROPRIÉTÉS"

QU'EST-CE QUE L'ÉTAT ET EN QUOI EST-IL DIFFÉRENT DES PROPS ?

```
import React, { useState } from "react";
import { View, Text, Pressable } from "react-native";
const ManagedText = ({checkedState}) => {
  return (
     <Text>this text will display the current status,        which is: {checkedState}</Text>
  );
};
export const ParentComponent = () => {
     const [checkedState, setCheckedState] =        useState("unchecked");
    return (
       <View>
          <ManagedText checkedState={checkedState} />
            <Pressable onPress={() =>              setCheckedState("checked")}>
            <Text>Press here to check/uncheck</Text>
          </Pressable>
     </View>
  );
};
```

```
const ManagedText = (fancyComponentStuff) => {
  return (
     <Text>this text will display the current status,        which is:{fancyComponentStuff}</Text>
  );
};
export const ParentComponent = () => {
     const [checkedState, setCheckedState] =        useState("unchecked");
    return (
       <View>
          <ManagedText fancyComponentStuff={checkedState} />
          <Pressable onPress={setCheckedState("checked")}>
            <Text>Press here to check/uncheck</Text>
          </Pressable>
     </View>
  );
};
```

La chose la plus importante à retenir à propos des propos est la suivante : ***les props sont immuables (ou en lecture seule).***

LA GESTION DES ETATS - REACT NATIVE

LES HOOKS

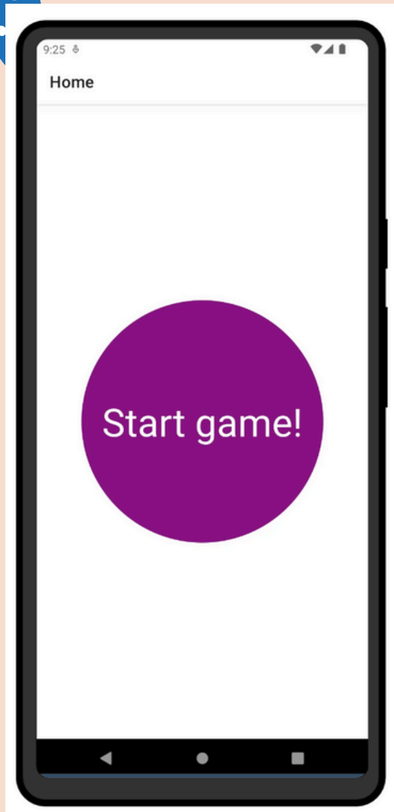
USESTATE ET USEEFFECT

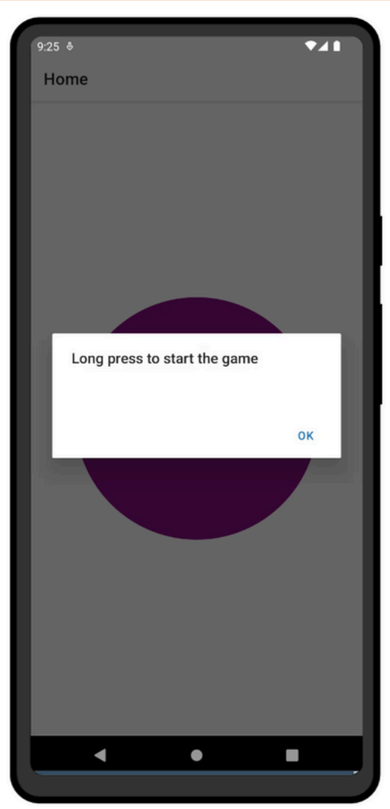
```
const LikesParentComponent = () => {
   const getCounterNumberFromApi =      someFunctionRetrievingDataFromAPI();
   const [counterNumber, setCounterNumber] =      useState(getCounterNumberFromApi)
 return (
   <LikesComponent counterNumber={counterNumber} />
 );
};
const LikesComponent = (counterNumber) => {
     const [likeState, setLikedState] = useState        ("haven't yet liked");
   return (
      <View>
         <Text>you {likeState} this post</Text>
         <Pressable onPress={setLikedState("liked")}>
           <Text>Press here to check/uncheck</Text>
         </Pressable>
           <Text>{counterNumber} other people liked this                post</Text>
    </View>
  );
 };
```

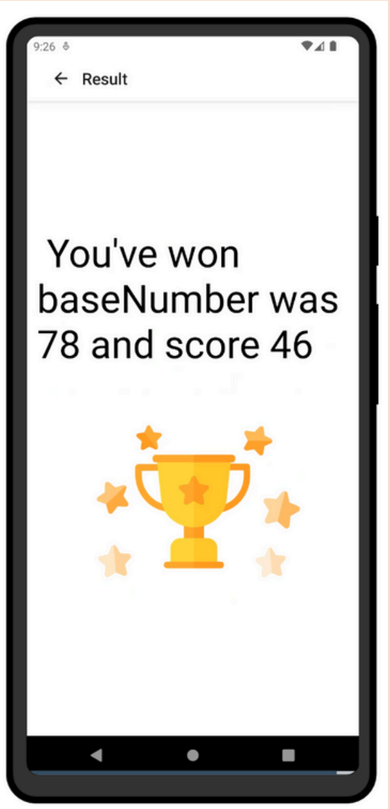
```
const LikesComponent = (counterNumber, setCountNumber) => {
     const [likeState, setLikedState] = useState        ("haven't yet liked");
   useEffect(() => {
      if (likeState === "liked") {
         setCounterNumber(counterNumber++)
      }
      else {
         setCounterNumber(counterNumber-1)
      }
   }, [likeState])
   return (
      <View>
         <Text>you {likeState} this post</Text>
         <Pressable onPress={setLikedState("liked")}>
           <Text>Press here to check/uncheck</Text>
         </Pressable>
           <Text>{counterNumber} other people liked this                post</Text>
    </View>
  );
 };
```

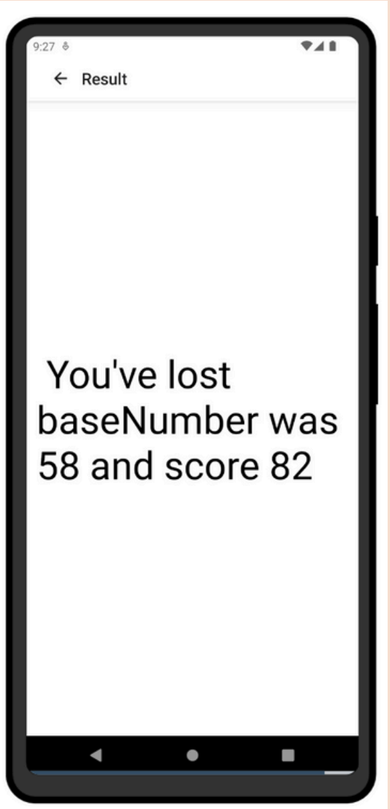
CAS D'APPLICATION - JEUX

PRÉSENTATION DU PROJET



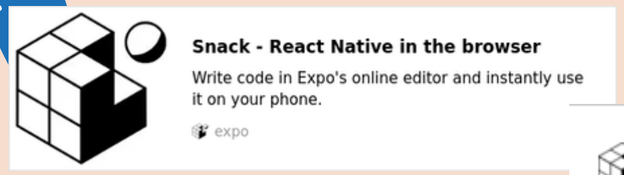


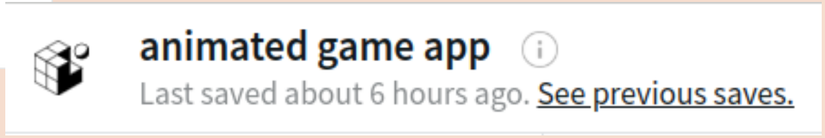




CAS D'APPLICATION - JEUX

MISE EN PLACE DU PROJET





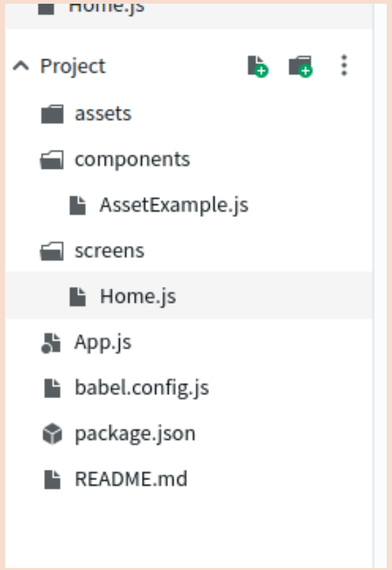

```
module.exports = function(api) {
  api.cache(true);
  return {
    presets: ['babel-preset-expo'],
  };
};
```

```
{
  "dependencies": {
    "react-native-paper": "4.9.2",
    "@expo/vector-icons": "^14.0.2",
    "@react-navigation/elements": "1.3.3",
    "@react-navigation/native": "6.0.10",
    "@react-navigation/native-stack": "6.6.1",
    "expo-status-bar": "~1.12.1",
    "lottie-react-native": "6.7.0",
    "react-native-gesture-handler": "~2.16.1",
    "react-native-safe-area-context": "4.10.5",
    "react-native-screens": "3.31.1",
    "@dotlottie/react-player": "1.6.1",
    "@lottiefiles/react-lottie-player": "3.5.3"
  }
}
```

```
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet, Text, View, SafeAreaView } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
const Stack = createNativeStackNavigator();


export default function App() {
  return (
    <SafeAreaView>
      <Text>
        Change code in the editor and watch it change on your phone! Save to get a shareable url.
      </Text>
    </SafeAreaView>
  );
}
```

CAS D'APPLICATION - JEUX

ÉCRAN HOME

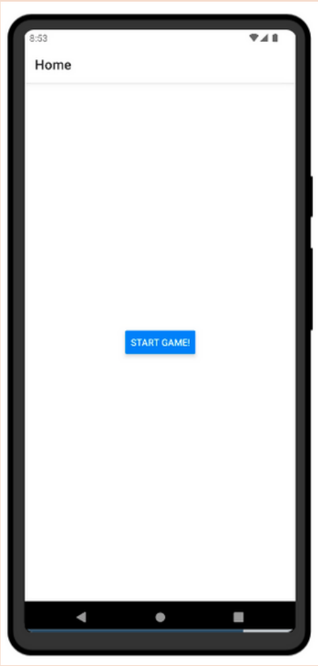


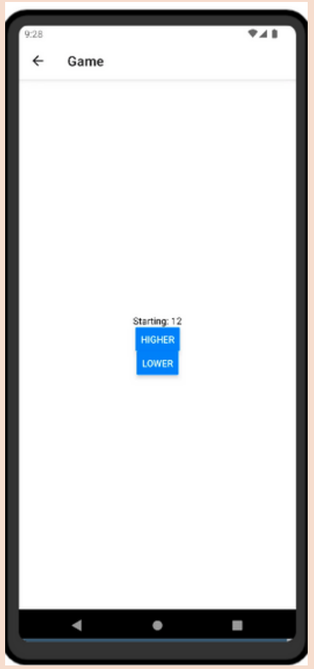

```jsx
import React from 'react';
import { StyleSheet, Text, View, StatusBar } from 'react-native';
export default function Home() {
  return (
    <View style={styles.container}>
     <StatusBar barStyle="dark-content" />
      <Text> home screen</Text>
    </View>
  );
}
// export default function Home() {
//   return (
//     <View style={styles.container}>
//       <StatusBar barStyle="dark-content" />
//       <Text> home screen</Text>
//     </View>
//   );
// }
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

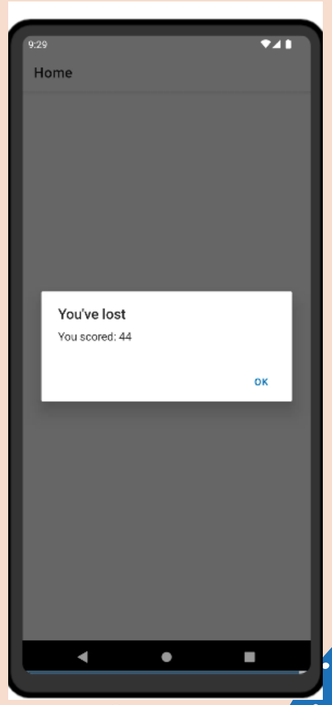
CAS D'APPLICATION - JEUX

```jsx
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
const Stack = createNativeStackNavigator();
import Home from './screens/Home';
import Game from './screens/Game';


export default function App() {
  return (
    <NavigationContainer>
      <StatusBar style='auto' />
      <Stack.Navigator initialRouteName='Home'>
        <Stack.Screen name='Home' component={Home} />
        <Stack.Screen name='Game' component={Game} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

```jsx
import React from 'react';
import { StyleSheet, Button, View, StatusBar } from 'react-native';
import { useNavigation } from '@react-navigation/native';

export default function Home() {
  const navigation = useNavigation();

  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Button onPress={() => navigation.navigate(
        'Game')} title='Start game!' />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```








```jsx
import React, { useEffect, useState } from 'react';
import { StyleSheet, Text, View, StatusBar, Button, Alert } from 'react-native';
import { useNavigation } from '@react-navigation/native';
export default function Game() {

  const [choice, setChoice] = useState('');
  const baseNumber = Math.floor(Math.random() * 100);
  const score = Math.floor(Math.random() * 100);

 const navigation = useNavigation();
 useEffect(() => {

   if (choice) {
  console.log(` base Number = ${baseNumber} et score = ${score}`)
     const winner =
       (choice === 'higher' && score > baseNumber) ||
       (choice === 'lower' && baseNumber > score);
     Alert.alert(`You've ${winner ? 'won' : 'lost'}`,
       `You scored: ${score}`);
     navigation.goBack();
   }
 }, [baseNumber, score, choice])

  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text>Starting: {baseNumber}</Text>
      <Button onPress={() => setChoice('higher')} title='Higher' />
      <Button onPress={() => setChoice('lower')} title='Lower' />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
```

CAS D'APPLICATION - JEUX

STYLE HOME

```jsx
import React from 'react';
import { StyleSheet, Button, View, StatusBar,TouchableHighlight, Text } from 'react-native';
import { useNavigation } from '@react-navigation/native';

export default function Home() {
  const navigation = useNavigation();

  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
        <TouchableHighlight onPress={() => navigation.navigate('Game')} style={styles.button} >
            <Text style={styles.buttonText}> Start game!</Text>
        </TouchableHighlight>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  button: {
    width: 300,
    height: 300,
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    borderRadius: 150,
    backgroundColor: 'purple',
  },
  buttonText: {
    color: 'white',
    fontSize: 48,
  },
});
```

CAS D'APPLICATION - JEUX

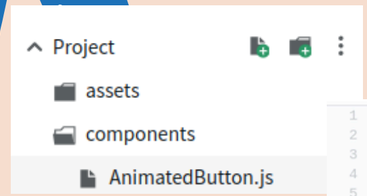
STYLE GAME

```jsx
import React, { useEffect, useState } from 'react';
import { StyleSheet, Text, View, StatusBar, TouchableHighlight, Alert } from 'react-native';
import { useNavigation } from '@react-navigation/native';
export default function Game() {

  const [choice, setChoice] = useState('');

  const baseNumber = Math.floor(Math.random() * 100);
  const score = Math.floor(Math.random() * 100);

  console.log(`beginning base Number = ${baseNumber} et score = ${score}`)

  const navigation = useNavigation();
  useEffect(() => {

    if (choice) {
      console.log(`in useeffect base Number = ${baseNumber} et score = ${score}`)
      const winner =
        (choice === 'higher' && score > baseNumber) ||
        (choice === 'lower' && baseNumber > score);
      Alert.alert(`You've ${winner ? 'won' : 'lost'}`,
        `You scored: ${score}`);

      navigation.goBack();
    }
  }, [baseNumber, score, choice])

  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text style={styles.baseNumber}>
        Starting: {baseNumber}</Text>
      <TouchableHighlight onPress={() =>
        setChoice('higher')} style={[styles.button, styles.buttonGreen]}>
        <Text style={styles.buttonText}>Higher
        </Text>
      </TouchableHighlight>
      <TouchableHighlight onPress={() =>
        setChoice('lower')} style={[styles.button, styles.buttonRed]}>
        <Text style={styles.buttonText} >Lower</Text>
      </TouchableHighlight>
```

```jsx
        </Text>
      </TouchableHighlight>
      <TouchableHighlight onPress={() =>
        setChoice('lower')} style={[styles.button, styles.buttonRed]}>
        <Text style={styles.buttonText} >Lower</Text>
      </TouchableHighlight>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center'
  },
  baseNumber: {
    fontSize: 48,
    marginBottom: 30,
  },
  button: {
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    borderRadius: 15,
    padding: 30,
    marginVertical: 15,
  },
    buttonRed: {
    backgroundColor: 'red',
   },
   buttonGreen: {
    backgroundColor: 'green',
   },
  buttonText: {
    color: 'white',
    fontSize: 24,
  },
});
```

CAS D'APPLICATION - JEUX

LES ANIMATIONS




```jsx
import React from 'react';
import { StyleSheet, Text, TouchableHighlight }
  from 'react-native';
export default function AnimatedButton({ action,
  onPress }) {
        return (
    <TouchableHighlight
      onPress={onPress}
      underlayColor="lightblue"
      style={[
        styles.button,
        action === 'higher' ? styles.buttonGreen :
          styles.buttonRed,
      ]}
    >
      <Text style={styles.buttonText}>{action}</Text>
    </TouchableHighlight>
  );
}

const styles = StyleSheet.create({
  button: {
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    borderRadius: 15,
    padding: 30,
    marginVertical: 15,
  },
  buttonRed: {
    backgroundColor: 'red',
  },
  buttonGreen: {
    backgroundColor: 'green',
  },
  buttonText: {
    color: 'white',
    fontSize: 24,
    textTransform: 'capitalize',
  },
});
```

See previous saves.

```jsx
import React, { useEffect, useState } from 'react';
import { StyleSheet, Text, View, StatusBar, Alert } from 'react-native';
import { useNavigation } from '@react-navigation/native';
import AnimatedButton from '../components/AnimatedButton';

export default function Game() {

  const [choice, setChoice] = useState('');

  const baseNumber = Math.floor(Math.random() * 100);
  const score = Math.floor(Math.random() * 100);

  console.log(`beginning base Number = ${baseNumber} et score = ${score}`)

  const navigation = useNavigation();
  useEffect(() => {

    if (choice) {
      console.log(`in useeffect base Number = ${baseNumber} et score = ${score}`)
      const winner =
        (choice === 'higher' && score > baseNumber) ||
        (choice === 'lower' && baseNumber > score);
      Alert.alert(`You've ${winner ? 'won' : 'lost'}`,
        `You scored: ${score}`);

      navigation.goBack();
    }
  }, [baseNumber, score, choice])

  return (
    <View style={styles.container}>
      <StatusBar barStyle="dark-content" />
      <Text style={styles.baseNumber}> Starting: {baseNumber}</Text>
      <AnimatedButton action='higher' onPress={() => setChoice('higher')} />
      <AnimatedButton action='lower' onPress={() => setChoice('lower')} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center'
  },
  baseNumber: {
    fontSize: 48,
    marginBottom: 30,
  }
});
```

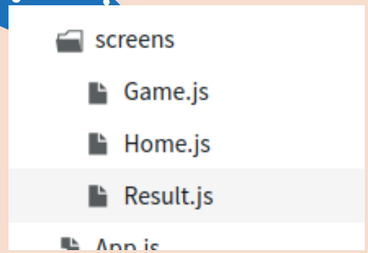
CAS D'APPLICATION - JEUX

```jsx
import React, {useRef} from 'react';
import { StyleSheet, Text, TouchableWithoutFeedback, Animated, View } from 'react-native';

export default function AnimatedButton({ action, onPress }) {
  return (
    <TouchableWithoutFeedback onPress={onPress}>
      <View style={[ styles.button, action === 'higher'
          ? styles.buttonGreen : styles.buttonRed ]}>
          <Text style={styles.buttonText}>{action}</Text>
      </View>
    </TouchableWithoutFeedback>
  );
}

const styles = StyleSheet.create({
  button: {
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    borderRadius: 15,
    padding: 30,
    marginVertical: 15,
  },
  buttonRed: {
    backgroundColor: 'red',
  },
  buttonGreen: {
    backgroundColor: 'green',
  },
  buttonText: {
    color: 'white',
    fontSize: 24,
    textTransform: 'capitalize',
  },
});

```

See previous saves. ✓

```jsx
import React, {useRef} from 'react';
import { StyleSheet, Text, TouchableWithoutFeedback, Animated, View } from 'react-native';

export default function AnimatedButton({ action, onPress }) {

  const opacity = useRef(new Animated.Value(1));

  return (
    <TouchableWithoutFeedback
      onPress={() => {
        Animated.timing(opacity.current, {
          toValue: 0.2,
          duration: 800,
          useNativeDriver: true,
        }).start(() => onPress());
      }}
    >
      <Animated.View
        style={[
          styles.button,
          action === 'higher' ? styles.buttonGreen :
            styles.buttonRed,
          { opacity: opacity.current },
        ]}
       >
        <Text style={styles.buttonText}>{action}
        </Text>
      </Animated.View>
    </TouchableWithoutFeedback>
  );
}

const styles = StyleSheet.create({
  button: {
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    borderRadius: 15,
    padding: 30,
    marginVertical: 15,
  },
  buttonRed: {
    backgroundColor: 'red',
  },
  buttonGreen: {
    backgroundColor: 'green',
  },
  buttonText: {
    color: 'white',
    fontSize: 24,
    textTransform: 'capitalize',
  },
});

```

CAS D'APPLICATION - JEUX



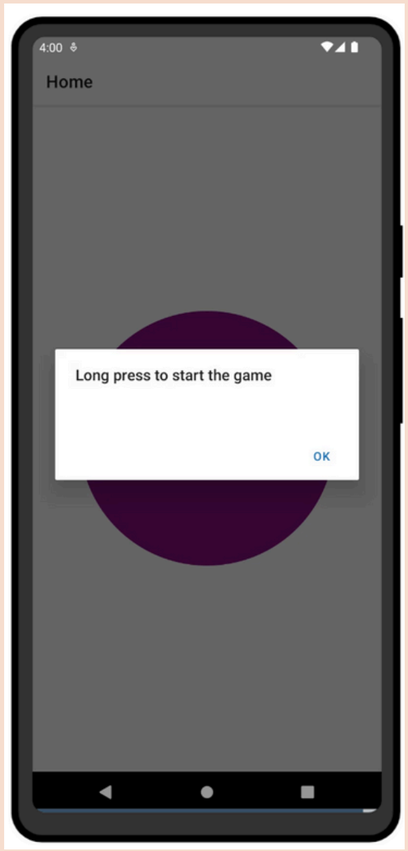
us saves. ✓

```jsx
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';
export default function Result() {
  return (
    <View style={styles.container}>
      <Text>Result screen </Text>
    </View>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

```jsx
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet } from 'react-native';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import { HeaderBackButton } from '@react-navigation/elements';
const Stack = createNativeStackNavigator();
import Home from './screens/Home';
import Game from './screens/Game';
import Result from './screens/Result';


export default function App() {
  return (
    <NavigationContainer>
      <StatusBar style='auto' />
      <Stack.Navigator initialRouteName='Home'>
        <Stack.Screen name='Home' component={Home} />
        <Stack.Screen name='Game' component={Game} />
        <Stack.Screen
          name='Result'
          component={Result}
          options={({ navigation }) => ({
            headerLeft: (props) => (
              <HeaderBackButton
                {...props}
                label='Home'
                onPress={() =>
                  navigation.navigate('Home')}
              />
            ),
          })}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

CAS D'APPLICATION - JEUX

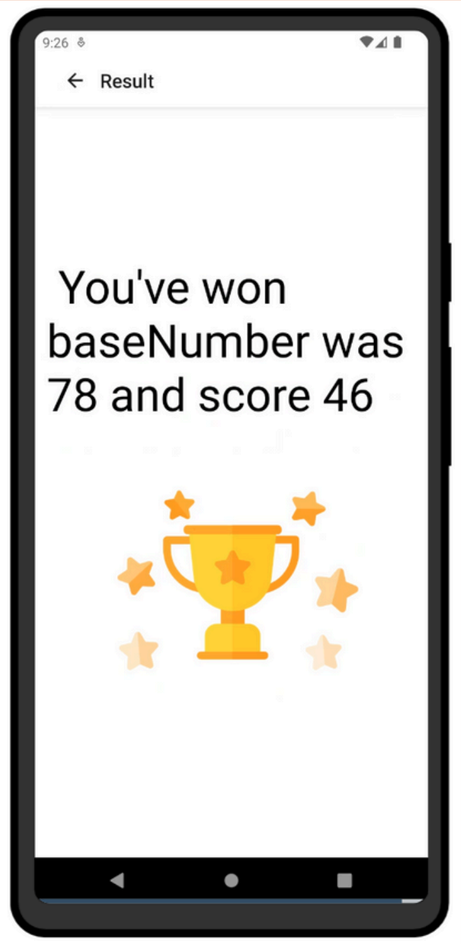
```jsx
import React from 'react';
import { StyleSheet, Text, View, Alert } from 'react-native';
import { useNavigation } from '@react-navigation/native';
import {
  GestureHandlerRootView,
  LongPressGestureHandler,
  TapGestureHandler,
  State,
} from 'react-native-gesture-handler';

export default function Home() {
  const navigation = useNavigation();

  function onLongPress(e) {
    if (e.nativeEvent.state === State.ACTIVE) {
      navigation.navigate('Game');
    }
  }

  function onTap(e) {
    if (e.nativeEvent.state === State.ACTIVE) {
      Alert.alert('Long press to start the game');
    }
  }

  return (
    <GestureHandlerRootView style={{ flex: 1 }}>
    <View style={styles.container}>
      <TapGestureHandler onHandlerStateChange={onTap}>
        <LongPressGestureHandler
          onHandlerStateChange={onLongPress}
          minDurationMs={600}
        >
          <View style={styles.button}>
            <Text style={styles.buttonText}>Start game!</Text>
          </View>
        </LongPressGestureHandler>
      </TapGestureHandler>
    </View>
    </GestureHandlerRootView>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  button: {
    width: 300,
    height: 300,
    display: 'flex',
    alignItems: 'center',
    justifyContent: 'space-around',
    borderRadius: 150,
    backgroundColor: 'purple',
```

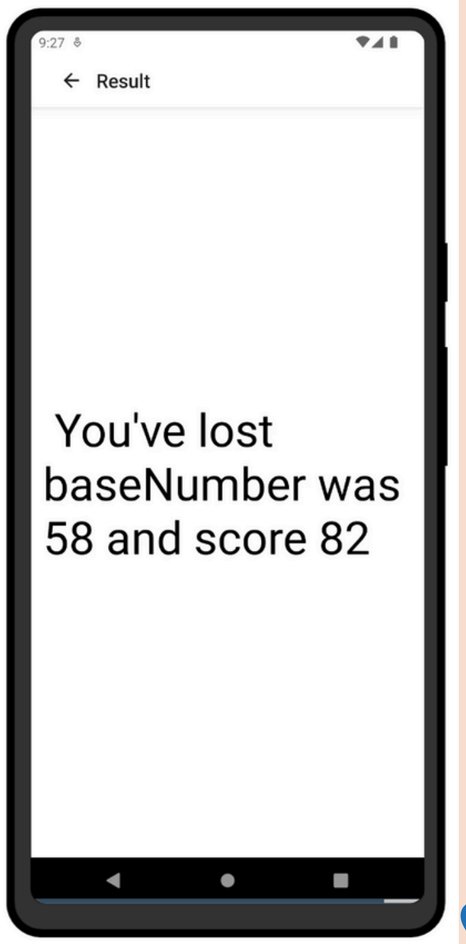


CAS D'APPLICATION - JEUX

```jsx
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';
import LottieView from 'lottie-react-native';
import { useRoute } from '@react-navigation/native';
export default function Result() {
  const route = useRoute();
  const { winner, baseNumber, score } = route.params;
  return (
    <View style={styles.container}>
      <Text style={styles.message}> You've {winner ? `won baseNumber was ${baseNumber} and score ${score}`  : `lost baseNumber was ${baseNumber} and score ${score}`} </Text>
      {winner && (
        <LottieView
          autoPlay
          style={{
            width: 300,
            height: 300,
          }}
          source={require('../assets/winner.json')}
        />
      )}
    </View>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  message: {
    fontSize: 48,
  }
});
```

CAS D'APPLICATION - JEUX



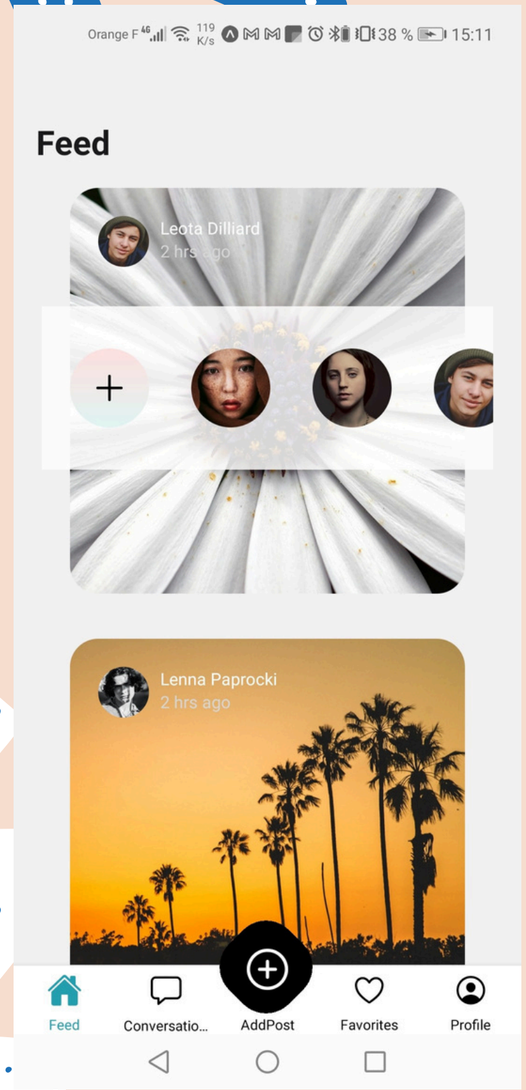


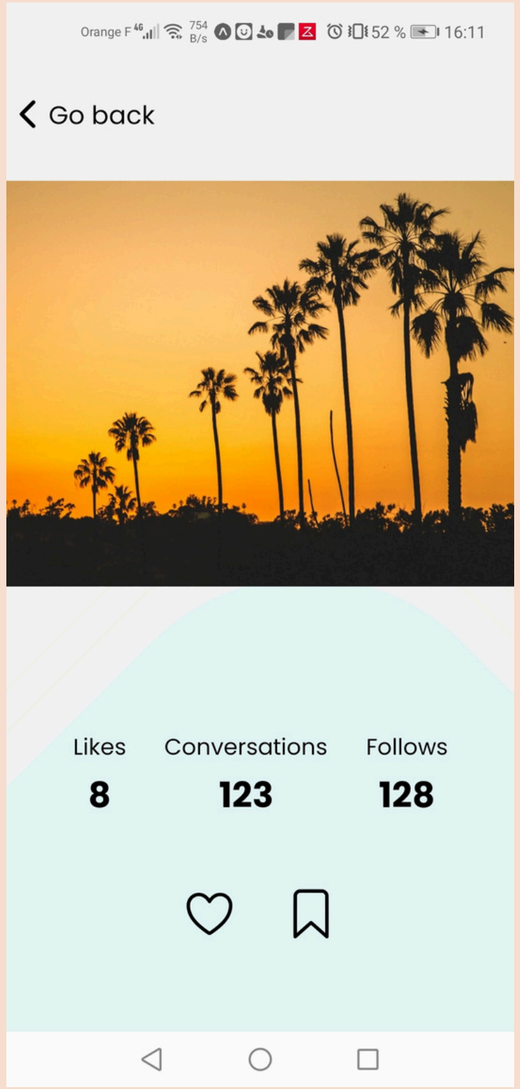
CAS D'APPLICATION - JEUX

LA GESTION DES ETATS - REACT NATIVE

LE PROJET EN FIL ROUGE

PRÉSENTATION (FEED) ET CRÉATION

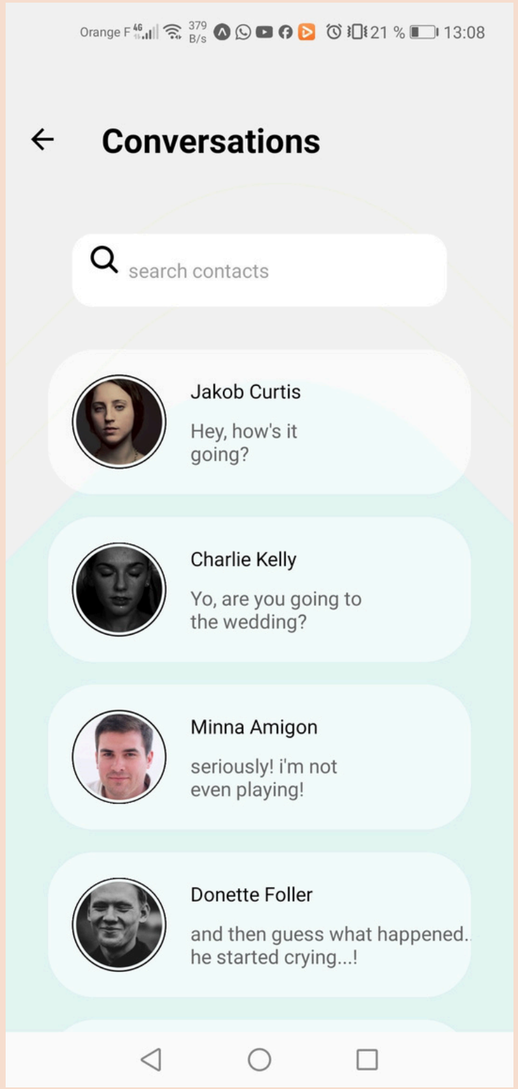


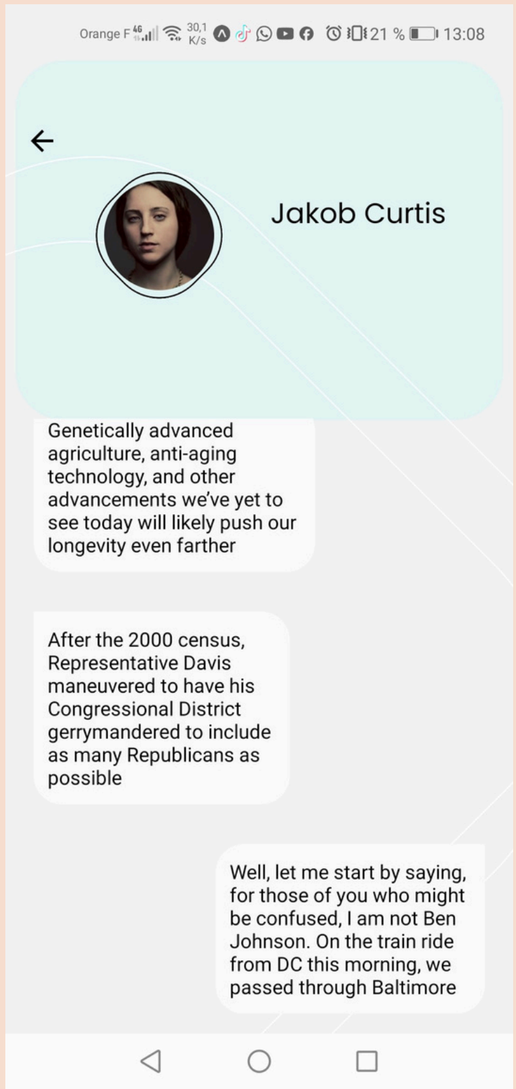


LA GESTION DES ETATS - REACT NATIVE

LE PROJET EN FIL ROUGE

PRESENTATION (CONVERSATION) ET CRÉATION

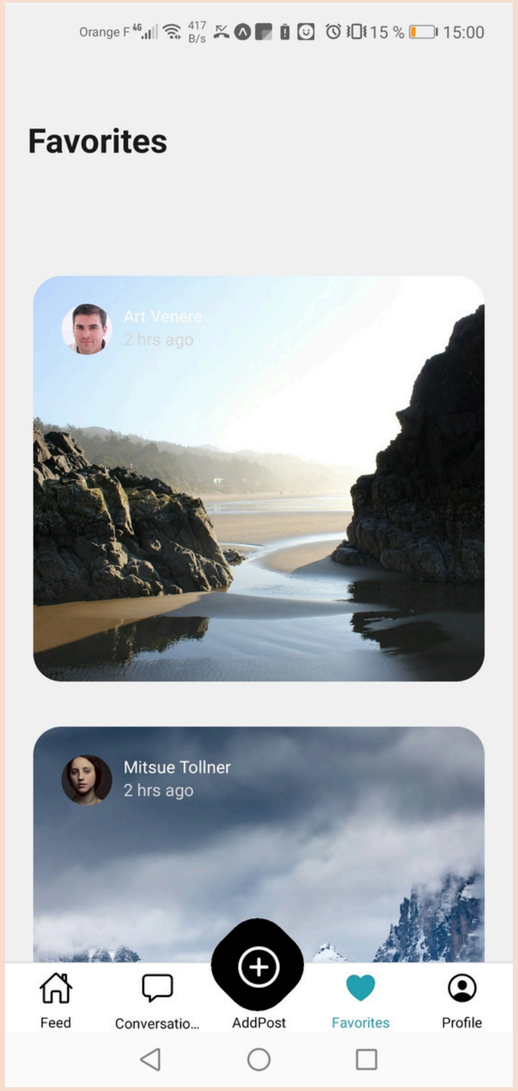




LA GESTION DES ETATS - REACT NATIVE

LE PROJET EN FIL ROUGE

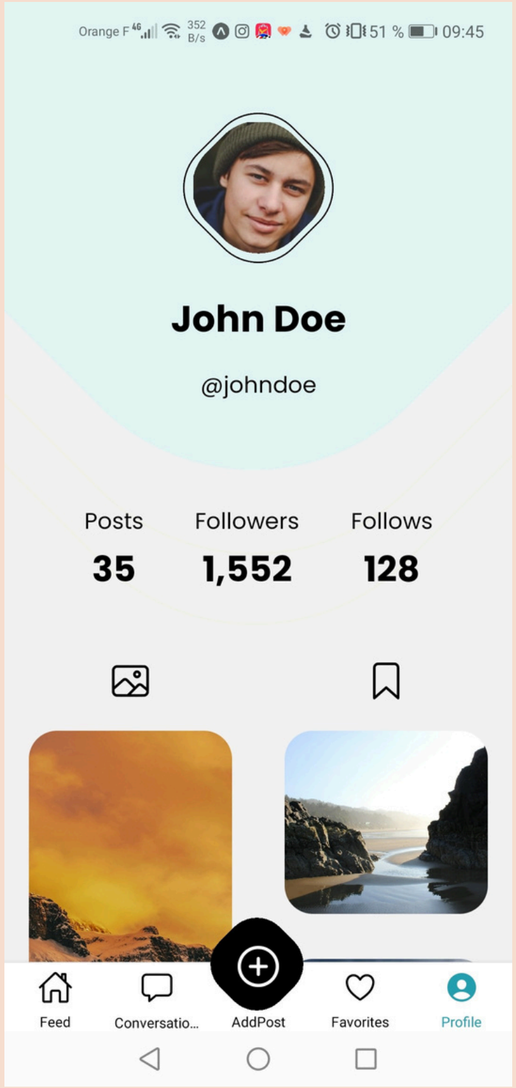
PRESENTATION (FAVORITE) ET CRÉATION



LA GESTION DES ETATS - REACT NATIVE

LE PROJET EN FIL ROUGE

PRESENTATION (PROFILE) ET CRÉATION



LA GESTION DES ETATS - REACT NATIVE

LE PROJET EN FIL ROUGE

CRÉATION

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ npx create-expo-app@latest --template
✔ Choose a template: › Blank
✔ What is your app named? … book-app
✔ Downloaded and extracted project files.
> npm install
```

```
added 1188 packages, and audited 1189 packages in 31s

131 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd book-app
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native$ cd book-app/
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ ls
App.js  app.json  assets  babel.config.js  node_modules  package.json  package-lock.json
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

LES MODULES EXTERNES

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm i @react-navigation/native

added 12 packages, and audited 1201 packages in 5s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npx expo install react-native-screens react-native-safe-area-context
› Installing 2 SDK 51.0.0 compatible native modules using npm
> npm install

added 4 packages, and audited 1205 packages in 4s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm i @react-navigation/stack

added 12 packages, and audited 1217 packages in 5s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npx expo install react-native-gesture-handler
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

up to date, audited 1217 packages in 4s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm i @react-navigation/bottom-tabs

added 1 package, and audited 1218 packages in 3s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

LES ECRANS DE BASE

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ mkdir -p src/surfaces
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ touch src/surfaces/Login.js
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ touch src/surfaces/Feed.js
```

Login.js U | Feed.js U | App.js M | package.json M

book-app > src > surfaces > Login.js > Login

```js
// ./src/surfaces/Login.js
import React from "react";
import { View, Text } from "react-native";
export const Login = () => {
    return (
        <View>
            <Text>this will be the login screen</Text>
        </View>
    );
};
```

Login.js U | Feed.js U | App.js M | package.json M

book-app > src > surfaces > Feed.js > Feed

```js
// ./src/surfaces/Feed.js
import React from "react";
import { View, Text } from "react-native";
export const Feed = () => {
  return (
    <View>
      <Text>this will be the feed screen</Text>
    </View>
  );
};
```

Login.js U | Feed.js U | App.js M | package.json M

book-app > App.js > Home

```js
// ./App.js
import 'react-native-gesture-handler';
import React, { useState } from "react";
import { NavigationContainer } from "@react-navigation/native";
import { createStackNavigator } from "@react-navigation/stack";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import { Login } from "./src/surfaces/Login";
import { Feed } from "./src/surfaces/Feed";
const Stack = createStackNavigator();
const Tab = createBottomTabNavigator();
function Home() {
  return (
    <Tab.Navigator>
      <Tab.Screen name="Feed" component={Feed} />
    </Tab.Navigator>
  );
}
export default function App() {
  const [userLoggedIn, setIsUserLoggedIn] = useState(true);
  return (
    <NavigationContainer>
      <Stack.Navigator>
        {!userLoggedIn ? (
          <Stack.Screen name="Login" component={Login} />
        ) : (
          <Stack.Screen
            name="Home"
            component={Home}
            options={{ headerShown: false }}
          />
        )}
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

LES ECRANS DE BASE

App.js M | Login.js U | Feed.js U | Conversations.js U

book-app > src > surfaces > Conversations.js > Conversations

```
1  import React from "react";
2  import { View, Text } from "react-native";
3  export const Conversations = () => {
4    return (
5      <View>
6        <Text>this will be the chat screen</Text>
7      </View>
8    );
9  };
```

App.js M | Login.js U | Feed.js U | Conversations.js U

book-app > src > surfaces > AddPost.js > ...

```
1  import React from "react";
2  import { View, Text } from "react-native";
3  export const AddPost = () => {
4    return (
5      <View>
6        <Text>this will be the AddPost screen</Text>
7      </View>
8    );
9  };
```

App.js M | Login.js U | Feed.js U | Conversations.js U

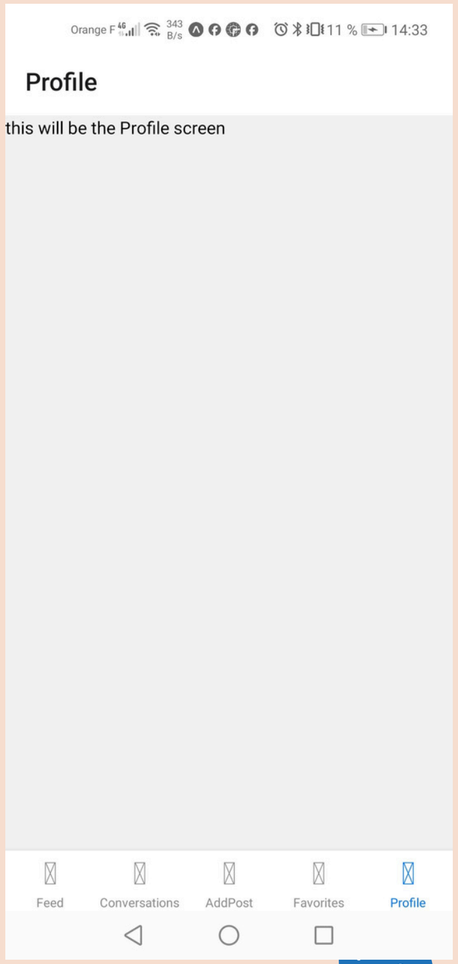
book-app > src > surfaces > Favorites.js > Favorites

```
1  import React from "react";
2  import { View, Text } from "react-native";
3  export const Favorites = () => {
4    return (
5      <View>
6        <Text>this will be the Favorites screen</Text>
7      </View>
8    );
9  };
```

App.js M | Login.js U | Feed.js U | Conversations.js U

book-app > src > surfaces > Profile.js > Profile

```
1  import React from "react";
2  import { View, Text } from "react-native";
3  export const Profile = () => {
4    return (
5      <View>
6        <Text>this will be the Profile screen</Text>
7      </View>
8    );
9  };
```



LA GESTION DES ETATS - REACT NATIVE

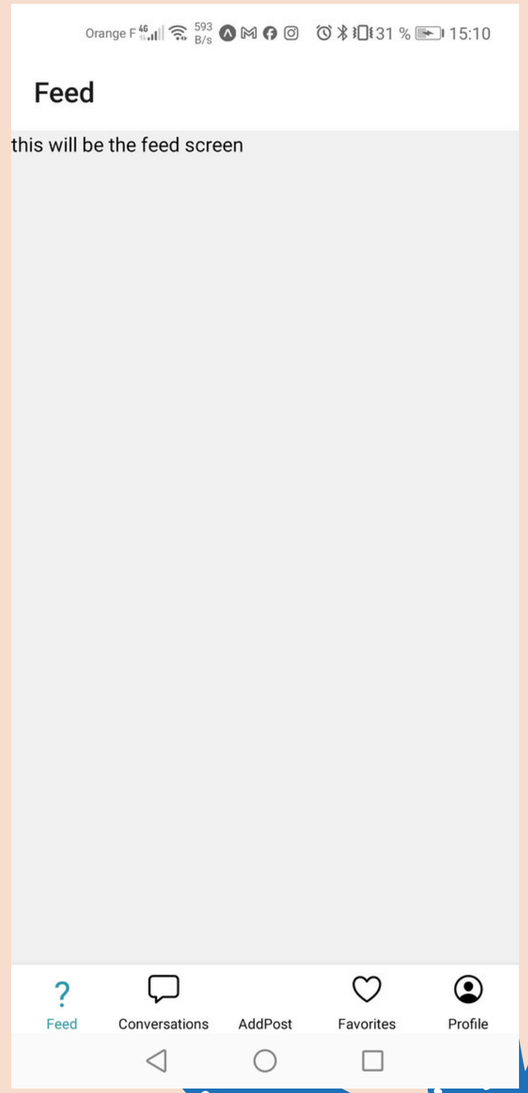
LE PROJET

LES ECRANS DE BASE

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npx expo install @expo/vector-icons
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

up to date, audited 1218 packages in 5s

132 packages are look
  run `npm fund` for

found 0 vulnerabiliti
```

```
const Stack = createStackNavigator();
const Tab = createBottomTabNavigator();
function Home() {
  return (
    <Tab.Navigator>
      <Tab.Screen name='Feed' component={Feed} />
      <Tab.Screen name='Conversations' component={Conversations} />
      <Tab.Screen name='AddPost' component={AddPost} />
      <Tab.Screen name='Favorites' component={Favorites} />
      <Tab.Screen name='Profile' component={Profile} />
    </Tab.Navigator>
  );
}
```

```
import { AddPost } from "./src/surfaces/AddPost";
import { Conversations } from "./src/surfaces/Conversations";
//import Ionicons from "@expo/vector-icons/Ionicons";
import { Ionicons } from '@expo/vector-icons';
//import Ionicons from 'react-native-vector-icons/Ionicons';

const Stack = createStackNavigator();
const Tab = createBottomTabNavigator();
function Home() {
  return (
    <Tab.Navigator screenOptions={({ route }) => ({
      tabBarIcon: ({ focused, color, size }) => {
        let iconName;
        if (route.name === "Feed") {
          iconName = focused ? "home" : "home-outline";
        } else if (route.name === "Conversations") {
          iconName = focused ? "chatbox" : "chatbox-outline";
        } else if (route.name === "Favorites") {
          iconName = focused ? "heart" : "heart-outline";
        } else if (route.name === "Profile") {
          iconName = focused ? "person-circle" : "person-circle-outline";
        }
        return <Ionicons name={iconName} size={size} color={color} />;
      },
      tabBarActiveTintColor: "#25A0B0",
      tabBarInactiveTintColor: "#000000",
    })}
    >
      <Tab.Screen name='Feed' component={Feed} />
      <Tab.Screen name='Conversations' component={Conversations} />
      <Tab.Screen name='AddPost' component={AddPost} />
      <Tab.Screen name='Favorites' component={Favorites} />
      <Tab.Screen name='Profile' component={Profile} />
    </Tab.Navigator>
```

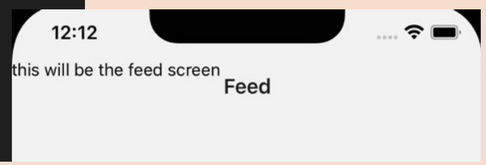


LA GESTION DES ETATS - REACT NATIVE

LE PROJET

SAFEAREAPROVIDER

```
import { Ionicons } from '@expo/vector-icons';
import { SafeAreaProvider } from 'react-native-safe-area-context';
//import Ionicons from 'react-native-vector-icons/Ionicons';

const Stack = createStackNavigator();
}
export default function App() {
  const [userLoggedIn, setIsUserLoggedIn] = useState(true);
  return (
    <SafeAreaProvider>
      <NavigationContainer>
        <Stack.Navigator>
          {!userLoggedIn ? (
            <Stack.Screen name="Login" component={Login} />
          ) : (
            <Stack.Screen
              name="Home"
              component={Home}
              options={{ headerShown: false }}
            />
          )}
        </Stack.Navigator>
      </NavigationContainer>
    </SafeAreaProvider>
  );
}
```

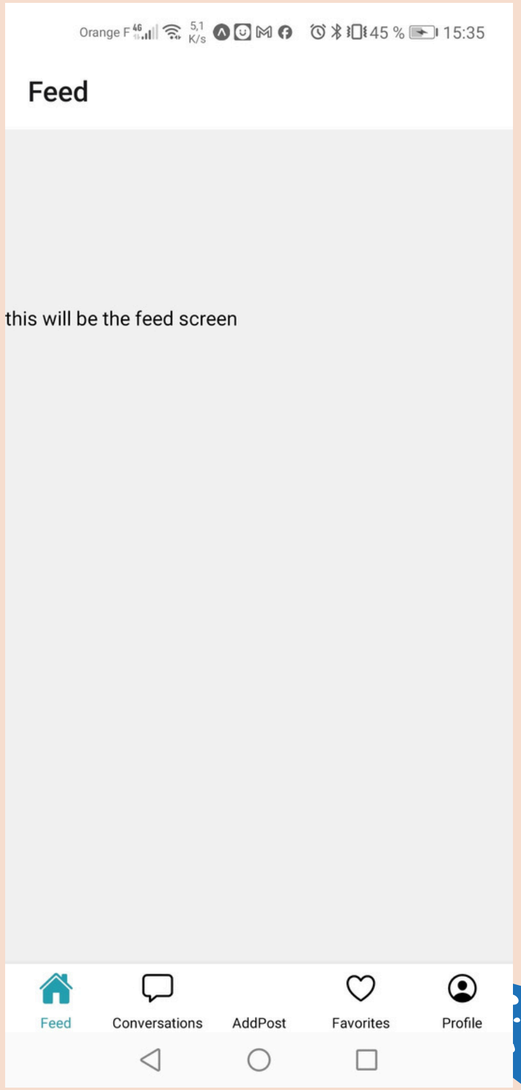


LA GESTION DES ETATS - REACT NATIVE

LE PROJET

USEHEADERHEIGHT()

```
import React from "react";
import { SafeAreaView } from "react-native-safe-area-context";
import { View, Text } from "react-native";
import { useHeaderHeight } from "@react-navigation/elements";
export const Feed = () => {
 const headerHeight = useHeaderHeight();
 return (
 <SafeAreaView style={{ flex: 1, paddingTop: headerHeight }}>
 <View>
 <Text>this will be the feed screen</Text>
 </View>
 </SafeAreaView>
 );
};
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

GOOGLE FONTS

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npx expo install expo-font
› Installing 1 SDK 51.0.0 compatible native module using npm
> npm install

up to date, audited 1223 packages in 6s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
› Added config plugin: expo-font
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npx expo install @expo-google-fonts/poppins
› Installing 1 other package using npm
> npm install --save @expo-google-fonts/poppins

added 1 package, and audited 1224 packages in 8s

132 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npx expo install expo-app-loading
› Installing 1 other package using npm
> npm install --save expo-app-loading
npm WARN deprecated glob@7.1.6: Glob versions prior to v9 are no longer supported

added 47 packages, and audited 1271 packages in 13s

132 packages are looking for funding
```

```
import AppLoading from "expo-app-loading";
import {
  useFonts,
  Poppins_400Regular,
  Poppins_700Bold
} from "@expo-google-fonts/poppins";
```

```
export default function App() {
  const [userLoggedIn, setIsUserLoggedIn] = useState(true);
  let [fontsLoaded] = useFonts({
    Poppins_400Regular,
    Poppins_700Bold
  });

  if (!fontsLoaded) {
    return <AppLoading />;
  }


  return (
    <SafeAreaProvider>
```

```
  );
}
export default function App() {
  const [userLoggedIn, setIsUserLoggedIn] = useState(true);
  let [fontsLoaded] = useFonts({
    Poppins_400Regular,
    Poppins_700Bold
  });

  // if (!fontsLoaded) {
  //   return <AppLoading />;
  // }

  if (!fontsLoaded) {
    return (
      <View>
        <Text>Loading fonts...</Text>
      </View>
    );
  }
```

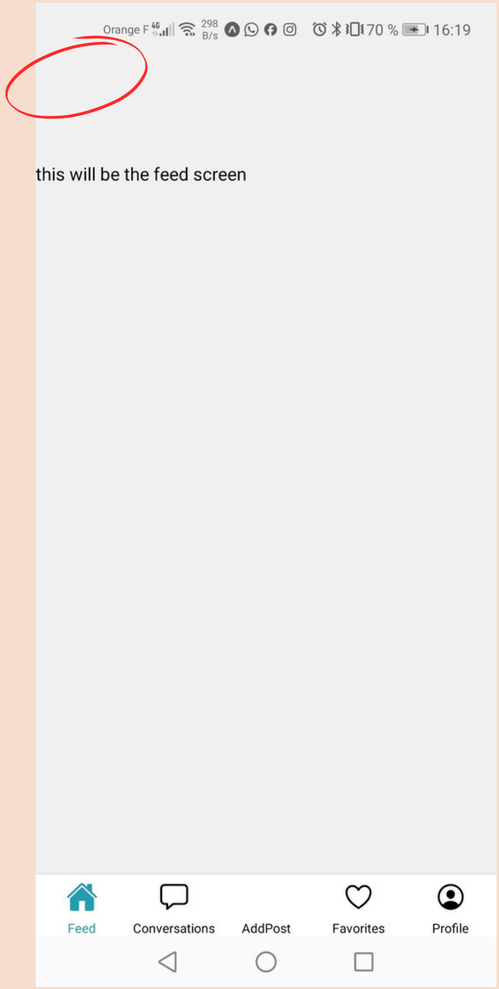
<Text style={{ fontFamily: "Poppins_400Regular"}}>

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

GOOGLE FONTS

```
const Tab = createBottomTabNavigator();
function Home() {
  return (
    <Tab.Navigator screenOptions={({ route }) => ({
      tabBarIcon: ({ focused, color, size }) => {
        let iconName;
        if (route.name === "Feed") {
          iconName = focused ? "home" : "home-outline";
        } else if (route.name === "Conversations") {
          iconName = focused ? "chatbox" : "chatbox-outline";
        } else if (route.name === "Favorites") {
          iconName = focused ? "heart" : "heart-outline";
        } else if (route.name === "Profile") {
          iconName = focused ? "person-circle" : "person-circle-outline";
        }
        return <Ionicons name={iconName} size={size} color={color} />;
      },
      tabBarActiveTintColor: "#25A0B0",
      tabBarInactiveTintColor: "#000000",
      //tabBarShowLabel: false,
      headerTransparent: true,
      headerTitleAlign: "right",
      headerTitleStyle: {
        paddingTop: 140,
        paddingBottom: 40,
        textAlign: "left",
        fontWeight: "bold",
      },
    })}
    >
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - LISTOFAVATARDS - LISTOFCARDS - LISTHEADERCOMPONENTS

book-app > src > surfaces > JS Feed.js > Feed

```jsx
import React from "react";
import { SafeAreaView } from "react-native-safe-area-context";
import { View, Text } from "react-native";
import { useHeaderHeight } from "@react-navigation/elements";
import { ListOfAvatars } from "../components/ListOfAvatars";
import { ListOfCards } from "../components/ListOfCards";
export const Feed = () => {
    const headerHeight = useHeaderHeight();
    return (
        <SafeAreaView
            style={{ flex: 1, paddingTop: headerHeight + 20, paddingHorizontal: 20 }}
        >
            <View>
                <ListOfAvatars />
                <ListOfCards />
            </View>
        </SafeAreaView>
    );
};
```

src > components > JS ListOfAvatars.js > ListOfAvatars > renderItem

```jsx
import React from "react";
import { View, FlatList, Pressable, Image } from "react-native";
import { ListHeaderComponent } from "./ListHeaderComponent";

export const ListOfAvatars = () => {
  const renderItem = ({ item }) => {
    return (
      <Pressable onPress={() => console.log("hello")}>
        <Image
          style={{ height: 56, width: 56, borderRadius: 28, marginRight: 30 }}
          source={{
            uri: item.url,
          }}
        />
      </Pressable>
    );
  };

  const arrayOfAvatars = [
    { id: 1, url: "https://randomuser.me/api/portraits/men/1.jpg" },
    { id: 2, url: "https://randomuser.me/api/portraits/women/2.jpg" },
    { id: 3, url: "https://randomuser.me/api/portraits/men/3.jpg" },
    { id: 4, url: "https://randomuser.me/api/portraits/women/4.jpg" },
    { id: 5, url: "https://randomuser.me/api/portraits/men/5.jpg" },
    { id: 6, url: "https://randomuser.me/api/portraits/women/6.jpg" },
    { id: 7, url: "https://randomuser.me/api/portraits/men/7.jpg" },
    { id: 8, url: "https://randomuser.me/api/portraits/women/8.jpg" },
    { id: 9, url: "https://randomuser.me/api/portraits/men/9.jpg" },
    { id: 10, url: "https://randomuser.me/api/portraits/women/10.jpg" }
  ];

    return (
        <View style={{
            zIndex: 100,
            paddingVertical: 30,
            paddingLeft: 20,
            backgroundColor: "rgba(255,255,255, 0.85)",
          }}
        >
          <FlatList
            data={arrayOfAvatars}
            renderItem={renderItem}
            keyExtractor={(item) => item.id}
            horizontal
            ListHeaderComponent={<ListHeaderComponent />}
            showsHorizontalScrollIndicator={false}
            snapToInterval={86}
            decelerationRate='fast'
          />
        </View>
      );
    };
```

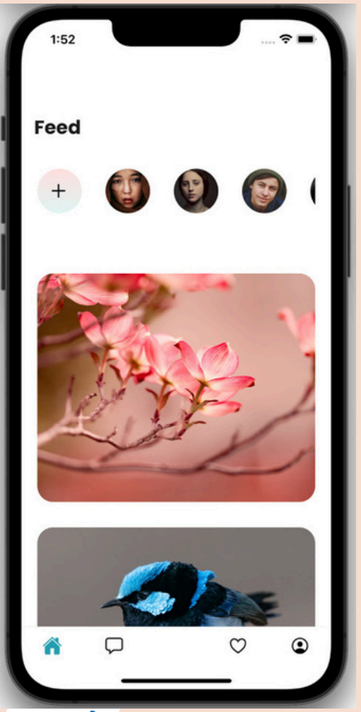
src > components > JS ListOfCards.js > ListOfCards

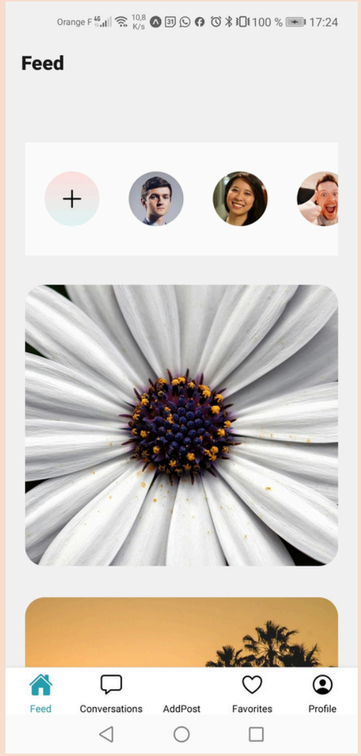
```jsx
import { Image, View } from "react-native";
import { FlatList } from "react-native-gesture-handler";

// src/components/ListOfCards.js
export const ListOfCards = () => {
    const renderItem = ({ item }) => {
      return (
        <Image
          style={{
            width: "100%",
            height: 288,
            borderRadius: 20,
            marginBottom: 32,
          }}
          source={{
            uri: item.url,
          }}
        />
      );
    };
    const arrayOfImages = […
      ]

    return (
      <View style={{ paddingVertical: 30 }}>
        <FlatList
          data={arrayOfImages}
          renderItem={renderItem}
          keyExtractor={(item) => item.id}
          showsVerticalScrollIndicator={false}
        />
      </View>
    );
  };
```

src > components > JS ListHeaderComponent.js > ListHeaderComponent

```jsx
import React from "react";
import { View, Pressable } from "react-native";
import Ionicons from "@expo/vector-icons/Ionicons";
import { LinearGradient } from "expo-linear-gradient";

export const ListHeaderComponent = () => {
  return (
    <Pressable onPress={() => console.log("pressed the add button")}>
      <View
        style={{
          width: 56,
          height: 56,
          marginRight: 30,
        }}
      >
        <LinearGradient
          colors={["#FFE1E0", "#E1F6F4"]}
          style={{
            borderRadius: 28,
            width: "100%",
            height: "100%",
            justifyContent: "center",
            alignItems: "center",
          }}
        >
          <Ionicons name="add" size={30} color="#000000" />
        </LinearGradient>
      </View>
    </Pressable>
  );
};
```

```jsx
};
const arrayOfImages = [
    {
      "itemId": 101,
      "authorId": 11,
      "timeStamp": "2 hrs ago",
      "url": "https://images.unsplash.com/photo-1653546226145-91aa8ce0cc96?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1376&q=80",
      "likes": "28",
      "conversations": "12"
    },
```





LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (APP)

App.js > App

```
1
2  import 'react-native-gesture-handler';
3  import React, { useState } from "react";
4  import { NavigationContainer } from "@react-navigation/native";
5  import { createStackNavigator } from "@react-navigation/stack";
6  import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
7  import { Login } from "./src/surfaces/Login";
8  import { SafeAreaProvider } from 'react-native-safe-area-context';
9  import {
10   useFonts,
11   Poppins_400Regular,
12   Poppins_700Bold
13 } from "@expo-google-fonts/poppins";
14 import { View, Text } from 'react-native';
15 import { ConversationsNavigation } from './src/surfaces/ConversationsNavigation';
16 import { Home } from './src/surfaces/Home';
17
18 const Stack = createStackNavigator();
19 const Tab = createBottomTabNavigator();
20
21 export default function App() {
22   const [userLoggedIn, setIsUserLoggedIn] = useState(true);
23   let [fontsLoaded] = useFonts({
24     Poppins_400Regular,
25     Poppins_700Bold
26   });
27
28   if (!fontsLoaded) {
29     return (
30       <View>
31         <Text>Loading fonts...</Text>
32       </View>
33     )
34 }
35
36
37   return (
38     <SafeAreaProvider>
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm i react-native-gesture-handler

up to date, audited 1272 packages in 11s

132 packages are looking for funding
  run `npm fund` for details
```

App.js > App

```
21 export default function App() {
29     return (
30       <View>
31         <Text>Loading fonts...</Text>
32       </View>
33     )
34   }
35
36
37   return (
38     <SafeAreaProvider>
39       <NavigationContainer>
40         <Stack.Navigator>
41           {!userLoggedIn ? (
42             <Stack.Screen name="Login" component={Login} />
43           ) : (
44             <>
45             <Stack.Screen
46               name="Home"
47               component={Home}
48               options={{ headerShown: false }}
49             />
50             <Stack.Screen
51                 name='ConversationsNav'
52                 component={ConversationsNavigation}
53                 options={{ headerShown: false }}
54               />
55               </>
56           )}
57         </Stack.Navigator>
58       </NavigationContainer>
59     </SafeAreaProvider>
60   );
61 }
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (HOME)

src > surfaces > JS Home.js > ...

```
1  import "react-native-gesture-handler";
2  import React from "react";
3  import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
4  import { Feed } from "./Feed";
5  import { Profile } from "./Profile";
6  import { Favorites } from "./Favorites";
7  import { AddPost } from "./AddPost";
8  import Ionicons from "@expo/vector-icons/Ionicons";
9  import { colors } from "../styles/colors";
10 import { View } from "react-native";
11
12 const Tab = createBottomTabNavigator();
13
14 const ConversationsBase = () => <View style={{ flex: 1 }} />;
15
16 export const Home = () => {
17   return (
18     <Tab.Navigator
19       screenOptions={({ route }) => ({
20         tabBarIcon: ({ focused, color, size }) => {
21           let iconName;
22
23           if (route.name === "Feed") {
24             iconName = focused ? "home" : "home-outline";
25           } else if (route.name === "Favorites") {
26             iconName = focused ? "heart" : "heart-outline";
27           } else if (route.name === "Profile") {
28             iconName = focused ? "person-circle" : "person-circle-outline";
29           }
30
31           return <Ionicons name={iconName} size={size} color={color} />;
32         },
33         tabBarActiveTintColor: colors.accentStroke,
34         tabBarInactiveTintColor: colors.black,
```

src > surfaces > JS Home.js > ...

```
16 export const Home = () => {
19       screenOptions={({ route }) => ({
20         tabBarIcon: ({ focused, color, size }) => {
29           }
30
31           return <Ionicons name={iconName} size={size} color={color} />;
32         },
33         tabBarActiveTintColor: colors.accentStroke,
34         tabBarInactiveTintColor: colors.black,
35         tabBarShowLabel: false,
36         headerTransparent: true,
37         headerTitleAlign: "left",
38         headerStyle: {
39           height: 160,
40         },
41         headerTitleStyle: {
42           textAlign: "left",
43           fontWeight: "bold",
44           fontFamily: "Poppins_700Bold",
45           fontSize: 24,
46         },
47       })}
48     >
49       <Tab.Screen name='Feed' component={Feed} />
50       <Tab.Screen
51         name='ConversationsMain'
52         component={ConversationsBase}
53         options={{
54           tabBarIcon: ({ size }) => (
55             <Ionicons name='chatbox-outline' color='#000000' size={size} />
56           ),
57         }}
58         listeners={({ navigation }) => ({
59           tabPress: (e) => {
```

src > surfaces > JS Home.js > ...

```
16 export const Home = () => {
58         listeners={({ navigation }) => ({
59           tabPress: (e) => {
60             e.preventDefault();
61             navigation.navigate("ConversationsNav");
62           },
63         })}
64       />
65       <Tab.Screen
66         name='AddPost'
67         component={AddPost}
68         options={{
69           tabBarIcon: ({ size }) => (
70             <View
71               style={{
72                 marginTop: -30,
73               }}
74             >
75               <View
76                 style={{
77                   position: "absolute",
78                   backgroundColor: "#000000",
79                   padding: 30,
80                   bottom: -10,
81                   left: -13,
82                   borderRadius: 23,
83                   transform: [{ rotate: "-45deg" }],
84                   shadowColor: "#000000",
85                   shadowOffset: { width: 0, height: 4 },
86                   shadowOpacity: 0.2,
87                   shadowRadius: 4,
88                 }}
89               />
90               <Ionicons name='add-circle-outline' color='#ffffff' size={36} />
91             </View>
92           ),
93         }}
94       />
95       <Tab.Screen name='Favorites' component={Favorites} />
96       <Tab.Screen
97         name='Profile'
98         component={Profile}
99         options={{ headerShown: false }}
100      />
101    </Tab.Navigator>
102  );
103 };
```

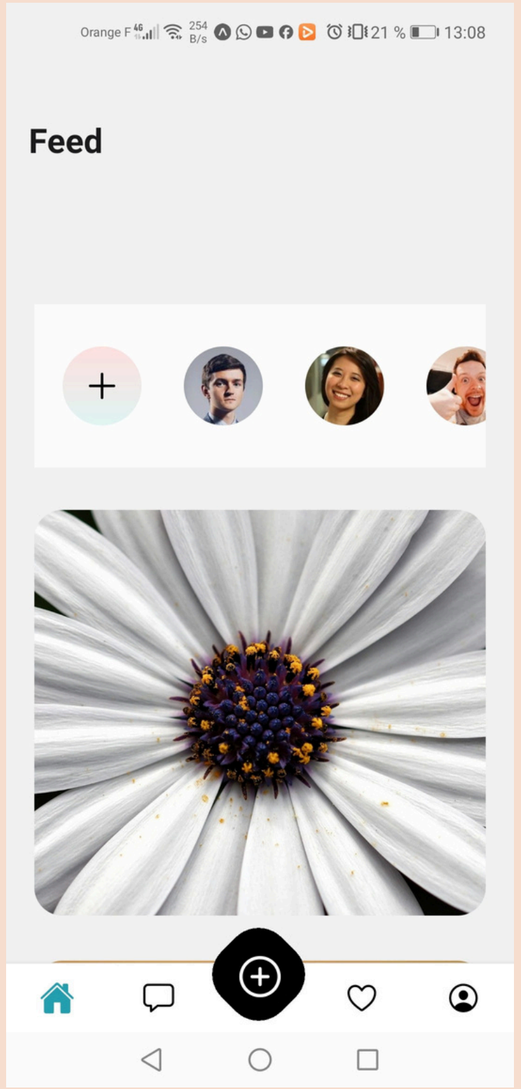
LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (CONVERSATIONSNAVIGATION)

src > surfaces > ConversationsNavigation.js > ConversationsNavigation > headerTitleStyle

```
1  import { Conversations } from "./Conversations";
2  import { Messages } from "./Messages";
3  import { createStackNavigator } from "@react-navigation/stack";
4
5  const Stack = createStackNavigator();
6
7  export const ConversationsNavigation = () => {
8    //const [conversationId, setConversationId] = useState(null);
9    return (
10
11       <Stack.Navigator
12         screenOptions={{
13           headerBackTitleVisible: false,
14           headerTintColor: "#000000",
15           headerTransparent: true,
16           headerTitleAlign: "left",
17           headerStyle: {
18             height: 160,
19           },
20           headerTitleStyle: {
21             textAlign: "left",
22             fontWeight: "bold",
23             fontFamily: "Poppins_700Bold",
24             fontSize: 24,
25           },
26         }}
27       >
28         <Stack.Screen name='Conversations' component={Conversations} />
29         <Stack.Screen
30           name='Messages'
31           component={Messages}
32           options={({ route }) => ({
33             title: route.params.name,
34             headerTitleStyle: {
35               textAlign: "center",
36               fontFamily: "Poppins_400Regular",
37               fontSize: 20,
38               position: "absolute",
39               top: 100,
40               left: 120,
41             },
42           })}
43         />
44       </Stack.Navigator>
45
46   );
47 };
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (CONVERSATIONS)

src > surfaces > JS Conversations.js > Conversations

```
1   import React from "react";
2   import { View, TextInput, Pressable } from "react-native";
3   import { useHeaderHeight } from "@react-navigation/elements";
4   import { SafeAreaView } from "react-native-safe-area-context";
5   import Ionicons from "@expo/vector-icons/Ionicons";
6   import { ListOfConvos } from "../components/ListOfConvos";
7
8   export const Conversations = ({ navigation }) => {
9     const headerHeight = useHeaderHeight();
10    const [text, onChangeText] = React.useState();
11
12    return (
13      <SafeAreaView style={{ flex: 1, paddingTop: headerHeight - 30 }}>
14        <View
15          style={{
16            width: 650,
17            height: 570,
18            borderRadius: 155,
19            borderWidth: 1,
20            borderColor: "#EEF2E2",
21            position: "absolute",
22            top: 210,
23            left: -160,
24            transform: [{ rotate: "-45deg" }],
25          }}
26        />
27        <View
28          style={{
29            width: 650,
30            height: 570,
31            borderRadius: 155,
32            borderWidth: 1,
33            borderColor: "#EEF2E2",
34            position: "absolute",
35            top: 280,
36            left: -160,
37            transform: [{ rotate: "-45deg" }],
38          }}
39        />
40        <View
41          style={{
42            width: 650,
43            height: 570,
44            borderRadius: 155,
45            position: "absolute",
```

src > surfaces > JS Conversations.js > Conversations

```
8   export const Conversations = ({ navigation }) => {
44            borderRadius: 155,
45            position: "absolute",
46            top: 350,
47            left: -160,
48            backgroundColor: "#E1F6F4",
49            transform: [{ rotate: "-45deg" }],
50          }}
51        />
52        <View style={{ marginHorizontal: 30, position: "relative" }}>
53          <View>
54            <TextInput
55              style={{
56                fontSize: 14,
57                paddingVertical: 12,
58                paddingLeft: 40,
59                marginHorizontal: 17,
60                borderRadius: 15,
61                backgroundColor: "#ffffff",
62                shadowColor: "#000000",
63                shadowOffset: { width: 0, height: 4 },
64                shadowOpacity: 0.1,
65                shadowRadius: 9,
66              }}
67              onChangeText={onChangeText}
68              value={text}
69              placeholder='search contacts'
70            />
71            <Ionicons
72              name='search'
73              size={24}
74              color='#000000'
75              style={{ position: "absolute", left: 28, top: 6 }}
76            />
77          </View>
78          <ListOfConvos navigation={navigation} />
79        </View>
80      </SafeAreaView>
81    );
82  };
```

src > styles > JS colors.js > ...

```
1   export const colors = {
2     black: "#000000",
3     grey: "#4E4E4E",
4     lightGrey: "#C4C4C4",
5     white: "#FFFFFF",
6     accentStroke: "#25A0B0",
7     mainTheme: "#E1F6F4",
8     pink: "#F5C8C6",
9     accentLines: "#EEF2E2",
10  };
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (CONVERSATIONS)

src > components > JS ListOfConvos.js > ...

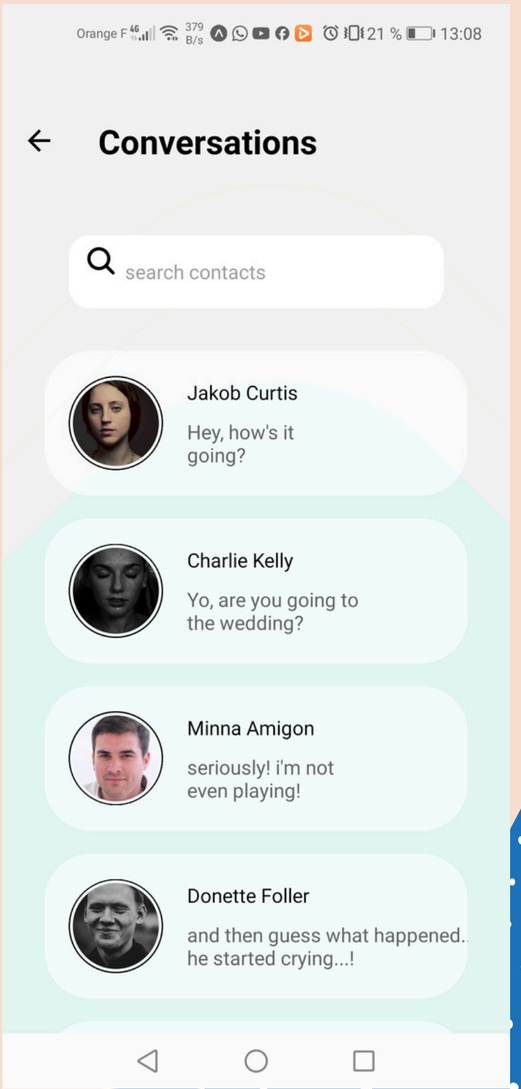
```
1   import React, { useState, useEffect } from "react";
2   import { View, FlatList, Pressable, Image, Text } from "react-native";
3
4   export const ListOfConvos = ({ navigation }) => {
5
6   const userList = [ ⋯
107   ]
108
109   const conversationsList = [ ⋯
145   ]
146
147   const renderItem = ({ item }) => {
148
149     const currentUser = userList.filter(
150       (user) => user.id === item.userId
151     );
152
153     return (
154       <Pressable
155       onPress={() => navigation.navigate("Messages", {name:currentUser[0].name, avatar:currentUser[0].url})}
156       style={{
157         height: 103,
158         backgroundColor: "rgba(255,255,255,0.6)",
159         borderRadius: 33,
160         marginBottom: 16,
161         justifyContent: "flex-start",
162         alignItems: "center",
163         flexDirection: "row",
164       }}
165       >
166         <View>
167           <View
168             style={{
169               width: 67,
170               height: 67,
171               borderRadius: 35,
172               borderColor: "#000000",
173               borderWidth: 1,
174               marginHorizontal: 17,
175             }}
176           >
177             <Image
```

src > components > JS ListOfConvos.js > [o] ListOfConvos > [o] renderItem

```
4   export const ListOfConvos = ({ navigation }) => {
147   const renderItem = ({ item }) => {
176           >
177             <Image
178               style={{
179                 width: 61,
180                 height: 61,
181                 borderRadius: 35,
182                 marginTop: 2,
183                 marginLeft: 2,
184               }}
185               source={{
186                 uri: currentUser[0].url,
187               }}
188             />
189           </View>
190         </View>
191         <View>
192           <Text style={{ fontSize: 14, paddingBottom: 9 }}>
193             {currentUser[0].name}
194           </Text>
195           <Text style={{ color: "#656565", width: "65%" }}>
196             {item.text}
197           </Text>
198         </View>
199       </Pressable>
200     )
201   };
202
203   return (
204     <View
205       style={{
206         paddingTop: 30,
207         marginTop: -30,
208         marginBottom: 50,
209       }}
210     >
211       <FlatList
212         data={conversationsList}
213         renderItem={renderItem}
214         keyExtractor={(item) => item.id}
215         showsVerticalScrollIndicator={false}
216         snapToInterval={119}
217         decelerationRate='fast'
218         ListHeaderComponent={<View style={{ height: 30 }} />}
219       />
220     </View>
221   );
222 };
```

```
6   const userList = [
7     {
8       "id": 1,
9       "name": "Malena Tudi",
10      "url": "https://raw.githubusercontent.com/PacktPublishing/Simplifying-State-Management-in-React-Native/cd04c474053275d4e22a8173695a2b972d012567/chapter-4/assets/avatars/1.png"
11    },
12    {
```

```
109   const conversationsList = [
110     {
111       "id": 1,
112       "userId": 2,
113       "text": "Hey, how's it going?"
114     },
115     {
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (MESSAGES)

src > surfaces > Messages.js > Messages

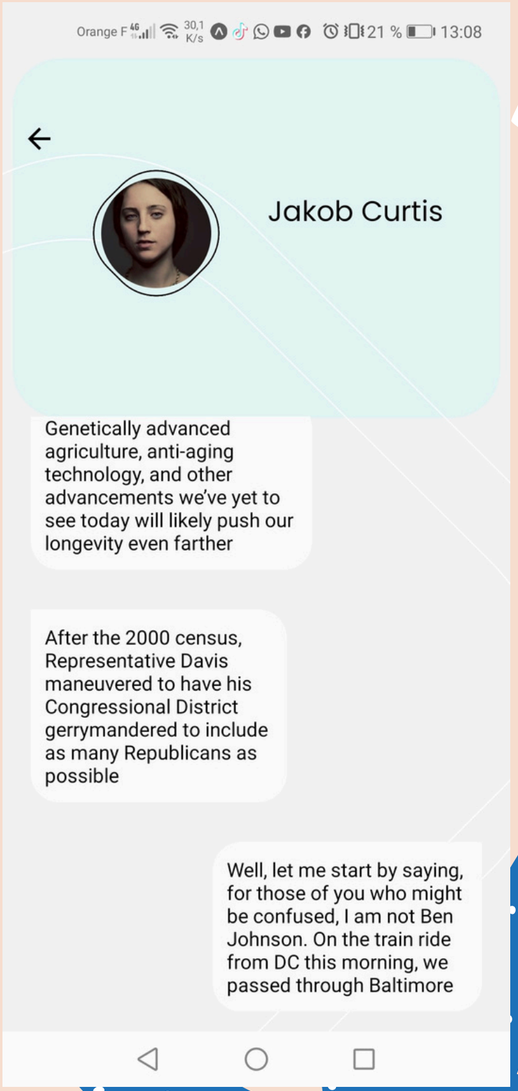
```
import React from "react";
import { View, Text, Image } from "react-native";
import { useHeaderHeight } from "@react-navigation/elements";
import { SafeAreaView } from "react-native-safe-area-context";
import { ListOfMessages } from "../components/ListOfMessages";


export const Messages = ({ route }) => {
  const headerHeight = useHeaderHeight();
  console.log("log route in message surface ",JSON.stringify(route))

  return (
    <SafeAreaView style={{ flex: 1, paddingTop: headerHeight + 100 }}>
          <>
            <View
              style={{
                position: "absolute",
                top: 40,
                left: "2%",
                backgroundColor: "#E1F6F4",
                width: "96%",
                height: 255,
                borderRadius: 34,
              }}
            />
            <View
              style={{
                width: 650,
                height: 570,
                borderRadius: 155,
                borderWidth: 1,
                borderColor: "#ffffff",
                position: "absolute",
                top: 250,
                left: -300,
                transform: [{ rotate: "-45deg" }],
              }}
            />
            <View
              style={{
```

src > surfaces > Messages.js > Messages

```
export const Messages = ({ route }) => {
            />
            <View
              style={{
                width: 650,
                height: 570,
                borderRadius: 155,
                borderWidth: 1,
                borderColor: "#ffffff",
                position: "absolute",
                top: 190,
                left: -290,
                transform: [{ rotate: "-45deg" }],
              }}
            />
            <View
              style={{
                position: "absolute",
                top: 125,
                left: 70,
              }}
            >
              <View
                style={{
                  width: 84,
                  height: 84,
                  borderRadius: 35,
                  borderColor: "#000000",
                  borderWidth: 1,
                  transform: [{ rotate: "-45deg" }],
                  position: "absolute",
                  top: -3,
                  left: -3,
                }}
              />
              <Image
                style={{ height: 78, width: 78, borderRadius: 40 }}
                source={{
                  uri: route.params.avatar,
                }}
              />
            </View>
            <ListOfMessages conversationId="2" />
          </>
    </SafeAreaView>
  );
};
```

src > components > ListOfMessages.js > styles > text

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

export const ListOfMessages = () => {
> const messages = {…
  }

  const renderItem = ({ item }) => {
    return (
      <View
        style={[
          styles.text,
          item.type === "from" ? styles.textTo : styles.textFrom,
        ]}
      >
        <Text style={{}}>{item.text}</Text>
      </View>
    );
  };
  return (
    <View
      style={{
        paddingHorizontal: 20,
      }}
    >
      <FlatList
        data={messages.messages}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        showsVerticalScrollIndicator={false}
        inverted
      />
    </View>
  );
};
const styles = StyleSheet.create({
  text: {
    backgroundColor: "#FAFAFA",
    borderBottomRightRadius: 20,
    borderBottomLeftRadius: 20,
    fontSize: 14,
    padding: 10,
    maxWidth: "65%",
    marginVertical: 14,
  },
  textFrom: {
    borderTopLeftRadius: 20,
    alignSelf: "flex-end",
  },
  textTo: {
    borderTopRightRadius: 20,
    alignSelf: "flex-start",
  },
});
```



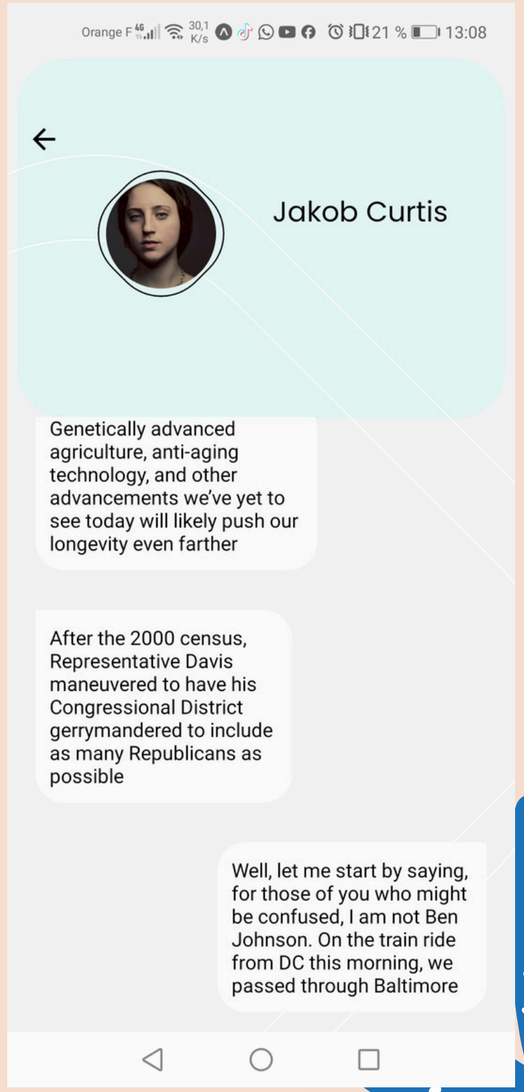
LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (MESSAGES)

src > components > ListOfMessages.js > ListOfMessages > messages > "messages"

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

export const ListOfMessages = () => {
const messages = {
    "id": 2,
    "messages": [
      {
        "id": 1,
        "type": "to",
        "text": "Well, let me start by saying, for those of you who might be confused, I am not Ben Johnson. On the train ride from DC this morning, we passed through Baltimore"
      },
      {
        "id": 2,
        "type": "from",
        "text": "After the 2000 census, Representative Davis maneuvered to have his Congressional District gerrymandered to include as many Republicans as possible"
      },
      {
        "id": 3,
        "type": "from",
        "text": "Genetically advanced agriculture, anti-aging technology, and other advancements we've yet to see today will likely push our longevity even farther"
      },
      {
        "id": 4,
        "type": "to",
        "text": "dIt shouldn't be surprising, then, that many of the biggest proponents of banning performance enhancing drugs in sports are also suspect of some The Rosenkranz Foundation"
      }
    ]
  }

  const renderItem = ({ item }) => {
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

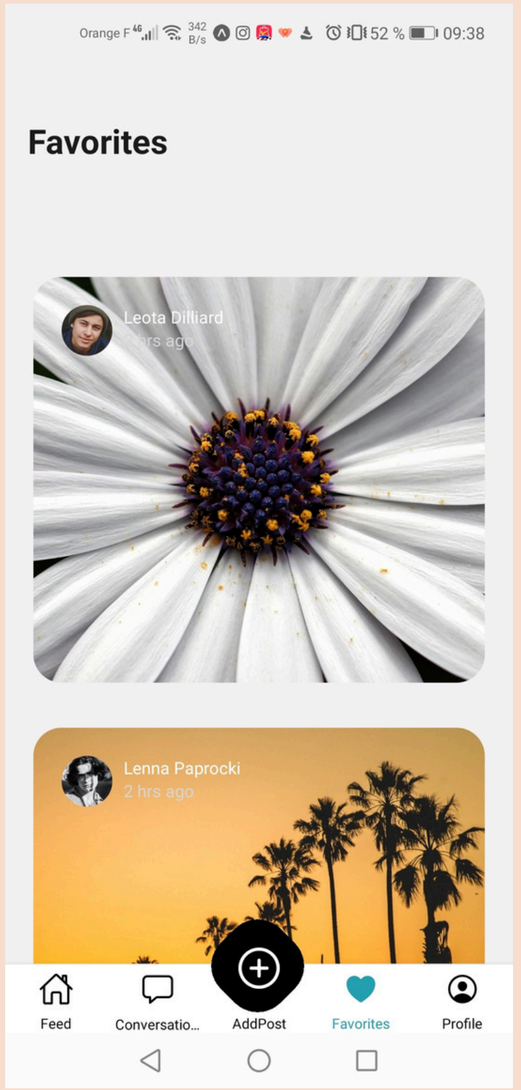
CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (FAVORITES)

src > surfaces > JS Favorites.js > ...

```jsx
import React from "react";
import { useHeaderHeight } from "@react-navigation/elements";
import { SafeAreaView } from "react-native-safe-area-context";
import { ListOfFavorites } from "../components/ListOfFavorites";

export const Favorites = () => {
  const headerHeight = useHeaderHeight();

  return (
    <SafeAreaView style={{ flex: 1, paddingTop: headerHeight }}>
      <ListOfFavorites />
    </SafeAreaView>
  );
};
```

src > components > JS ListOfFavorites.js > ListOfFavorites

```jsx
import React, { useState, useEffect } from "react";
import { View, FlatList } from "react-native";
import { Card } from "../components/Card";
import AppLoading from "expo-app-loading";
import { requestBase } from "../utils/constants";

export const ListOfFavorites = () => {
  const [cardList, setCardList] = useState(null);

  async function fetchCardData() {
    const response = await fetch(requestBase + "/home.json");
    setCardList(await response.json());
  }

  useEffect(() => {
    fetchCardData();
  }, []);

  if (!cardList) {
    return <AppLoading />;
  }
  const renderItem = ({ item }) => {
    return <Card item={item} />;
  };
  return (
    <View
      style={{
        paddingHorizontal: 20,
      }}
    >
      <FlatList
        data={cardList.reverse()}
        renderItem={renderItem}
        keyExtractor={(item) => item.itemId}
        showsVerticalScrollIndicator={false}
        snapToInterval={312}
        decelerationRate='fast'
      />
    </View>
  );
};
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (PROFILE)

src > surfaces > JS Profile.js > Profile

```
1  import React from "react";
2  import { View, Text, Image } from "react-native";
3  import { SafeAreaView } from "react-native-safe-area-context";
4  import { ProfileStatistics } from "../components/ProfileStatistics";
5  import { AddedImages } from "../components/AddedImages";
6  import { FavoritedImages } from "../components/FavoritedImages";
7  import Ionicons from "@expo/vector-icons/Ionicons";
8
9  export const Profile = () => {
10   return (
11     <SafeAreaView style={{ flex: 1 }}>
12       <View
13         style={{
14           width: 650,
15           height: 570,
16           borderRadius: 155,
17           borderWidth: 1,
18           borderColor: "#EEF2E2",
19           position: "absolute",
20           top: -210,
21           left: -120,
22           transform: [{ rotate: "-45deg" }],
23         }}
24       />
25       <View
26         style={{
27           width: 650,
28           height: 570,
29           borderRadius: 155,
30           borderWidth: 1,
31           borderColor: "#EEF2E2",
32           position: "absolute",
33           top: -260,
34           left: -120,
35           transform: [{ rotate: "-45deg" }],
36         }}
37       />
38       <View
39         style={{
40           width: 650,
41           height: 570,
```

src > surfaces > JS Profile.js > Profile

```
9  export const Profile = () => {
40           width: 650,
41           height: 570,
42           borderRadius: 155,
43           position: "absolute",
44           top: -320,
45           left: -120,
46           backgroundColor: "#E1F6F4",
47           transform: [{ rotate: "-45deg" }],
48         }}
49       />
50       <View style={{ paddingTop: 50 }}>
51         <View
52           style={{
53             width: 96,
54             height: 96,
55             borderRadius: 35,
56             borderColor: "#000000",
57             borderWidth: 1,
58             transform: [{ rotate: "-45deg" }],
59             alignSelf: "center",
60           }}
61         />
62
63         <View
64           style={{
65             overflow: "hidden",
66             alignSelf: "center",
67             transform: [{ rotate: "-45deg" }],
68             borderRadius: 35,
69             height: 87,
70             width: 87,
71             marginTop: -92,
72           }}
73         >
74           <Image
75             style={{
76               height: 105,
77               width: 105,
78               transform: [{ rotate: "45deg" }],
79               alignSelf: "center",
```

src > surfaces > JS Profile.js > Profile

```
9  export const Profile = () => {
80           }}
81           source={{
82             uri: "https://raw.githubusercontent.com/PacktPublishing
83           }}
84         />
85       </View>
86
87       <Text
88         style={{
89           fontFamily: "Poppins_700Bold",
90           fontSize: 26,
91           alignSelf: "center",
92           marginTop: 30,
93         }}
94       >
95         John Doe
96       </Text>
97       <Text
98         style={{
99           fontFamily: "Poppins_400Regular",
100          fontSize: 16,
101          alignSelf: "center",
102          marginTop: 10,
103        }}
104      >
105        @johndoe
106      </Text>
107    </View>
108    <ProfileStatistics />
109    <View
110      style={{
111        marginTop: 40,
112        flexDirection: "row",
113        width: "100%",
114        justifyContent: "space-evenly",
115      }}
116    >
117      <View style={{ alignItems: "center" }}>
118        <Ionicons name='image-outline' color='#000000' size={30} />
119        <AddedImages />
120      </View>
121      <View style={{ alignItems: "center" }}>
122        <Ionicons name='bookmark-outline' color='#000000' size={30}
123        <FavoritedImages />
124      </View>
125    </View>
126  </SafeAreaView>
127 );
128 };
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

CRÉATION ET STYLE DE COMPOSANTS : FEED - CONVERSATIONS - MESSAGES (PROFILE)

src > components > JS ProfileStatistics.js > ProfileStatistics

```jsx
import React from "react";
import { View, Text, StyleSheet } from "react-native";

export const ProfileStatistics = () => {
  return (
    <View
      style={{
        paddingTop: 70,
        paddingHorizontal: 20,
        flexDirection: "row",
        width: "100%",
        justifyContent: "space-evenly",
      }}
    >
      <View style={styles.general}>
        <Text style={styles.headerText}>Posts</Text>
        <Text style={styles.stats}>35</Text>
      </View>
      <View style={styles.general}>
        <Text style={styles.headerText}>Followers</Text>
        <Text style={styles.stats}>1,552</Text>
      </View>
      <View style={styles.general}>
        <Text style={styles.headerText}>Follows</Text>
        <Text style={styles.stats}>128</Text>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  general: {
    alignItems: "center",
  },
  headerText: {
    fontSize: 16,
    fontFamily: "Poppins_400Regular",
  },
  stats: {
    fontFamily: "Poppins_700Bold",
    fontSize: 25,
  },
});
```

src > components > JS AddedImages.js > ...

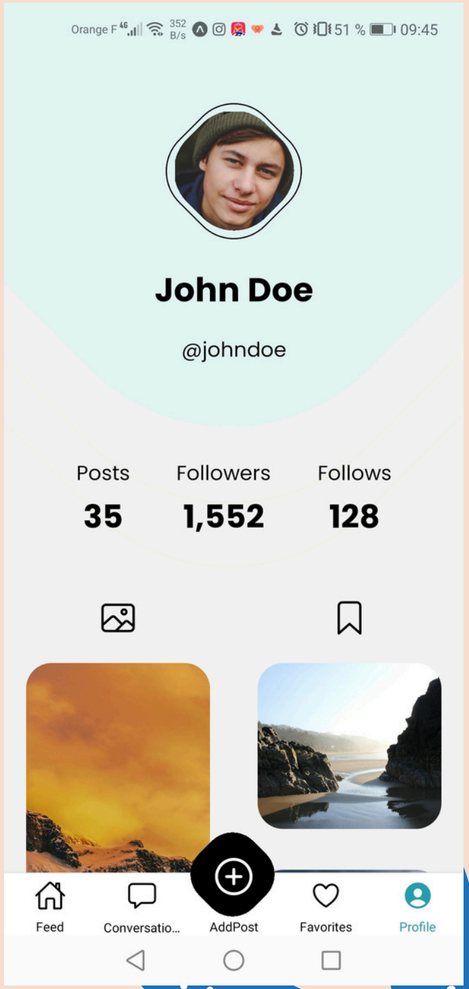
```jsx
import React from "react";
import {
  View,
  FlatList,
  Pressable,
  Image,
  useWindowDimensions,
} from "react-native";

const arrayOfImages = [ ...
```

Click to expand the range.

```jsx
export const AddedImages = () => {
  const imageWidth = useWindowDimensions().width * 0.4;

  const renderItem = ({ item }) => {
    return (
      <Image
        style={{
          width: imageWidth,
          height: 220,
          borderRadius: 20,
          marginBottom: 32,
          borderColor: "#000000",
        }}
        source={{
          uri: item.url,
        }}
      />
    );
  };
  return (
    <View
      style={{
        paddingHorizontal: 20,
        paddingTop: 20,
      }}
    >
      <FlatList
        data={arrayOfImages}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        showsVerticalScrollIndicator={false}
        snapToInterval={240}
        decelerationRate='fast'
      />
    </View>
  );
};
```

src > components > JS FavoritedImages.js > ...

```jsx
import React from "react";
import {
  View,
  FlatList,
  Pressable,
  Image,
  useWindowDimensions,
} from "react-native";

const arrayOfImages = [ ...
];

export const FavoritedImages = () => {
  const imageWidth = useWindowDimensions().width * 0.4;

  const renderItem = ({ item }) => {
    return (
      <Image
        style={{
          width: imageWidth,
          height: 130,
          borderRadius: 20,
          marginBottom: 32,
          borderColor: "#000000",
        }}
        source={{
          uri: item.url,
        }}
      />
    );
  };
  return (
    <View
      style={{
        paddingHorizontal: 20,
        paddingTop: 20,
      }}
    >
      <FlatList
        data={arrayOfImages}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        showsVerticalScrollIndicator={false}
      />
    </View>
  );
};
```




```jsx
const arrayOfImages = [
  {
    id: 1,
    url: "https://images.unsplash.com/photo-1654512799227-94e56fbed599?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1740&q=80",
  },
  {
    id: 2,
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (PROPS)

```
// src/App.js
export default function App() {
  const [userLoggedIn, setIsUserLoggedIn] = useState(true);
  const [userList, setUserList] = useState(null);
//...
  async function fetchUserData(id) {
    const response = await fetch(requestBase + "/users.json");
    setUserList(await response.json());
  }
  useEffect(() => {
    fetchUserData();
  }, []);
  //...
  if (!userList) {
    return <AppLoading />;
  }
```

```
<Stack.Screen name='Home' options={{ headerShown: false }}>
    {(props) => <Home {...props} userList={userList} />}
</Stack.Screen>
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (CONTEXT API)

src > JS context.js > ...

```
1   import React from "react";
2
3   export const UserListContext = React.createContext(null);
```

src > components > JS ListOfAvatars.js > ListOfAvatars > <function>

```
3   import { ListHeaderComponent } from "./ListHeaderComponent";
4   import { UserListContext } from "../context";
5
6   export const ListOfAvatars = () => {
7     const renderItem = ({ item }) => {
8       return (
9         <Pressable onPress={() => console.log("hello")}>
10          <Image
11            style={{ height: 56, width: 56, borderRadius: 28, marginRight: 30 }}
12            source={{
13              uri: item.url,
14            }}
15          />
16        </Pressable>
17      );
18    };
19    return (
20      <UserListContext.Consumer>
21        {({ userList }) => (
22          <View style={{
23            zIndex: 100,
24            paddingVertical: 30,
25            paddingLeft: 20,
26            backgroundColor: "rgba(255,255,255, 0.85)",
27          }}
28          >
29            <FlatList
30              data={userList}
31              renderItem={renderItem}
32              keyExtractor={(item) => item.id}
33              horizontal
34              ListHeaderComponent={<ListHeaderComponent />}
35              showsHorizontalScrollIndicator={false}
36              snapToInterval={86}
37              decelerationRate='fast'
38            />
39          </View>
40        )}
41      </UserListContext.Consumer>
42    )
43  }
```

JS App.js > App

```
17  import { UserListContext } from './src/context';
18  import { requestBase } from "./src/utils/constants";
19
20
21  const Stack = createStackNavigator();
22  const Tab = createBottomTabNavigator();
23
24  export default function App() {
25    const [userLoggedIn, setIsUserLoggedIn] = useState(true);
26    const [userList, setUserList] = useState(null);
27
28    async function fetchUserData(id) {
29      const response = await fetch(requestBase + "/users.json");
30      setUserList(await response.json());
31    }
32    useEffect(() => {
33      fetchUserData();
34    }, []);
```

JS App.js > App

```
24  export default function App() {
60    return (
61      <SafeAreaProvider>
62        <UserListContext.Provider value={{ userList: userList }}>
63        <NavigationContainer>
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (CONTEXT API) - LISTOFAVATARS / LISTOFCARDS

src > components > JS ListOfAvatars.js > [@] ListOfAvatars > <function>

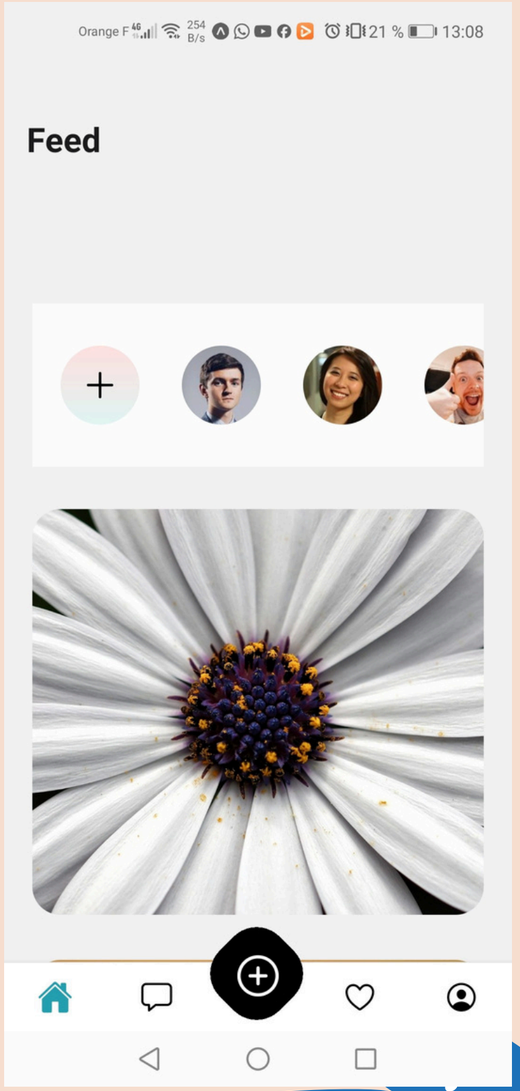
```
import { ListHeaderComponent } from "./ListHeaderComponent";
import { UserListContext } from "../context";

export const ListOfAvatars = () => {
  const renderItem = ({ item }) => {
    return (
      <Pressable onPress={() => console.log("hello")}>
        <Image
          style={{ height: 56, width: 56, borderRadius: 28, marginRight: 30 }}
          source={{
            uri: item.url,
          }}
        />
      </Pressable>
    );
  };
  return (
    <UserListContext.Consumer>
      {({ userList }) => (
        <View style={{
          zIndex: 100,
          paddingVertical: 30,
          paddingLeft: 20,
          backgroundColor: "rgba(255,255,255, 0.85)",
        }}
        >
          <FlatList
            data={userList}
            renderItem={renderItem}
            keyExtractor={(item) => item.id}
            horizontal
            ListHeaderComponent={<ListHeaderComponent />}
            showsHorizontalScrollIndicator={false}
            snapToInterval={86}
            decelerationRate='fast'
          />
        </View>
      )}
    </UserListContext.Consumer>
  )
}
```

src > components > JS ListOfCards.js > ...

```
import { useEffect, useState } from "react";
import { View, Text } from "react-native";
import { Card } from "../components/Card";
import { FlatList } from "react-native-gesture-handler";
import { requestBase } from "../utils/constants";

export const ListOfCards = () => {

  const [cardList, setCardList] = useState(null);
  async function fetchCardData() {
    try {
      console.log(`requestBase ${requestBase}`)
      const response = await fetch(requestBase + "/home.json");
      const data = await response.json()
      console.log(`data json ok => ${JSON.stringify(data)}`)
      setCardList(data);
    } catch (error) {
      console.log(`listcard fetch error ${error.message}`)
    }

  }
  useEffect(() => {
    fetchCardData();
  }, []);

  if (!cardList) {
    return (
      <View>
        <Text>Loading cardList ...</Text>
      </View>
    )
  }
  const renderItem = ({ item }) => {
    return <Card item={item} />;
  };

  return (
    <View
      style={{
        marginTop: -200,
        paddingHorizontal: 20,
        marginBottom: 160,
      }}
    >
      <FlatList
        data={cardList}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        showsVerticalScrollIndicator={false}
      />
    </View>
  );
};
```

src > components > JS Card.js > [@] Card > <function>

```
import { View, Text, Image } from "react-native";
import { UserListContext } from "../context";

export const Card = ({ item }) => {
  return (
    <UserListContext.Consumer>
      {({ userList }) => {
        const currentUser = userList.filter(
          (user) => user.id === item.authorId
        );
        return (
          <View>
            <Image
              style={{
                width: "100%",
                height: 288,
                borderRadius: 20,
                marginBottom: 32,
              }}
              source={{
                uri: item.url,
              }}
            />
            <View
              style={{
                position: "absolute",
                top: 20,
                left: 20,
                flexDirection: "row",
              }}
            >
              <Image
                style={{
                  width: 36,
                  height: 36,
                  borderRadius: 20,
                  marginRight: 8,
                }}
                source={{
                  uri: currentUser[0].url,
                }}
              />
              <View>
                <Text style={{ color: "#ffffff", fontSize: 12 }}>
                  {currentUser[0].name}
                </Text>
                <Text style={{ color: "#D8D8D8", fontSize: 12 }}>
                  2 hrs ago
                </Text>
              </View>
            </View>
          </View>
        );
      }}
    </UserListContext.Consumer>
  );
};
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (CONTEXT API) - CONVERSATIONS - LISTOFCONVOS

src > context.js > ...

```js
import React from "react";

export const UserListContext = React.createContext(null);
export const ConversationContext = React.createContext(null);
```

src > surfaces > ConversationsNavigation.js > ConversationsNavigation > <function>

```js
import { Conversations } from "./Conversations";
import { Messages } from "./Messages";
import { createStackNavigator } from "@react-navigation/stack";
import { ConversationContext } from "../context";
import { useState } from "react";

const Stack = createStackNavigator();

export const ConversationsNavigation = () => {
  const [conversationId, setConversationId] = useState(null);
  return (
    <ConversationContext.Provider
      value={{
        conversationId: conversationId,
        setConversationId: setConversationId,
      }}
    >
      <Stack.Navigator
        screenOptions={{
          headerBackTitleVisible: false,
          headerTintColor: "#000000",
          headerTransparent: true,
          headerTitleAlign: "left",
          headerStyle: {
            height: 160,
          },
          headerTitleStyle: {
            textAlign: "left",
            fontWeight: "bold",
            fontFamily: "Poppins_700Bold",
            fontSize: 24,
          },
        }}
      >
        <Stack.Screen name='Conversations' component={Conversations} />
        <Stack.Screen
          name='Messages'
          component={Messages}
          options={({ route }) => ({
            title: route.params.name,
            headerTitleStyle: {
              textAlign: "center",
              fontFamily: "Poppins_400Regular",
              fontSize: 20,
              position: "absolute",
              top: 100,
              left: 120,
            },
          })}
        />
      </Stack.Navigator>
    </ConversationContext.Provider>
  );
};
```

src > surfaces > Conversations.js > ...

```js
import React from "react";
import { View, TextInput, Pressable } from "react-native";
import { useHeaderHeight } from "@react-navigation/elements";
import { SafeAreaView } from "react-native-safe-area-context";
import Ionicons from "@expo/vector-icons/Ionicons";
import { ListOfConvos } from "../components/ListOfConvos";

export const Conversations = ({ navigation }) => {
  const headerHeight = useHeaderHeight();
  const [text, onChangeText] = React.useState();

  return (
    <SafeAreaView style={{ flex: 1, paddingTop: headerHeight - 30 }}>
      <View
        style={{
          width: 650,
          height: 570,
          borderRadius: 155,
          borderWidth: 1,
          borderColor: "#EEF2E2",
          position: "absolute",
          top: 210,
          left: -160,
          transform: [{ rotate: "-45deg" }],
        }}
      />
      <View
        style={{
          width: 650,
          height: 570,
          borderRadius: 155,
          borderWidth: 1,
          borderColor: "#EEF2E2",
          position: "absolute",
          top: 280,
          left: -160,
          transform: [{ rotate: "-45deg" }],
        }}
      />
      <View
        style={{
          width: 650,
          height: 570,
          borderRadius: 155,
          position: "absolute",
          top: 350,
          left: -160,
          backgroundColor: "#E1F6F4",
```

src > surfaces > Conversations.js > Conversations

```js
export const Conversations = ({ navigation }) => {
          top: 350,
          left: -160,
          backgroundColor: "#E1F6F4",
          transform: [{ rotate: "-45deg" }],
        }}
      />
      <View style={{ marginHorizontal: 30, position: "relative" }}>
        <View>
          <TextInput
            style={{
              fontSize: 14,
              paddingVertical: 12,
              paddingLeft: 40,
              marginHorizontal: 17,
              borderRadius: 15,
              backgroundColor: "#ffffff",
              shadowColor: "#000000",
              shadowOffset: { width: 0, height: 4 },
              shadowOpacity: 0.1,
              shadowRadius: 9,
            }}
            onChangeText={onChangeText}
            value={text}
            placeholder='search contacts'
          />
          <Ionicons
            name='search'
            size={24}
            color='#000000'
            style={{ position: "absolute", left: 28, top: 6 }}
          />
        </View>
        <ListOfConvos navigation={navigation} />
      </View>
    </SafeAreaView>
  );
};
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (CONTEXT API) - CONVERSATIONS - LISTOFCONVOS - CONVERSATIONITEM

src > components > JS ListOfConvos.js > ListOfConvos > renderItem

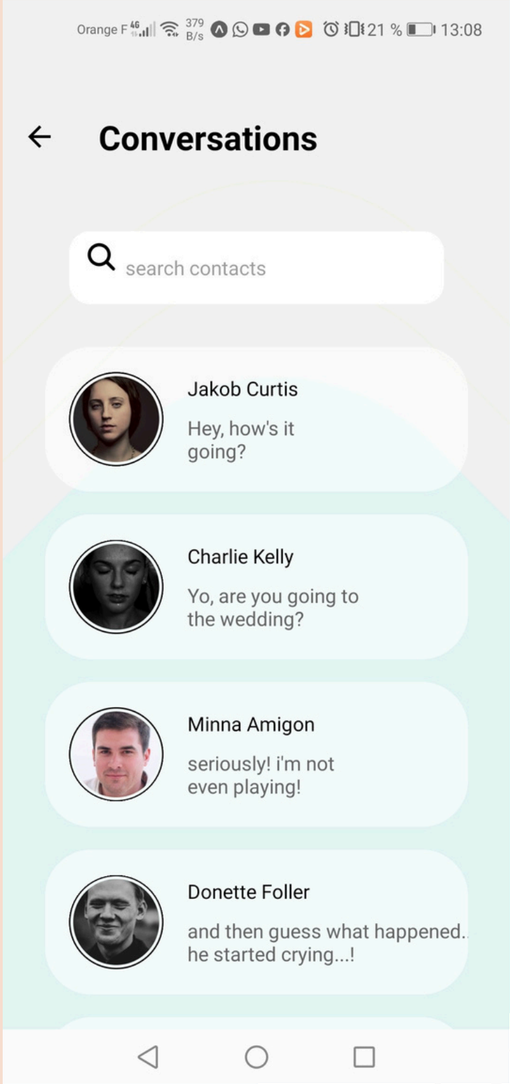
```
import React, { useState, useEffect } from "react";
import { View, FlatList} from "react-native";
import { ConversationItem } from "./ConversationItem";
import { requestBase } from "../utils/constants";
import Loading from "./Loading";

export const ListOfConvos = ({ navigation }) => {

  const [conversationsList, setConversationsList] = useState(null);

  async function fetchConversationData() {
    const response = await fetch(requestBase + "/conversations.json");
    setConversationsList(await response.json());
  }

  useEffect(() => {
    fetchConversationData();
  }, []);

  if (!conversationsList) {
    return <Loading message="Loading conv list"/>
  }

  const renderItem = ({ item }) => {
    return <ConversationItem navigation={navigation} item={item} />;
  };

  return (
    <View
      style={{
        paddingTop: 30,
        marginTop: -30,
        marginBottom: 50,
      }}
    >
      <FlatList
        data={conversationsList}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        showsVerticalScrollIndicator={false}
        snapToInterval={119}
        decelerationRate='fast'
        ListHeaderComponent={<View style={{ height: 30 }} />}
      />
    </View>
  );
};
```

src > components > JS ConversationItem.js > ConversationItem

```
import React from "react";
import { View, Pressable, Image, Text } from "react-native";
import { UserListContext, ConversationContext } from "../context";

export const ConversationItem = ({ navigation, item }) => {

  const onPressItem = (setConversationId, currentUser) => {
    setConversationId(item.id);
    navigation.navigate("Messages", {
      name: currentUser[0].name,
      avatar: currentUser[0].url,
    });
  };

  return (
    <ConversationContext.Consumer>
      {({ setConversationId }) => (
        <UserListContext.Consumer>
          {({ userList }) => {
            const currentUser = userList.filter(
              (user) => user.id === item.userId
            );
            return (
              <Pressable
                onPress={() => onPressItem(setConversationId, currentUser)}
                style={{
                  height: 103,
                  backgroundColor: "rgba(255,255,255,0.6)",
                  borderRadius: 33,
                  marginBottom: 16,
                  justifyContent: "flex-start",
                  alignItems: "center",
                  flexDirection: "row",
                }}
              >
                <View>
                  <View
                    style={{
                      width: 67,
                      height: 67,
                      borderRadius: 35,
                      borderColor: "#000000",
                      borderWidth: 1,
                      marginHorizontal: 17,
                    }}
                  >
                    <Image
```

src > components > JS ConversationItem.js > ConversationItem > <function> > <function>

```
export const ConversationItem = ({ navigation, item }) => {
      {({ setConversationId }) => (
        {({ userList }) => {
                  >
                    <Image
                      style={{
                        width: 61,
                        height: 61,
                        borderRadius: 35,
                        marginTop: 2,
                        marginLeft: 2,
                      }}
                      source={{
                        uri: currentUser[0].url,
                      }}
                    />
                  </View>
                </View>
                <View>
                  <Text style={{ fontSize: 14, paddingBottom: 9 }}>
                    {currentUser[0].name}
                  </Text>
                  <Text style={{ color: "#656565", width: "65%" }}>
                    {item.text}
                  </Text>
                </View>
              </Pressable>
            );
          }}
        </UserListContext.Consumer>
      )}
    </ConversationContext.Consumer>
  );
};
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (CONTEXT API) - CONVERSATIONS - MESSAGES - LISTOFMESSAGES

src > surfaces > JS Messages.js > Messages

```
import React from "react";
import { View, Image } from "react-native";
import { useHeaderHeight } from "@react-navigation/elements";
import { SafeAreaView } from "react-native-safe-area-context";
import { ListOfMessages } from "../components/ListOfMessages";
import { ConversationContext } from "../context";

export const Messages = ({ route }) => {
  const headerHeight = useHeaderHeight();

  return (
    <SafeAreaView style={{ flex: 1, paddingTop: headerHeight + 100 }}>
      <ConversationContext.Consumer>
        {({ conversationId }) => (
          <>
            <View
              style={{
                position: "absolute",
                top: 40,
                left: "2%",
                backgroundColor: "#E1F6F4",
                width: "96%",
                height: 255,
                borderRadius: 34,
              }}
            />
            <View
              style={{
                width: 650,
                height: 570,
                borderRadius: 155,
                borderWidth: 1,
                borderColor: "#ffffff",
                position: "absolute",
                top: 250,
                left: -300,
                transform: [{ rotate: "-45deg" }],
              }}
            />
            <View
              style={{
                width: 650,
                height: 570,
                borderRadius: 155,
                borderWidth: 1,
                borderColor: "#ffffff",
                position: "absolute",
```

src > surfaces > JS Messages.js > Messages > <function>

```
export const Messages = ({ route }) => {
        {({ conversationId }) => (
                borderWidth: 1,
                borderColor: "#ffffff",
                position: "absolute",
                top: 190,
                left: -290,
                transform: [{ rotate: "-45deg" }],
              }}
            />
            <View
              style={{
                position: "absolute",
                top: 125,
                left: 70,
              }}
            >
              <View
                style={{
                  width: 84,
                  height: 84,
                  borderRadius: 35,
                  borderColor: "#000000",
                  borderWidth: 1,
                  transform: [{ rotate: "-45deg" }],
                  position: "absolute",
                  top: -3,
                  left: -3,
                }}
              />
              <Image
                style={{ height: 78, width: 78, borderRadius: 40 }}
                source={{
                  uri: route.params.avatar,
                }}
              />
            </View>
            <ListOfMessages conversationId={conversationId} />
          </>
        )}
      </ConversationContext.Consumer>
    </SafeAreaView>
  );
};
```

src > components > JS ListOfMessages.js > ListOfMessages

```
import React, { useState, useEffect } from "react";
import { View, FlatList, Text, StyleSheet } from "react-native";
import { requestBase } from "../utils/constants";
import Loading from "./Loading";

export const ListOfMessages = ({ conversationId }) => {
  const [messages, setMessages] = useState(null);

  async function fetchMessages() {
    const response = await fetch(
      requestBase + "/messages/" + conversationId + ".json"
    );
    setMessages(await response.json());
  }

  useEffect(() => {
    fetchMessages();
  }, []);
  if (!messages) {
    return <Loading message="loading message"/>
  }

  const renderItem = ({ item }) => {
    return (
      <View
        style={[
          styles.text,
          item.type === "from" ? styles.textTo : styles.textFrom,
        ]}
      >
        <Text style={{}}>{item.text}</Text>
      </View>
    );
  };
  return (
    <View
      style={{
        paddingHorizontal: 20,
      }}
    >
      <FlatList
        data={messages.messages}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
```

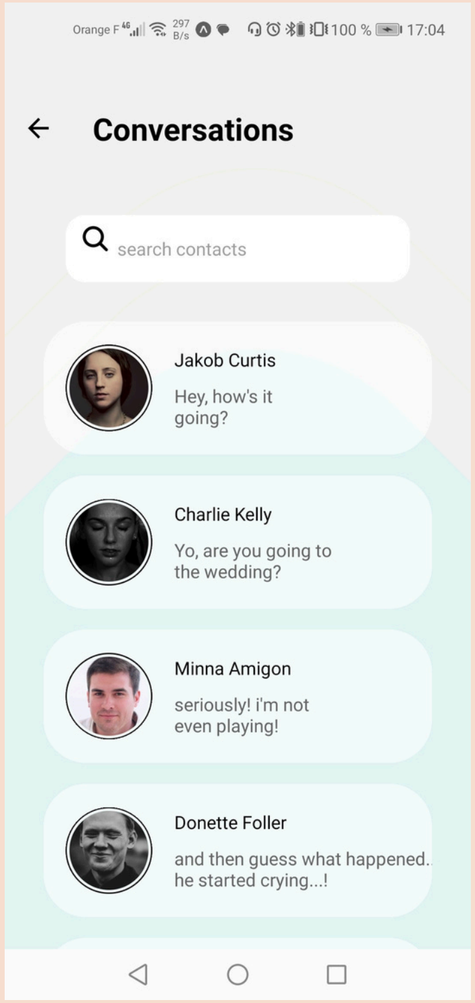
src > components > JS ListOfMessages.js > ...

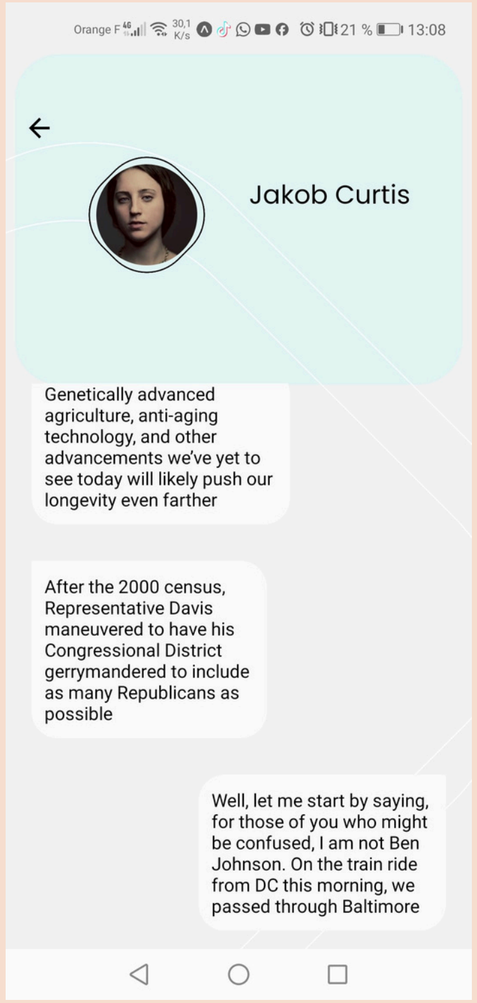
```
export const ListOfMessages = ({ conversationId }) => {
        data={messages.messages}
        renderItem={renderItem}
        keyExtractor={(item) => item.id}
        showsVerticalScrollIndicator={false}
        inverted
      />
    </View>
  );
};

const styles = StyleSheet.create({
  text: {
    backgroundColor: "#FAFAFA",
    borderBottomRightRadius: 20,
    borderBottomLeftRadius: 20,
    fontSize: 14,
    padding: 10,
    maxWidth: "65%",
    marginVertical: 14,
  },
  textFrom: {
    borderTopLeftRadius: 20,
    alignSelf: "flex-end",
  },
  textTo: {
    borderTopRightRadius: 20,
    alignSelf: "flex-start",
  },
});
```

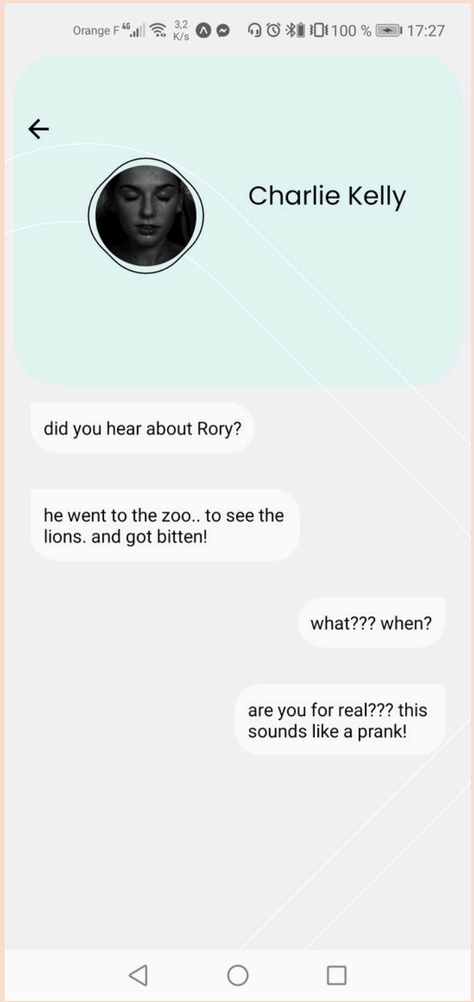
LA GESTION DES ETATS - REACT NATIVE

LE PROJET

UTILISATION DES ÉTATS : EXTRAIRE DES DONNÉES POUR L'APPLICATION (CONTEXT API) - CONVERSATIONS - MESSAGES - LISTOFMESSAGES







LA GESTION DES ETATS - REACT NATIVE

LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : STORE - ACTIONS - REDUCERS

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm install @reduxjs/toolkit

added 5 packages, and audited 1277 packages in 4s

133 packages are looking for funding
  run `npm fund` for details
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm install react-redux

added 3 packages, and audited 1280 packages in 3s

133 packages are looking for funding
  run `npm fund` for details
```

```
balde@balde-ASUS-TUF-Gaming-F15-FX507ZC4-FX567ZC4:~/www/react_native/book-app$ npm install --save-dev @redux-devtools/core

added 2 packages, and audited 1282 packages in 4s

133 packages are looking for funding
  run `npm fund` for details
```

store.js U user.js U

store.js > ...

```
export const user = {
    userLoggedIn: false,
    userData: null,
};
```

store.js U user.js U

reducers > user.js > login

```
import { user } from "../store";
export const login = (state=user, action) => {
  if (action.type === 'LOGIN') {
    return {
        ...state,
        user: {
          userLoggedIn: true,
          user: action.payload,
        },
    }
  } else {
    return state
  }
}
```

store.js U user.js U

reducers > user.js > login

```
import { user } from "../store";
export const login = (state=user, action) => {
  switch (action.type) {
    case "LOGIN": {
      return {
        ...state,
        user: {
          userLoggedIn: true,
          user: action.payload,
        },
      };
    }
    case "LOGOUT": {
      return {
        ...state,
        user: {
          userLoggedIn: false,
          user: null,
        },
      };
    }
    default:
      return state;
  }
}
```

store.js U user.js U likedImages.js U

reducers > likedImages.js > likedImagesReducer

```
export const likedImagesReducer = (state = [], action) => {
    switch (action.type) {
      case "LIKE_IMAGE": {
        const newLikedImage = action.payload;
        return [...state, newLikedImage];
      }
      case "UNLIKE_IMAGE": {
        const stateWithoutLikedImage = state.filter(
          (item) => item !== action.payload
        );
        return stateWithoutLikedImage;
      }
      default: {
        throw new Error(`Unhandled action type: ${action.type}`);
      }
    }
  };
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : COMBINAISON DE REDUCERS - TOOLKIT REDUX

store.js U | user.js U

```
export const user = {
    userLoggedIn: false,
    userData: null,
};
```

store.js U | user.js U | likedImages.js U

```
import { configureStore } from "@reduxjs/toolkit";
import usersReducer from "./reducers/users";
import likedImagesReducer from "./reducers/likedImages";
export const store = configureStore({
  reducer: {
    user: usersReducer,
    likedImages: likedImagesReducer,
  },
});
```

store.js U | App.js M | user.js U | likedImages.js U

```
19 import Loading from './src/components/Loading';
20 import { store } from "./store";
21 import { Provider } from "react-redux";
22
```

App.js > App

```
27 export default function App() {
53
54   return (
55     <SafeAreaProvider>
56       <Provider store={store}>
57       <UserListContext.Provider value={{ userList: userList }}>
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : COMBINAISON DE REDUCERS - TOOLKIT REDUX : FAVORITE

reducers > JS likedImages.js > [⊙] likedImagesSlice

```js
import { createSlice } from "@reduxjs/toolkit";
import { fetchLikedImages } from "../asyncFetches";

export const likedImagesSlice = createSlice({
  name: "likedImages",
  initialState: {
    likedImages: [],
    loading: true,
  },
  reducers: {
    likeImage: (state, action) => {
      const newLikedImage = action.payload;
      return { ...state, likedImages: [...state.likedImages, newLikedImage] };
    },
    unLikeImage: (state, action) => {
      const stateWithoutLikedImage = state.likedImages.filter(
        (item) => item.itemId !== action.payload.itemId
      );
      state.likedImages = stateWithoutLikedImage;
    },
  },
  extraReducers: (builder) => {
    builder.addCase(fetchLikedImages.pending, (state) => {
      state.loading = true;
    });
    builder.addCase(fetchLikedImages.fulfilled, (state, action) => {
      state.likedImages = action.payload;
      state.loading = false;
    });
    builder.addCase(fetchLikedImages.rejected, (state) => {
      state.loading = false;
    });
  },
});

export const { initLikedImages, likeImage, unLikeImage } =
  likedImagesSlice.actions;

export default likedImagesSlice.reducer;
```

JS asyncFetches.js > ...

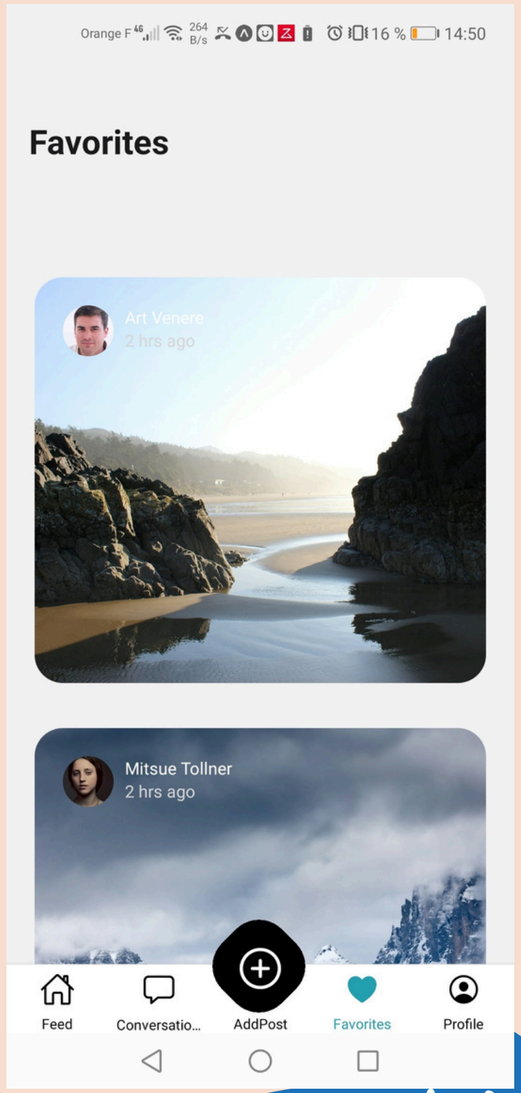
```js
import { createAsyncThunk } from "@reduxjs/toolkit";
import { requestBase } from "./src/utils/constants";

export const fetchLikedImages = createAsyncThunk(
  "likedImages/initLikedImages",
  async () => {
    const response = await fetch(requestBase + "/john_doe/likedImages.json");
    return await response.json();
  }
);
```

src > surfaces > JS Home.js > ...

```js
import { View } from "react-native";
import { fetchLikedImages } from "../../asyncFetches";
import { useDispatch } from "react-redux";

const Tab = createBottomTabNavigator();

const ConversationsBase = () => <View style={{ flex: 1 }} />;

export const Home = () => {

  const dispatch = useDispatch();

  useEffect(() => {
    dispatch(fetchLikedImages());
  }, []);

  return (
    <Tab.Navigator
```

src > components > JS ListOfFavorites.js > [⊙] ListOfFavorites > useEffect() callback

```js
import React, { useState, useEffect } from "react";
import { View, FlatList } from "react-native";
import { Card } from "../components/Card";
import { useSelector } from "react-redux";
import Loading from "./Loading";

export const ListOfFavorites = ({ navigation }) => {
  const { likedImages } = useSelector((state) => state.likedImages);
  const [imageList, setImageList] = useState([]);

  if (!imageList) {
    return <Loading message="loading image list"/>
  }

  useEffect(() => {
    const reversedImages = [...likedImages].reverse();
    setImageList(reversedImages);
  }, [likedImages]);

  const renderItem = ({ item }) => {
    return <Card item={item} navigation={navigation} />;
  };
  return (
    <View
      style={{
        paddingHorizontal: 20,
      }}
    >
      <FlatList
        data={imageList}
        renderItem={renderItem}
        keyExtractor={(item) => item.itemId}
        showsVerticalScrollIndicator={false}
        snapToInterval={312}
        decelerationRate='fast'
      />
    </View>
  );
};
```



LA GESTION DES ETATS - REACT NATIVE

LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : COMBINAISON DE REDUCERS - TOOLKIT REDUX : USERDETAILSMODAL

App.js > App > fetchUserData

```
import Loading from './src/components/Loading';
import store from "./store";
import { Provider } from "react-redux";
import { UserDetailsModal } from './src/surfaces/UserDetailsModal';

```

```
<SafeAreaProvider>
  <Provider store={store}>
    <UserListContext.Provider value={{ userList: userList }}>
      <NavigationContainer>
      <Stack.Navigator>
              <Stack.Group>
                {!userLoggedIn ? (
                  <Stack.Screen name='Login' component={Login} />
                ) : (
                  <>
                    <Stack.Screen
                      name='Home'
                      component={Home}
                      options={{ headerShown: false }}
                    />
                    <Stack.Screen
                      name='ConversationsNav'
                      component={ConversationsNavigation}
                      options={{ headerShown: false }}
                    />
                  </>
                )}
              </Stack.Group>
              <Stack.Group screenOptions={{ presentation: "modal" }}>
                <Stack.Screen
                  name='UserDetailsModal'
                  component={UserDetailsModal}
                  options={{ headerShown: false }}
                />

              </Stack.Group>
            </Stack.Navigator>
      </NavigationContainer>
    </UserListContext.Provider>
  </Provider>
</SafeAreaProvider>
```

src > surfaces > UserDetailsModal.js > UserDetailsModal

```
import React from "react";
import { View, Text, Pressable, Image } from "react-native";
import { SafeAreaView } from "react-native-safe-area-context";
import Ionicons from "@expo/vector-icons/Ionicons";
import { UserDetailsModalImages } from "../components/UserDetailsModalImages";

export const UserDetailsModal = ({ navigation, route }) => {
  return (
    <SafeAreaView style={{ flex: 1, paddingTop: 30 }}>
      <View
        style={{
          width: 650,
          height: 570,
          borderRadius: 155,
          borderWidth: 1,
          borderColor: "#EEF2E2",
          position: "absolute",
          top: -80,
          left: -360,
          transform: [{ rotate: "-45deg" }],
        }}
      />
      <View
        style={{
          width: 650,
          height: 570,
          borderRadius: 155,
          borderWidth: 1,
          borderColor: "#EEF2E2",
          position: "absolute",
          top: -80,
          left: -410,
          transform: [{ rotate: "-45deg" }],
        }}
      />
      <View
        style={{
          width: 650,
          height: 570,
          borderRadius: 155,
          position: "absolute",
          top: -80,
          left: -460,
          backgroundColor: "#E1F6F4",
          transform: [{ rotate: "-45deg" }],
        }}
      />
      <Pressable
        onPress={() => navigation.goBack()}
        style={{ flexDirection: "row" }}
      >
        <Ionicons name='chevron-back-outline' size={30} color='#000000' />
        <Text
          style={{
            fontFamily: "Poppins_400Regular",
            fontSize: 18,
```

src > surfaces > UserDetailsModal.js > UserDetailsModal

```
export const UserDetailsModal = ({ navigation, route }) => {
            marginTop: 3,
          }}
        >
          Go back
        </Text>
      </Pressable>
      <View style={{ paddingTop: 20, marginLeft: 20, flexDirection: "row" }}>
        <View
          style={{
            width: 76,
            height: 76,
            borderRadius: 25,
            borderColor: "#000000",
            borderWidth: 1,
            transform: [{ rotate: "-45deg" }],
            alignSelf: "flex-start",
          }}
        />
        <View
          style={{
            overflow: "hidden",
            alignSelf: "flex-start",
            transform: [{ rotate: "-45deg" }],
            borderRadius: 25,
            height: 67,
            width: 67,
            marginLeft: -72,
            marginTop: 4,
          }}
        >
          <Image
            style={{
              height: 95,
              width: 95,
              transform: [{ rotate: "45deg" }],
              alignSelf: "center",
            }}
            source={{
              uri: route.params.user.url,
            }}
          />
        </View>
        <Text
          style={{
            fontFamily: "Poppins_700Bold",
            fontSize: 19,
            alignSelf: "center",
            marginLeft: 30,
          }}
        >
          {route.params.user.name}
        </Text>
      </View>
      <UserDetailsModalImages />
    </SafeAreaView>
  );
```

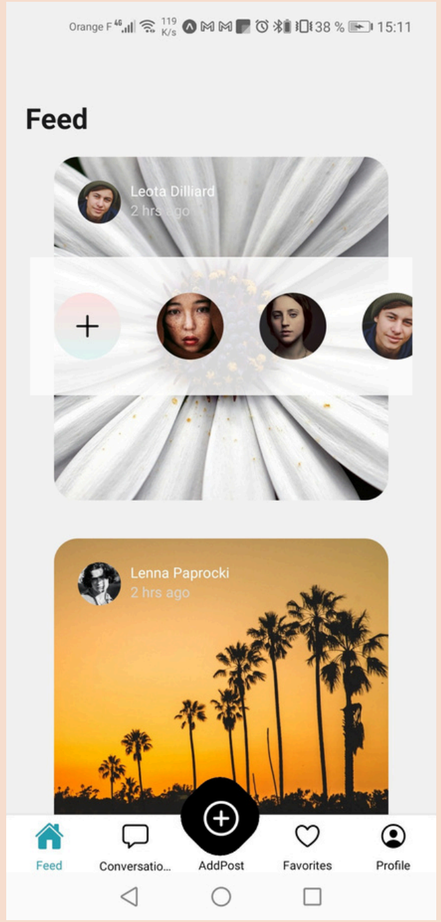
LA GESTION DES ETATS - REACT NATIVE

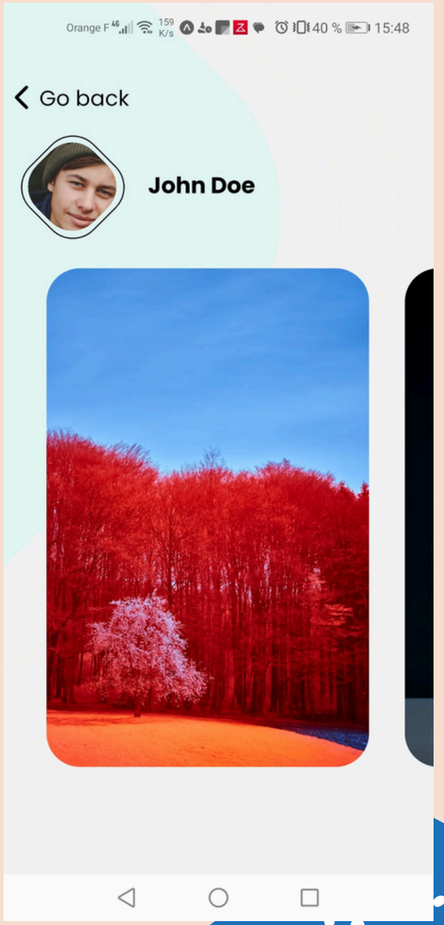
LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : COMBINAISON DE REDUCERS - TOOLKIT REDUX : USERDETAILSMODAL

src > components > JS UserDetailsModalImages.js > [⊙] UserDetailsModalImages

```
1   import React, { useState, useEffect } from "react";
2   import { FlatList, Image, View, Text } from "react-native";
3   import { requestBase } from "../utils/constants";
4   import { useWindowDimensions } from "react-native";
5   import Loading from "./Loading";
6
7   export const UserDetailsModalImages = () => {
8     const randomImageSet = Math.round(Math.random() * 4);
9     const [imageList, setImageList] = useState(null);
10    const { height, width } = useWindowDimensions();
11
12    async function fetchImageData() {
13      const response = await fetch(
14          requestBase + "/userImages/" + randomImageSet + ".json"
15      );
16      setImageList(await response.json());
17    }
18
19    useEffect(() => {
20      fetchImageData();
21    }, []);
22
23    if (!imageList) {
24      return <Loading message="Loading image list ...."/>
25    }
26    const renderItem = ({ item }) => {
27      return (
28        <Image
29          style={{
30            height: height * 0.6,
31            width: width * 0.75,
32            borderRadius: 28,
33            marginRight: 30,
34          }}
35          source={{
36            uri: item.url,
37          }}
38        />
39      );
40    };
41
42    return (
43      <View style={{ paddingTop: 30 }}>
44        <FlatList
45          data={imageList.listOfitems}
46          renderItem={renderItem}
47          keyExtractor={(item) => item.itemId}
48          horizontal
49          showsHorizontalScrollIndicator={false}
50          snapToInterval={width * 0.825}
51          decelerationRate='fast'
52          ListHeaderComponent={<View style={{ width: width * 0.1 }} />}
53        />
54      </View>
55    );
56  };
```





LA GESTION DES ETATS - REACT NATIVE

LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : COMBINAISON DE REDUCERS - TOOLKIT REDUX : IMAGEDETAILSMODAL

App.js > ...

```
import { Provider } from 'react-redux';
import { UserDetailsModal } from './src/surfaces/UserDetailsModal';
import { ImageDetailsModal } from './src/surfaces/ImageDetailsModal';
```

App.js > ...

```
export default function App() {
                    component={UserDetailsModal}
                    options={{ headerShown: false }}
                  />
                  <Stack.Screen
                    name='ImageDetailsModal'
                    component={ImageDetailsModal}
                    options={{ headerShown: false }}
                  />
                </Stack.Group>
```

src > components > ListOfCards.js > ListOfCards

```
export const ListOfCards = ({navigation}) => {
  const renderItem = ({ item }) => {
    return <Card item={item} navigation={navigation} />;
  };
```

src > components > Card.js > Card > <function>

```
export const Card = ({ item, navigation }) => {
  return (
    <UserListContext.Consumer>
      {({ userList }) => {
        const currentUser = userList.filter(
          (user) => user.id === item.authorId
        );
        return (
          <Pressable
            onPress={() =>
              navigation.navigate("ImageDetailsModal", { imageItem: item })
            }
          >
            <Image
              style={{
                width: "100%",
                height: 288,
                borderRadius: 20,
                marginBottom: 32,
              }}
              source={{
                uri: item.url,
              }}
            />
            <View
              style={{
                position: "absolute",
                top: 20,
                left: 20,
                flexDirection: "row",
              }}
            >
```

ImageDetailsModal.js > ImageDetailsModal

```
import React, { useEffect, useState } from "react";
import { View, Text, Pressable, Image, StyleSheet } from "react-native";
import { SafeAreaView } from "react-native-safe-area-context";
import Ionicons from "@expo/vector-icons/Ionicons";
import { likeImage, unLikeImage } from "../../reducers/likedImages";
import { useDispatch, useSelector } from "react-redux";

export const ImageDetailsModal = ({ navigation, route }) => {
const { likedImages } = useSelector((state) => state.likedImages);
const [isCurrentImageLiked, setIsCurrentImageLiked] = useState(false);
const dispatch = useDispatch();

useEffect(() => {
  const checkIfLiked =
    likedImages?.filter(
      (favoritedImg) => favoritedImg.itemId === route.params.imageIte
    ).length > 0;
  setIsCurrentImageLiked(checkIfLiked);
}, [likedImages]);

return (
  <SafeAreaView style={{ flex: 1, paddingTop: 30 }}>
    <View
      style={{
        width: 650,
        height: 570,
        borderRadius: 155,
        borderWidth: 1,
        borderColor: "#EEF2E2",
        position: "absolute",
        bottom: -260,
        left: -140,
        transform: [{ rotate: "-45deg" }],
      }}
    />
    <View
      style={{
        width: 650,
        height: 570,
        borderRadius: 155,
        borderWidth: 1,
        borderColor: "#EEF2E2",
        position: "absolute",
        bottom: -290,
        left: -140,
        transform: [{ rotate: "-45deg" }],
      }}
    />
    <View
      style={{
        width: 650,
        height: 570,
        borderRadius: 155,
        position: "absolute",
        bottom: -330,
        left: -140,
```

ImageDetailsModal.js > ImageDetailsModal

```
export const ImageDetailsModal = ({ navigation, route }) => {
        bottom: -330,
        left: -140,
        backgroundColor: "#E1F6F4",
        transform: [{ rotate: "-45deg" }],
      }}
    />
    <Pressable
      onPress={() => navigation.goBack()}
      style={{ flexDirection: "row" }}
    >
      <Ionicons name='chevron-back-outline' size={30} color='#000000' />
      <Text
        style={{
          fontFamily: "Poppins_400Regular",
          fontSize: 18,
          marginTop: 3,
        }}
      >
        Go back
      </Text>
    </Pressable>
    <View style={{ paddingTop: 30 }}>
      <Image
        style={{
          width: "100%",
          height: 288,
          marginBottom: 32,
        }}
        source={{
          uri: route.params.imageItem.url,
        }}
      />
    </View>
    <View
      style={{
        paddingTop: 70,
        paddingHorizontal: 20,
        flexDirection: "row",
        width: "100%",
        justifyContent: "space-evenly",
      }}
    >
      <View style={styles.general}>
        <Text style={styles.headerText}>Likes</Text>
        <Text style={styles.stats}>{route.params.imageItem.likes}</Te
      </View>
      <View style={styles.general}>
        <Text style={styles.headerText}>Conversations</Text>
        <Text style={styles.stats}>
          {route.params.imageItem.conversations}
        </Text>
      </View>
      <View style={styles.general}>
        <Text style={styles.headerText}>Follows</Text>
        <Text style={styles.stats}>128</Text>
```

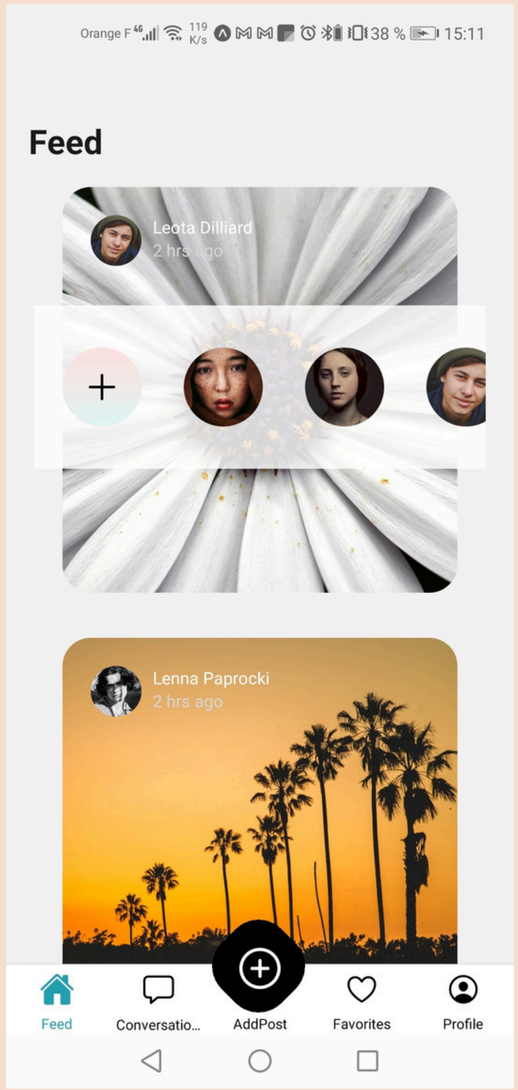
ImageDetailsModal.js > ImageDetailsModal

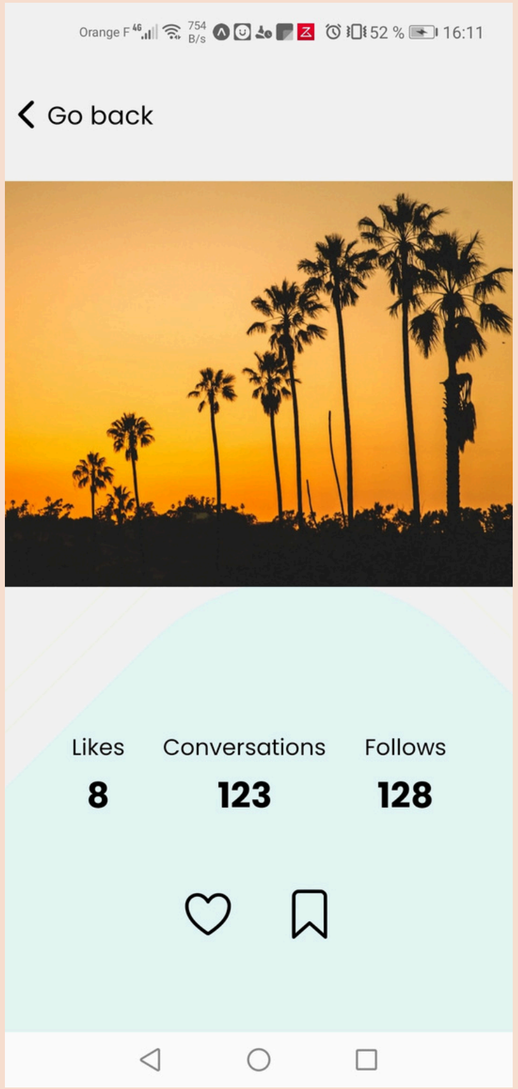
```
export const ImageDetailsModal = ({ navigation, route }) => {
        <Text style={styles.headerText}>Follows</Text>
        <Text style={styles.stats}>128</Text>
      </View>
    </View>
    <View
      style={{
        flexDirection: "row",
        justifyContent: "space-around",
        alignSelf: "center",
        marginTop: 40,
        width: "40%",
      }}
    >
      <Pressable
        onPress={() => {
          if (isCurrentImageLiked) {
            dispatch(unLikeImage(route.params.imageItem));
          } else {
            dispatch(likeImage(route.params.imageItem));
          }
        }}
      >
        <Ionicons
          name={isCurrentImageLiked ? "heart" : "heart-outline"}
          size={40}
          color='#000000'
        />
      </Pressable>
      <Pressable onPress={() => console.log("yo! pressed bookmark")}>
        <Ionicons name='bookmark-outline' size={40} color='#000000' />
      </Pressable>
    </View>
  </SafeAreaView>
);

const styles = StyleSheet.create({
  general: {
    alignItems: "center",
  },
  headerText: {
    fontSize: 16,
    fontFamily: "Poppins_400Regular",
  },
  stats: {
    fontFamily: "Poppins_700Bold",
    fontSize: 25,
  },
});
```

LA GESTION DES ETATS - REACT NATIVE

LE PROJET

IMPLÉMENTATION DE REDUX DANS NOTRE APPLICATION : COMBINAISON DE REDUCERS - TOOLKIT REDUX : IMAGEDETAILSMODAL





MERCI POUR L'ÉCOUTE
&
LA PARTICIPATION