Développement Mobile

Native, Hybride et Cross-platform

Réalisé par Yassine SAIDANE
Enseigné par Mr. Malek BEN SALEM

Application Native

Une application native est une application mobile qui est développée spécifiquement pour un des systèmes d'exploitation utilisé par les smartphones et tablettes (**iOS**, **Android**, ...) utilisant des langages comme **Objective-C**, **Swift**, **Java** et **Kotlin**.

Développement Native

Avantages	Inconvénients
• Performance • Sécurité • Interactivité et Intuitivité • Accès plus profond • Stabilité	• Coûteux • Chronophage

Si on cible plusieurs plateformes en même temps:

- Comment peut-on diminuer les coûts?
- Comment peut-on diminuer le temps de développement?

Application Hybride

Une application hybride est une application mobile qui nous donne quelques fonctionnalités d’une application native et la simplicité et la rapidité de développement d’une application web.
Elle peut être distribuée en tant qu'application sur les plateformes **App Store**, **Play Store** …

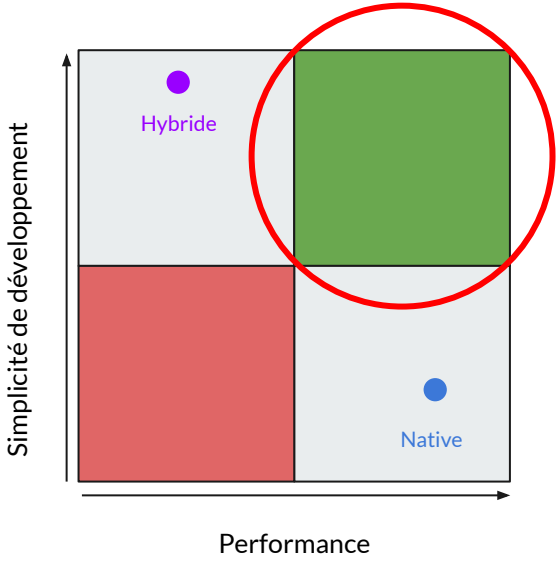
Développement Hybride

Une application hybride peut être développée en utilisant **HTML**, **CSS** et **JavaScript.** Il y a des frameworks comme **Cordova, Titanium.** Ou, il y a une autre technique comme **PWA.**

Développement Hybride

Avantages	Inconvénients
• Abordable • Rapide • Mise à jour instantané	• Basse performance • Accès limité à certaines fonctionnalitées

Comment peut-on avoir les avantages du développement native et hybride en même temps?



Application Cross-platform

Une application cross-platform est une application développée utilisant des frameworks comme **Flutter**, **React Native** ou **Xamarin** qui sont basés sur des langages de programmation comme **Dart**, **JavaScript** ou **C#** et s'exécute nativement sur plusieurs plateformes comme **iOS**, **Android** et même le **Web**.

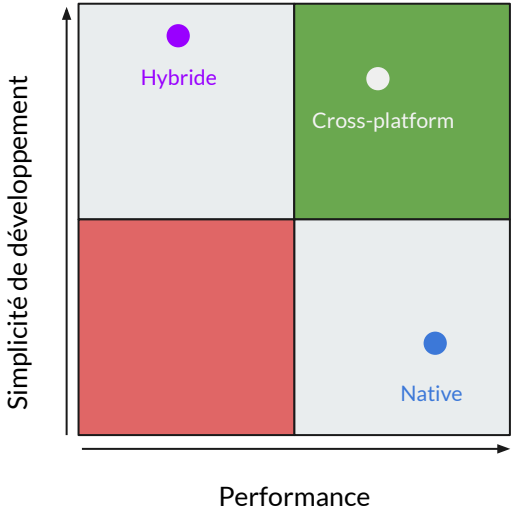
Application Cross-Platform

L’opération de construction d’une application cross-platform consiste à deux étapes : exporter, compiler.

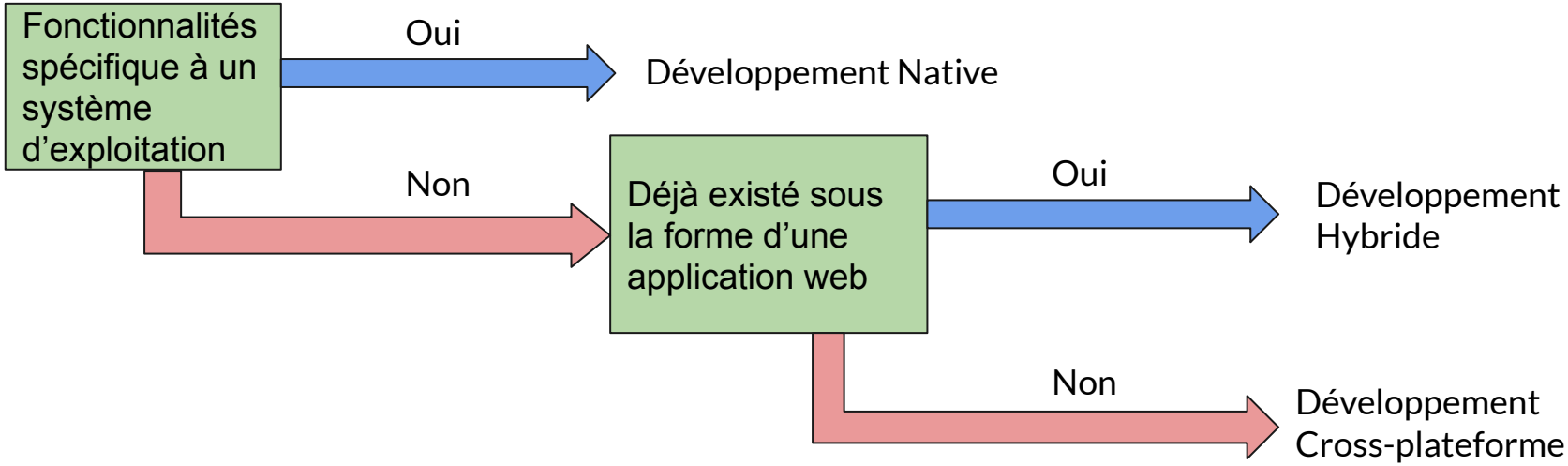
Développement Cross-platform

Avantages	Inconvénients
• Performance • Accès plus profond • Abordable • Rapide • Mise à jour instantané	• Couches intermédiaires • Moins stable

Comment choisir la meilleure technologie pour mon projet?



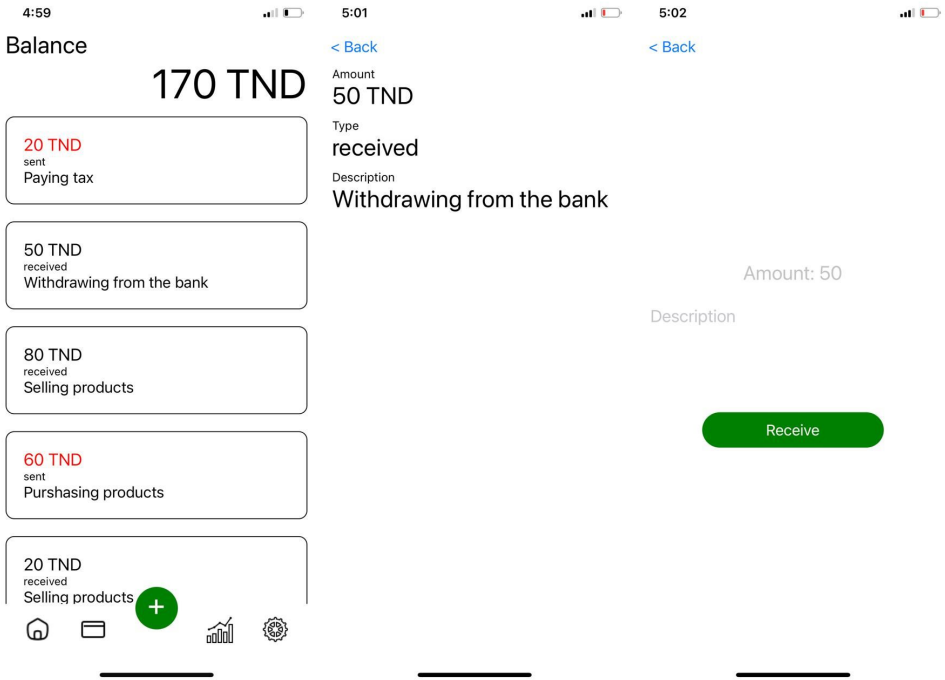
Arbre de décision pour choisir la technologie



On pratique!

Qu’est ce que on va faire?

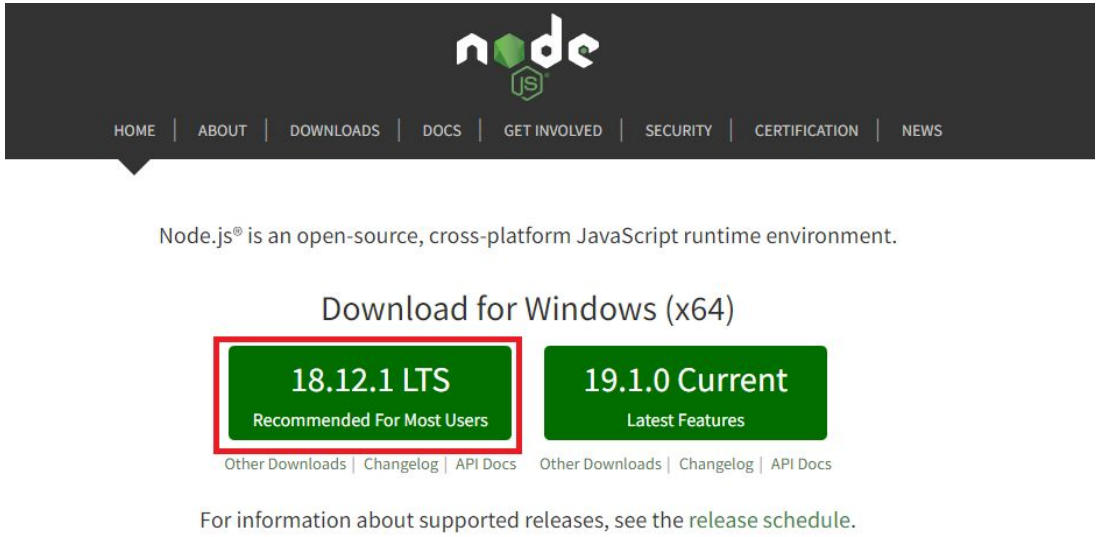
Une application pour gérer notre argent.

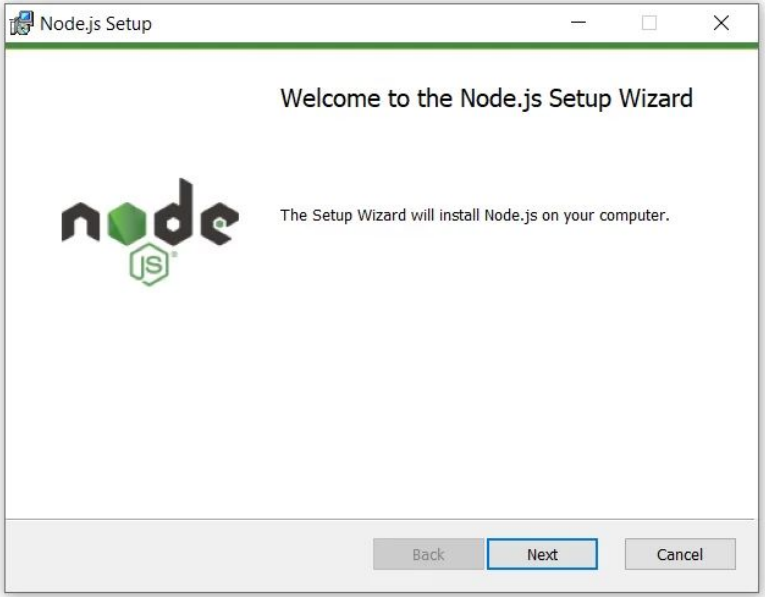


Comment peut-on commencer?

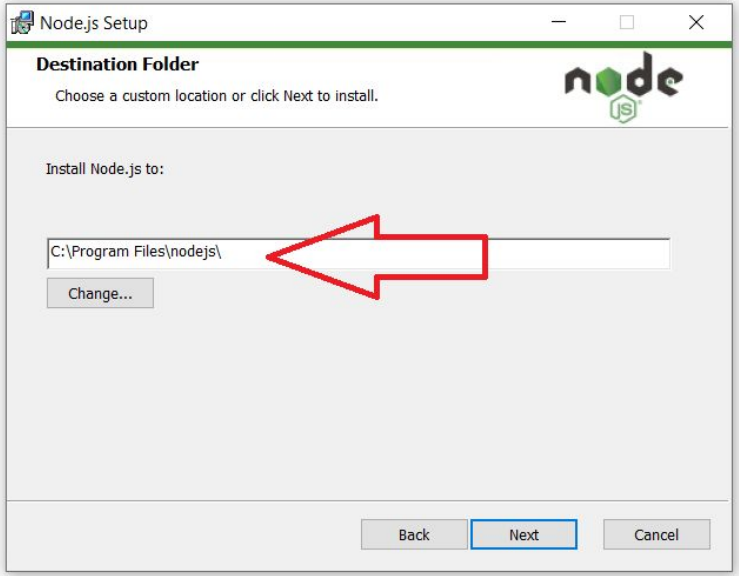
- Éditeur de texte (VS Code, notepad++, ...)
- NodeJS
- NPM (généralement inclus avec NodeJS)
- Expo

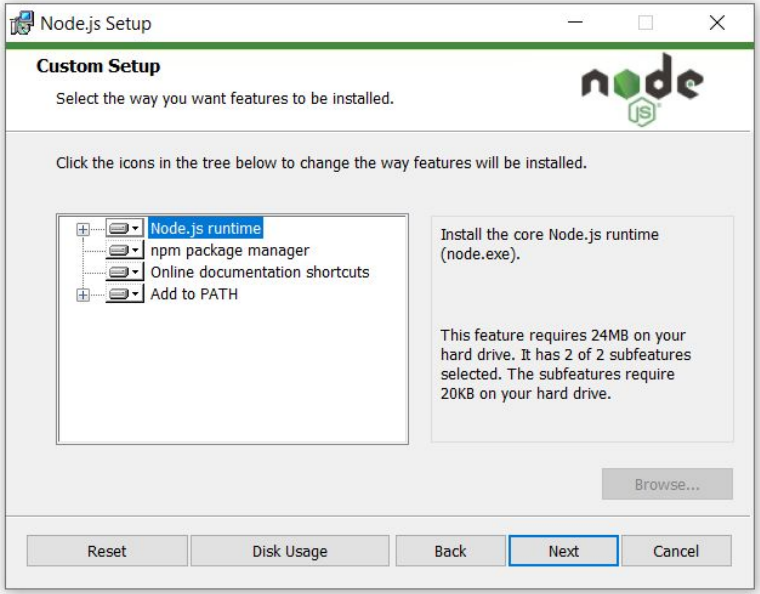
n.b. Il est fortement recommandé d'utiliser une version NodeJS LTS, et un nombre de version pair avec expo

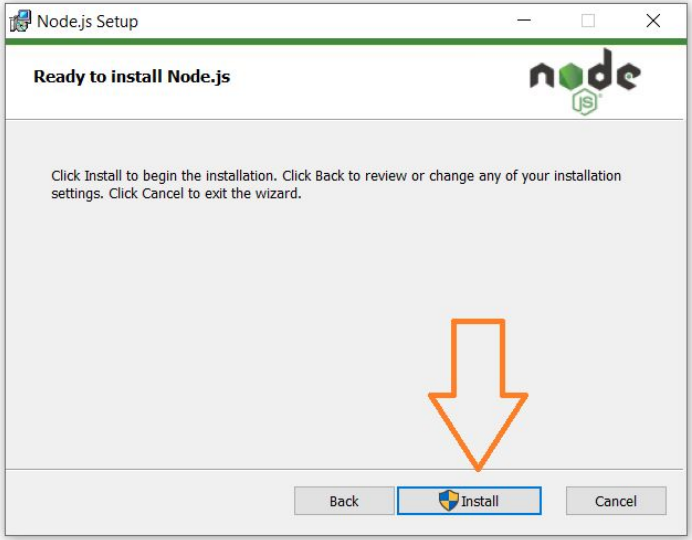

node
HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS
Node.js® is an open-source, cross-platform JavaScript runtime environment.
Download for Windows (x64)
18.12.1 LTS
Recommended For Most Users
19.1.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs
For information about supported releases, see the release schedule.

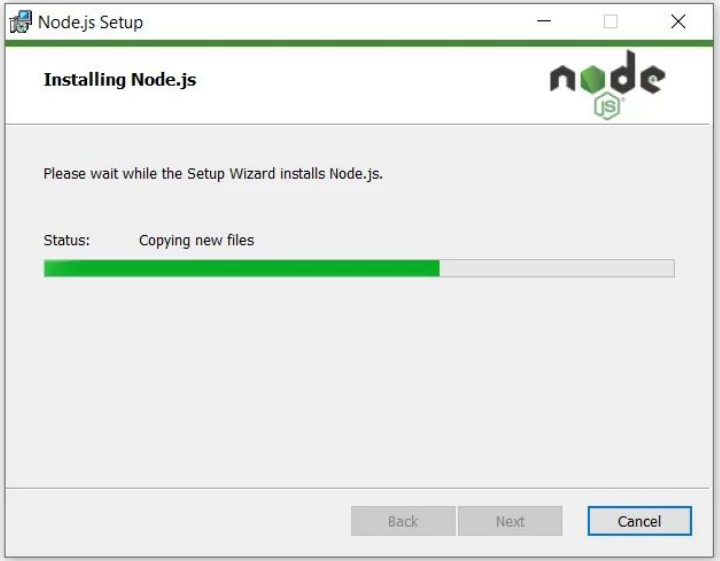

Node.js Setup
Welcome to the Node.js Setup Wizard
The Setup Wizard will install Node.js on your computer.
Back
Next
Cancel

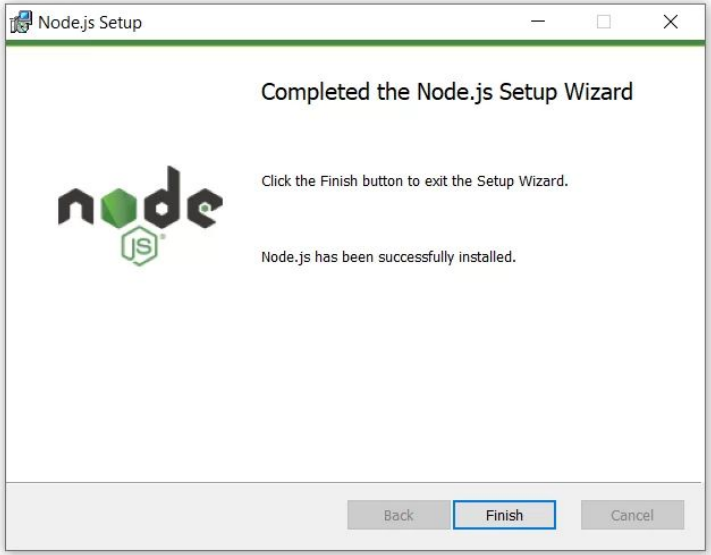

Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
I accept the terms in the License Agreement
Print
Back
Next
Cancel

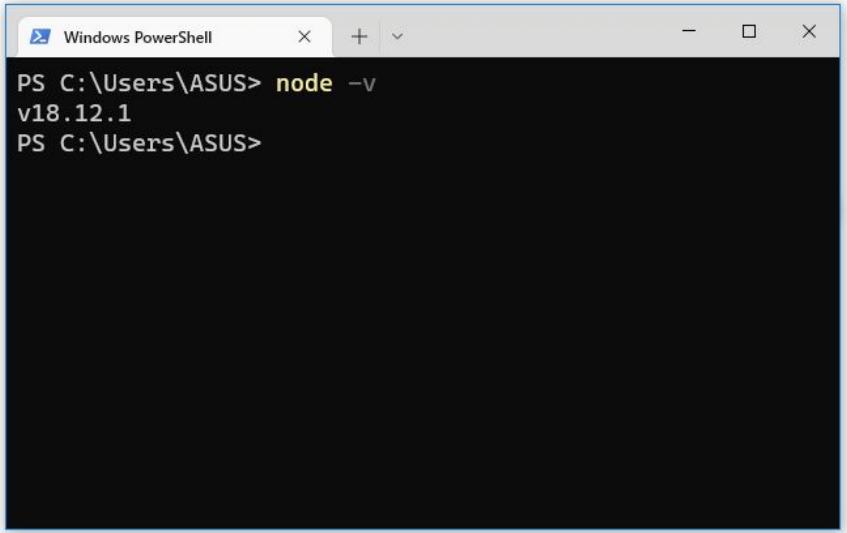

Node.js Setup
Destination Folder
Choose a custom location or click Next to install.
Install Node.js to:
C:\Program Files\nodejs\
Change...
Back
Next
Cancel


Node.js Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
Node.js runtime
npm package manager
Online documentation shortcuts
Add to PATH
Install the core Node.js runtime (node.exe).
This feature requires 24MB on your hard drive. It has 2 of 2 subfeatures selected. The subfeatures require 20KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel


Node.js Setup
Ready to install Node.js
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


Node.js Setup
Installing Node.js
Please wait while the Setup Wizard installs Node.js.
Status:
Copying new files
Back
Next
Cancel


Node.js Setup
Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.
Back
Finish
Cancel


Windows PowerShell
PS C:\Users\ASUS> node -v
v18.12.1
PS C:\Users\ASUS>

Pourquoi Expo?

Pour bénéficier:

- De l’application de test Expo Go.
- Des librairies préparés comme expo-camera, expo-sensors …
- De système de construction automatique des applications pour Android et iOS sur les serveurs.

Expo CLI

On initialise notre projet avec la commande suivante:

npx create-expo-app mybank --template

Utilisant React Native CLI:

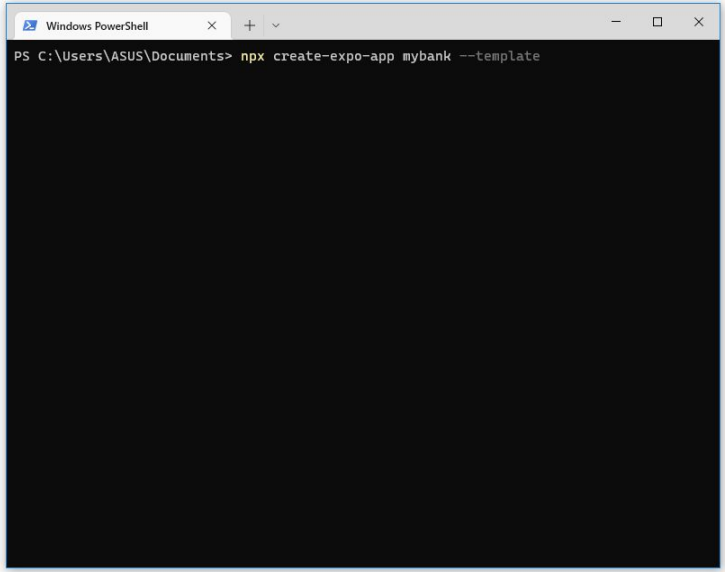
npx react-native init mybank

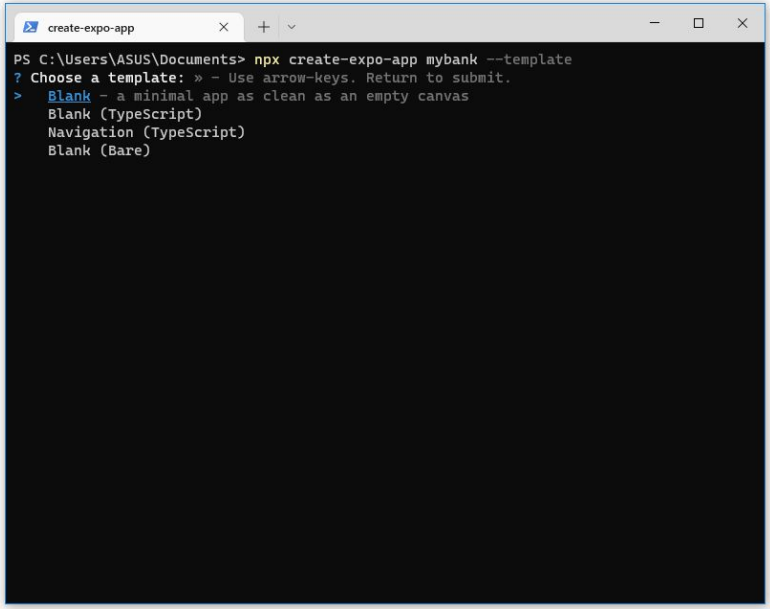
Expo CLI

Expo nous permet de choisir parmi 2 types de workflow:

- Bare : le dossier React Native exposé
- Managed : le dossier React Native caché

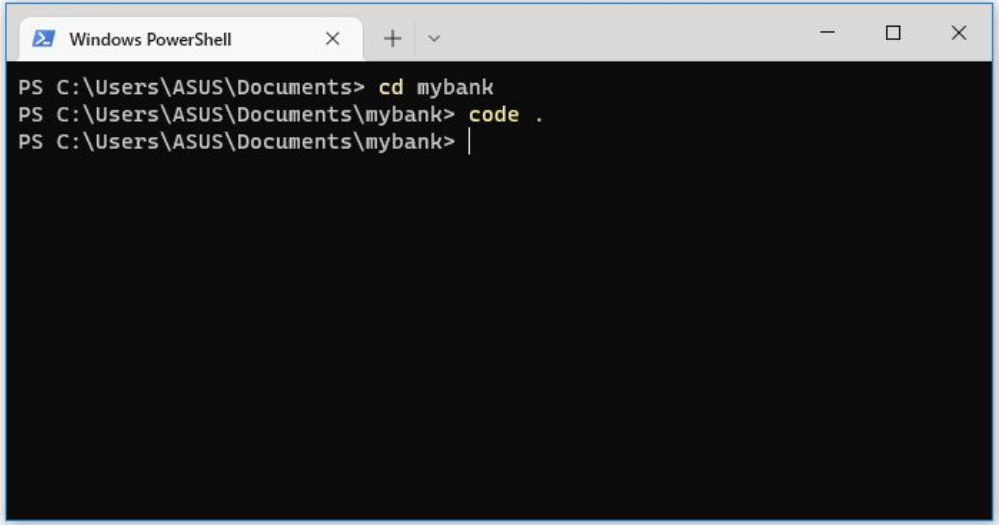
Initialisation de l’environnement de travail

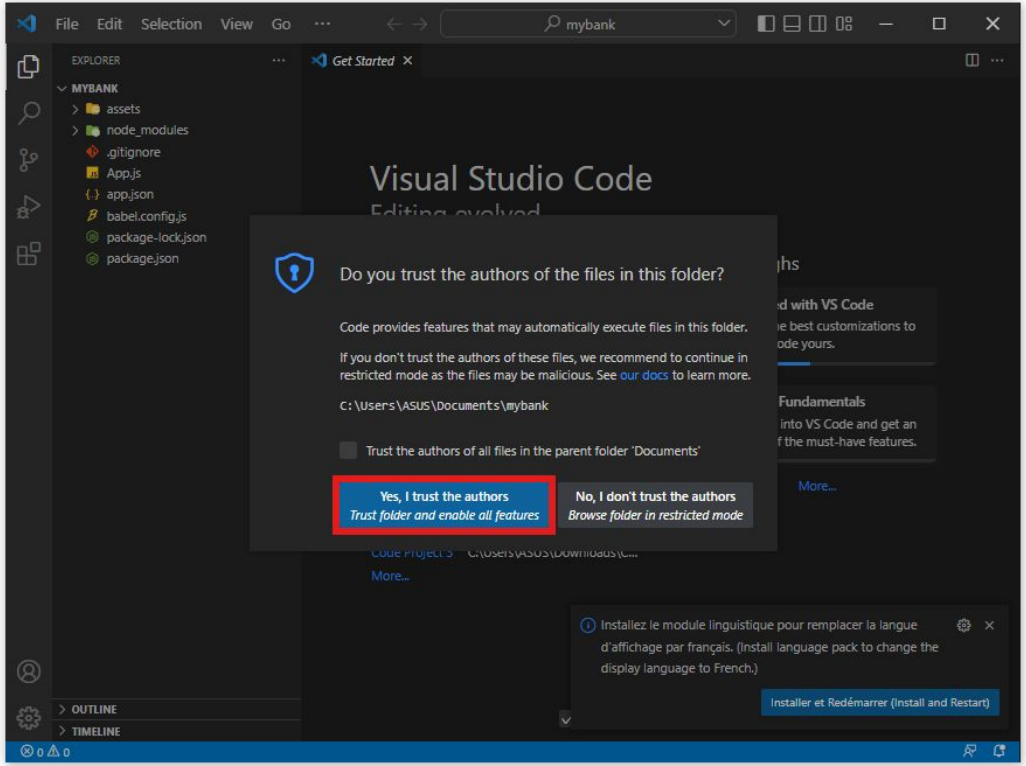

Windows PowerShell
PS C:\Users\ASUS\Documents> npx create-expo-app mybank --template

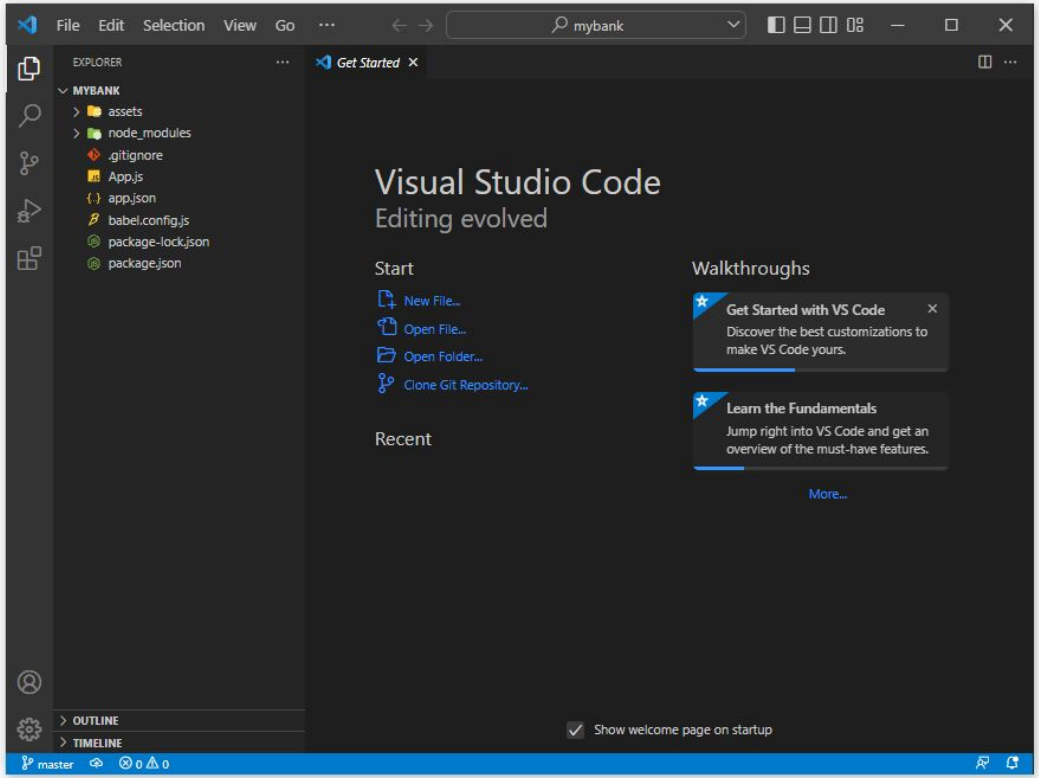

create-expo-app
PS C:\Users\ASUS\Documents> npx create-expo-app mybank --template
? Choose a template: » - Use arrow-keys. Return to submit.
> Blank - a minimal app as clean as an empty canvas
Blank (TypeScript)
Navigation (TypeScript)
Blank (Bare)

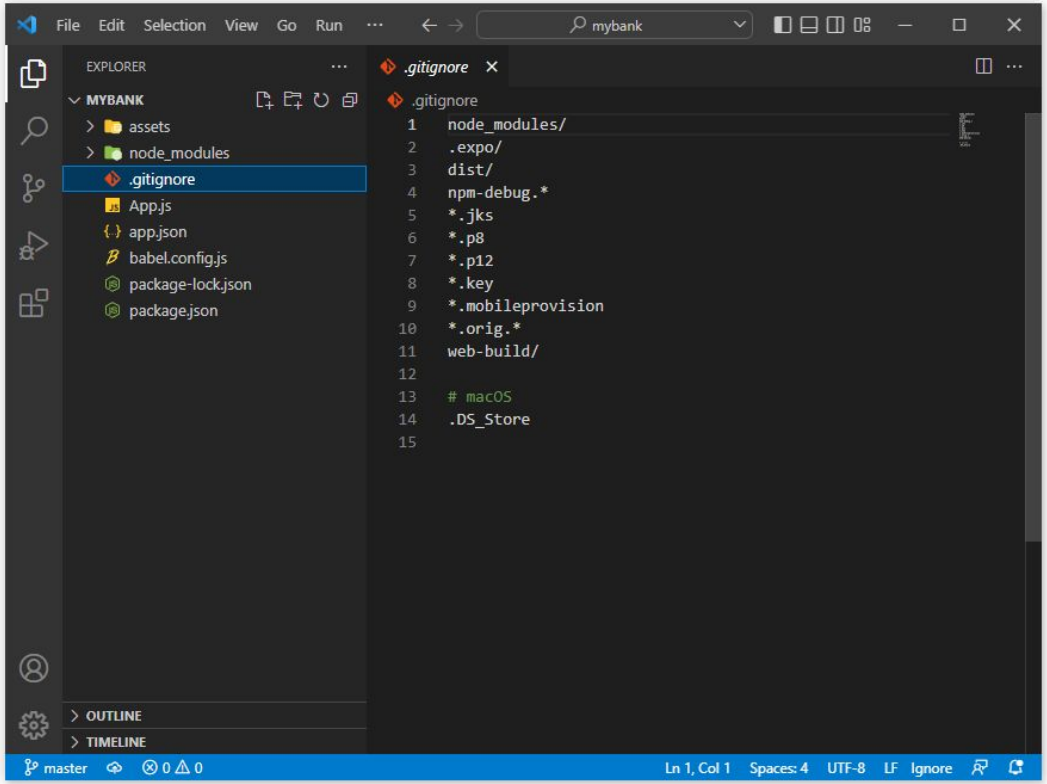
Windows PowerShell

```
PS C:\Users\ASUS\Documents> npx create-expo-app mybank --template
√ Choose a template: » Blank
√ Downloaded and extracted project files.
√ Installed JavaScript dependencies.

✅ Your project is ready!

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

- cd mybank
- 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 with
out a Mac
- npm run web
PS C:\Users\ASUS\Documents>
```


Windows PowerShell
PS C:\Users\ASUS\Documents> cd mybank
PS C:\Users\ASUS\Documents\mybank> code .
PS C:\Users\ASUS\Documents\mybank>


File Edit Selection View Go
mybank
EXPLORER
MYBANK
assets
node_modules
.gitignore
App.js
app.json
babel.config.js
package-lock.json
package.json
Get Started
Visual Studio Code
Do you trust the authors of the files in this folder?
Code provides features that may automatically execute files in this folder.
If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.
C:\Users\ASUS\Documents\mybank
Trust the authors of all files in the parent folder 'Documents'
Yes, I trust the authors
Trust folder and enable all features
No, I don't trust the authors
Browse folder in restricted mode
Fundamentals
More...
Installez le module linguistique pour remplacer la langue d'affichage par français. (Install language pack to change the display language to French.)
Installer et Redémarrer (Install and Restart)
OUTLINE
TIMELINE


File
Edit
Selection
View
Go
mybank
EXPLORER
MYBANK
assets
node_modules
.gitignore
App.js
app.json
babel.config.js
package-lock.json
package.json
OUTLINE
TIMELINE
Get Started
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Recent
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
More...
Show welcome page on startup
master


File Edit Selection View Go Run
mybank
EXPLORER
MYBANK
assets
node_modules
.gitignore
App.js
app.json
babel.config.js
package-lock.json
package.json
OUTLINE
TIMELINE
.gitignore
node_modules/
.expo/
dist/
npm-debug.*
*.jks
*.p8
*.p12
*.key
*.mobileprovision
.orig.
web-build/
macOS
.DS_Store
master
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Ignore

File Edit Selection View Go Run ... mybank

EXPLORER

MYBANK

- assets
- node_modules
- .gitignore
- App.js
- app.json
- babel.config.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

App.js > ...

```
1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3
4  export default function App() {
5    return (
6      <View style={styles.container}>
7        <Text>Open up App.js to start working on your app
8        <StatusBar style="auto" />
9      </View>
10   );
11 }
12
13 const styles = StyleSheet.create({
14   container: {
15     flex: 1,
16     backgroundColor: '#fff',
17     alignItems: 'center',
18     justifyContent: 'center',
19   },
20 });
21
```

master ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

File Edit Selection View Go Run ··· mybank

EXPLORER

MYBANK

- assets
- node_modules
- .gitignore
- App.js
- app.json
- babel.config.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

app.json > ...

```
1  {
2    "expo": {
3      "name": "mybank",
4      "slug": "mybank",
5      "version": "1.0.0",
6      "orientation": "portrait",
7      "icon": "./assets/icon.png",
8      "userInterfaceStyle": "light",
9      "splash": {
10       "image": "./assets/splash.png",
11       "resizeMode": "contain",
12       "backgroundColor": "#ffffff"
13     },
14     "updates": {
15       "fallbackToCacheTimeout": 0
16     },
17     "assetBundlePatterns": [
18       "**/*"
19     ],
20     "ios": {
21       "supportsTablet": true
22     },
23     "android": {
24       "adaptiveIcon": {
25         "foregroundImage": "./assets/adaptive-icon.png"
26         "backgroundColor": "#FFFFFF"
27       }
28     },
```

master ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JSON

File Edit Selection View Go Run ··· mybank

EXPLORER

MYBANK

- assets
- node_modules
- .gitignore
- App.js
- app.json
- babel.config.js
- package-lock.json
- package.json

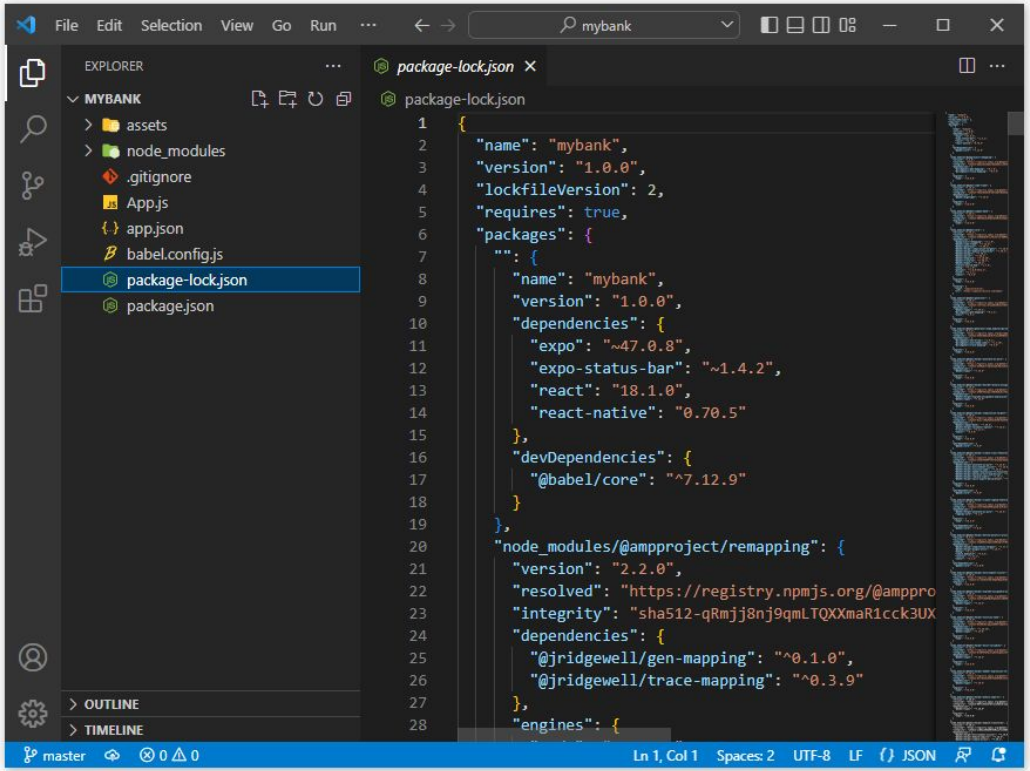
OUTLINE

TIMELINE

babel.config.js > <unknown>

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

master ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript


File Edit Selection View Go Run
mybank
EXPLORER
MYBANK
assets
node_modules
.gitignore
App.js
app.json
babel.config.js
package-lock.json
package.json
OUTLINE
TIMELINE
package-lock.json
{
"name": "mybank",
"version": "1.0.0",
"lockfileVersion": 2,
"requires": true,
"packages": {
"": {
"name": "mybank",
"version": "1.0.0",
"dependencies": {
"expo": "~47.0.8",
"expo-status-bar": "~1.4.2",
"react": "18.1.0",
"react-native": "0.70.5"
},
"devDependencies": {
"@babel/core": "^7.12.9"
}
},
"node_modules/@ampproject/remapping": {
"version": "2.2.0",
"resolved": "https://registry.npmjs.org/@ampproj
"integrity": "sha512-qRmjj8nj9qmLTQXXmaR1cck3UX
"dependencies": {
"@jridgewell/gen-mapping": "^0.1.0",
"@jridgewell/trace-mapping": "^0.3.9"
},
"engines": {
master
Ln 1, Col 1
Spaces: 2
UTF-8
LF
JSON

File Edit Selection View Go Run ··· mybank

EXPLORER

MYBANK

- assets
- node_modules
- .gitignore
- App.js
- app.json
- babel.config.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

package.json ×

package.json > ...

```
1  {
2    "name": "mybank",
3    "version": "1.0.0",
4    "main": "node_modules/expo/AppEntry.js",
       ▷ Debug
5    "scripts": {
6      "start": "expo start",
7      "android": "expo start --android",
8      "ios": "expo start --ios",
9      "web": "expo start --web"
10   },
11   "dependencies": {
12     "expo": "~47.0.8",
13     "expo-status-bar": "~1.4.2",
14     "react": "18.1.0",
15     "react-native": "0.70.5"
16   },
17   "devDependencies": {
18     "@babel/core": "^7.12.9"
19   },
20   "private": true
21 }
22
```

master ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JSON

File Edit Selection View Go Run ... mybank

EXPLORER

MYBANK

- assets
- node_modules
- .gitignore
- App.js
- app.json
- babel.config.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

App.js > ...

```
1  import { StatusBar } from 'expo-status-bar';
2  import { StyleSheet, Text, View } from 'react-native';
3
4  export default function App() {
5    return (
6      <View style={styles.container}>
7        <Text>Open up App.js to start working on your app
8        <StatusBar style="auto" />
9      </View>
10   );
11 }
12
13 const styles = StyleSheet.create({
14   container: {
15     flex: 1,
16     backgroundColor: '#fff',
17     alignItems: 'center',
18     justifyContent: 'center',
19   },
20 });
21
```

master ⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

Expo CLI

Nous pouvons tester notre application:

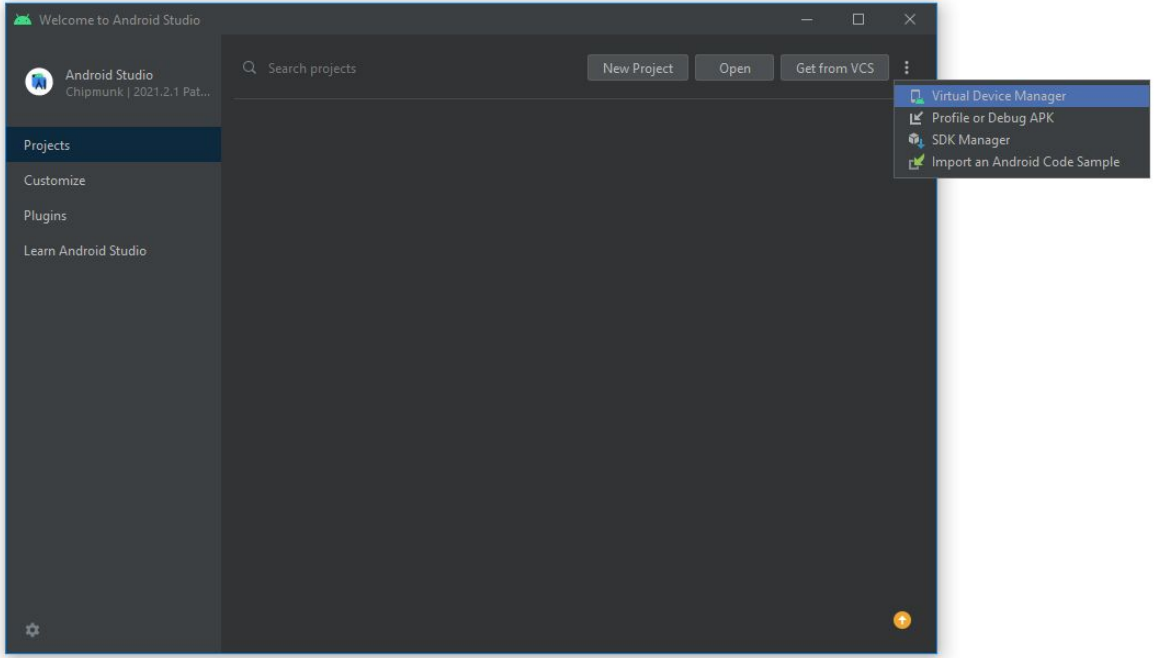
npm run android
npm run ios
npm run web

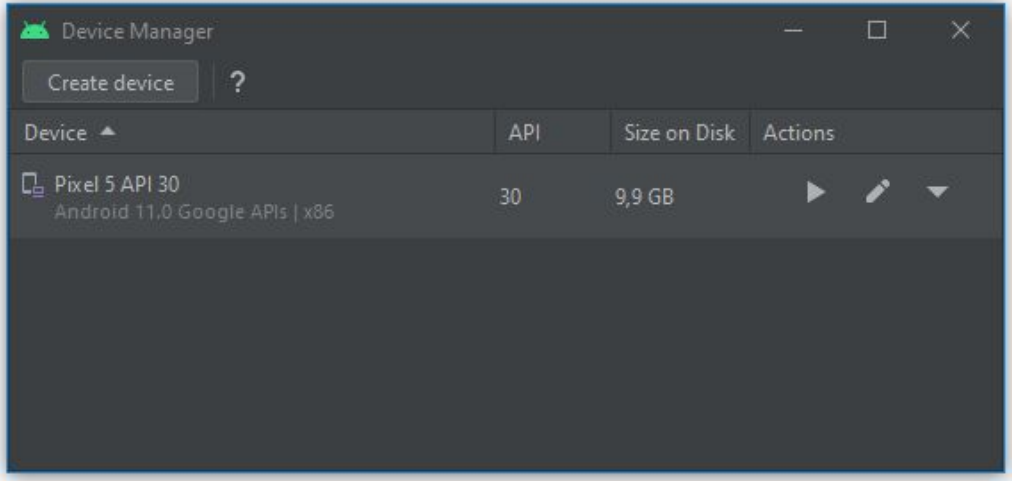
Utilisant React Native CLI:

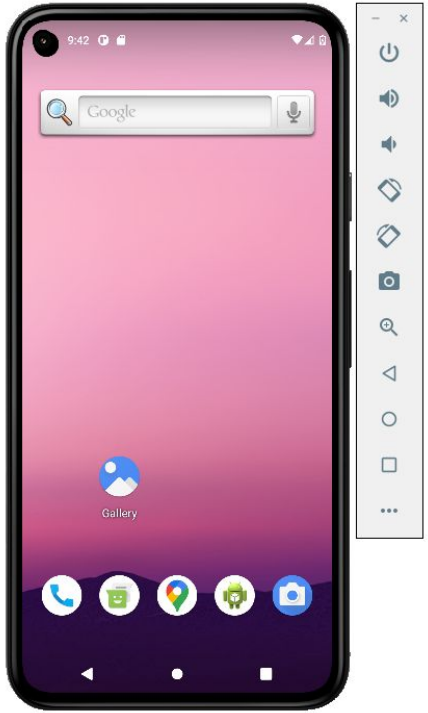
npx react-native run-android

n.b. une machine macOS est obligatoire pour exécuter l'application avec un simulateur iOS

React Native sur Emulateur Android

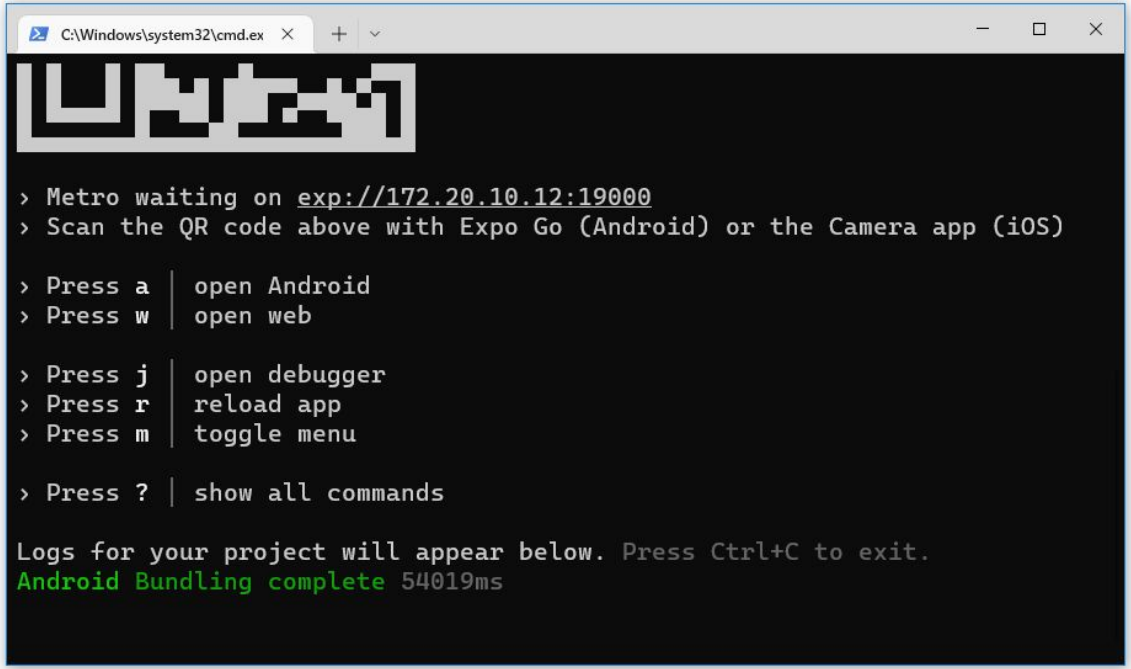

Welcome to Android Studio
Android Studio
Chipmunk | 2021.2.1 Pat...
Search projects
New Project
Open
Get from VCS
Virtual Device Manager
Profile or Debug APK
SDK Manager
Import an Android Code Sample
Projects
Customize
Plugins
Learn Android Studio

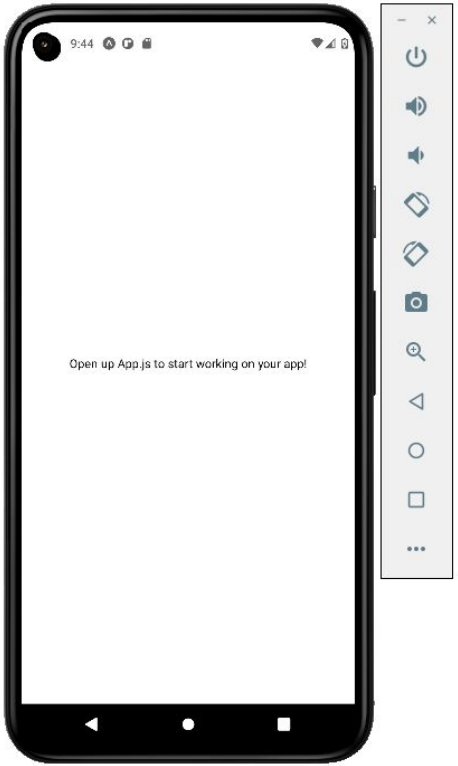

Device Manager
Create device
?
Device
API
Size on Disk
Actions
Pixel 5 API 30
Android 11.0 Google APIs | x86
30
9,9 GB


9:42
Google
Gallery

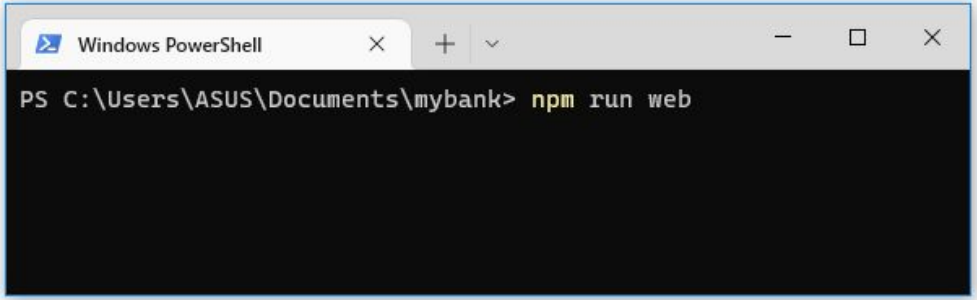
Windows PowerShell

```
PS C:\Users\ASUS\Documents\mybank> npm run android
```


C:\Windows\system32\cmd.ex
› Metro waiting on exp://172.20.10.12:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
› Press a | open Android
› Press w | open web
› Press j | open debugger
› Press r | reload app
› Press m | toggle menu
› Press ? | show all commands
Logs for your project will appear below. Press Ctrl+C to exit.
Android Bundling complete 54019ms


9:44
Open up App.js to start working on your app!

React native sur WEB


Windows PowerShell
PS C:\Users\ASUS\Documents\mybank> npm run web

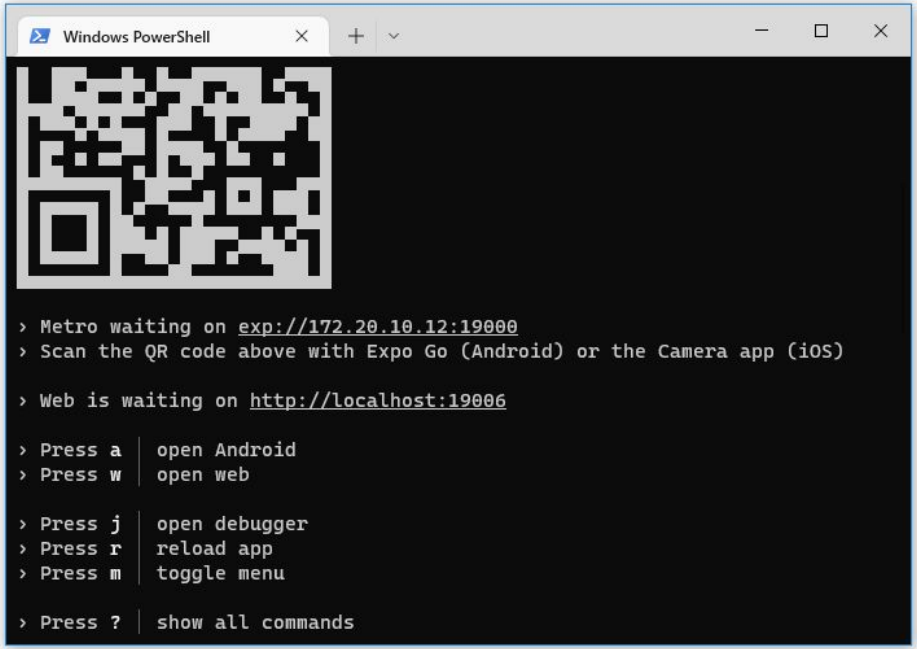
```
Windows PowerShell

PS C:\Users\ASUS\Documents\mybank> npm run web

> mybank@1.0.0 web
> expo start --web

Starting project at C:\Users\ASUS\Documents\mybank
CommandError: It looks like you're trying to use web support but don't have the required dependencies
installed.

Please install react-native-web@~0.18.9, react-dom@18.1.0, @expo/webpack-config@^0.17.2 by running:

npx expo install react-native-web@~0.18.9 react-dom@18.1.0 @expo/webpack-config@^0.17.2

If you're not using web, please ensure you remove the "web" string from the platforms array in the pro
ject
Expo config.

PS C:\Users\ASUS\Documents\mybank>
```

Windows PowerShell

```
PS C:\Users\ASUS\Documents\mybank> npx expo install react-native-web@~0.18.9 react-dom@18.1.0 @expo/webpack-config@^0.17.2
```

Windows PowerShell

```
 fewer dependencies
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x
 fewer dependencies
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams AP
I instead.
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 690 packages, and audited 1811 packages in 47s

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

23 vulnerabilities (2 moderate, 17 high, 4 critical)

To address issues that do not require attention, run:
  npm audit fix

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.

PS C:\Users\ASUS\Documents\mybank>
```


Windows PowerShell
› Metro waiting on exp://172.20.10.12:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
› Web is waiting on http://localhost:19006
› Press a | open Android
› Press w | open web
› Press j | open debugger
› Press r | reload app
› Press m | toggle menu
› Press ? | show all commands

Expo CLI

Nous pouvons compiler notre application :

npx expo export:web

npx expo run:ios --configuration Release

npx expo run:android --variant release

ou

npx react-native run-android --variant=release

Astuce finale

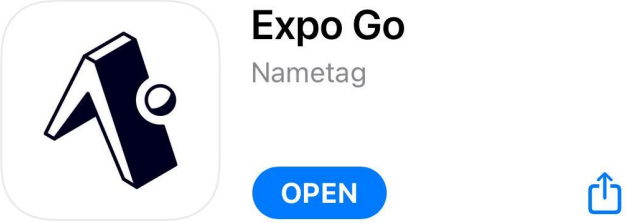
Utiliser les smartphones pour tester l'application

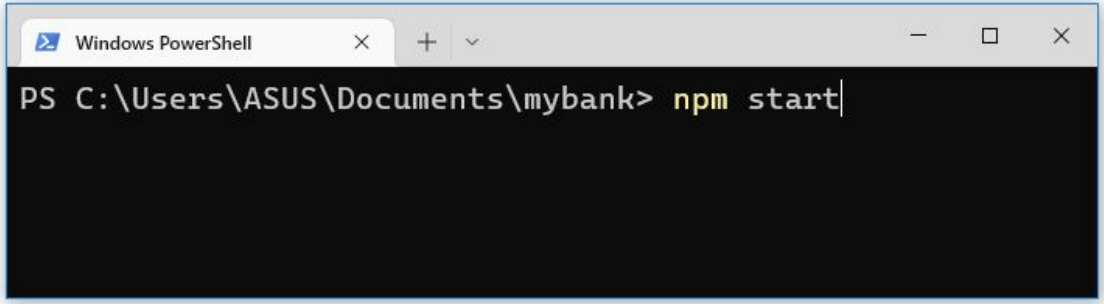
- Installer Expo Go
- taper *npm start*
- scanner le code QR avec le smartphone
- Happy hacking!

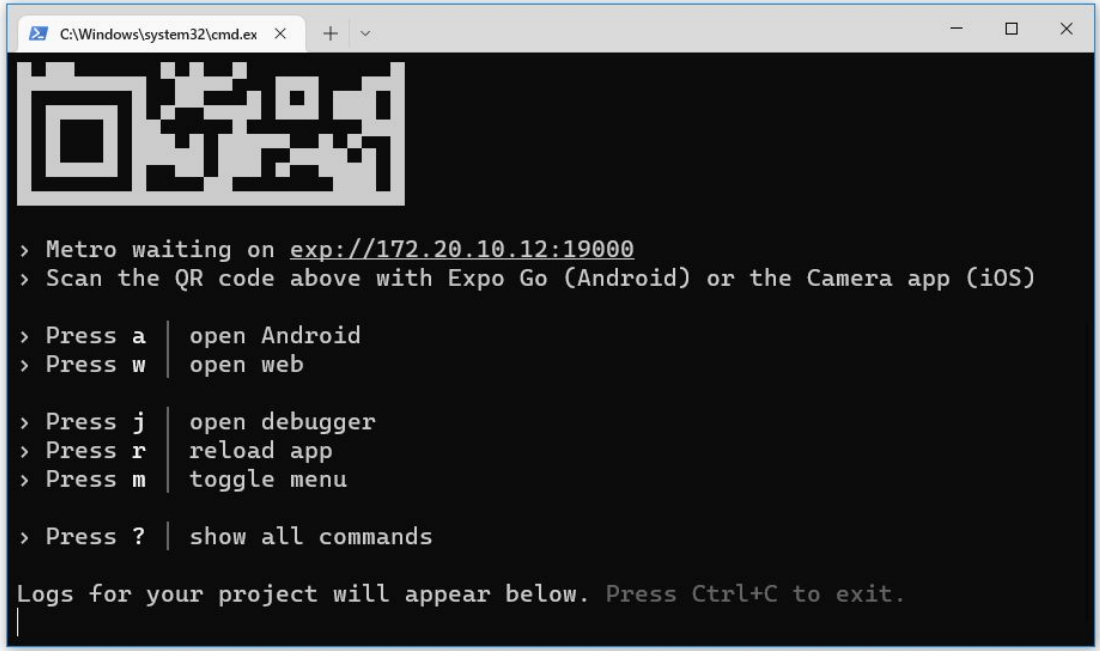
n.b. Il faut bien vérifier que la machine de développement et l'appareil mobile sont connectés sur le même réseau.

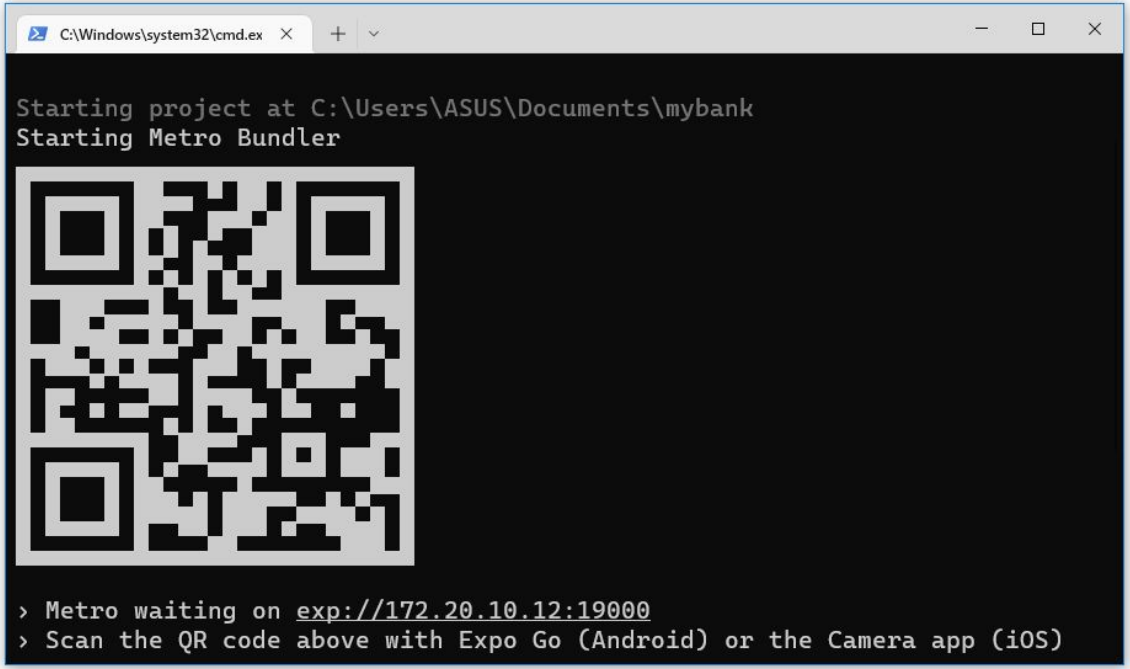
React Native avec Expo Go


Expo
Expo Project
3,2★
8,03 k avis
1 M+
Téléchargements
3+
3 ans et plus
Installer
Ajouter à la liste de souhaits


Expo Go
Nametag
OPEN


Windows PowerShell
PS C:\Users\ASUS\Documents\mybank> npm start


C:\Windows\system32\cmd.ex
› Metro waiting on exp://172.20.10.12:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
› Press a | open Android
› Press w | open web
› Press j | open debugger
› Press r | reload app
› Press m | toggle menu
› Press ? | show all commands
Logs for your project will appear below. Press Ctrl+C to exit.


C:\Windows\system32\cmd.ex
Starting project at C:\Users\ASUS\Documents\mybank
Starting Metro Bundler
› Metro waiting on exp://172.20.10.12:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

Composantes React Native

React Native

View: conteneur qui contient d'autre éléments React Native (eq **div** en HTML, View en Android)
librairie: react-native
import:

```
import { View } from 'react-native';
```

syntax:

```
<View>
    { composantes react native }
</View>
```

React Native

Text: affiche un texte (eq **div** en HTML, TextView en Android)
librairie: react-native
import:

```
import { Text } from 'react-native';
```

syntax:

```
<Text>{text}</Text>
```

React Native

TextInput: champ de saisie (eq **input** en HTML, EditText en Android)
librairie: react-native
import:

```
import { TextInput } from ‘react-native’;
```

syntax:

```
<TextInput value={value} onChangeText={function} />
```

React Native

Button: bouton (eq **button** en HTML, Button en Android)
librairie: react-native
import:

```
import { Button } from ‘react-native’;
```

syntax:

```
<Button title=”Click me” onPress={function}/>
```

React Native

TouchableHighlight : conteneur cliquable avec un effet spécial
librairie: react-native
import:

```
import { TouchableHighlight } from ‘react-native’;
```

syntax:

```
<TouchableHighlight onPress={function}>
{ composantes react native }
</TouchableHighlight>
```

React Native

TouchableOpacity : conteneur cliquable avec un effet de transparence
librairie: react-native
import:

```
import { TouchableOpacity } from ‘react-native’;
```

syntax:

```
<TouchableOpacity  onPress={function}>
{ composantes react native }
</TouchableOpacity>
```

React Native

TouchableWihoutFeedback : conteneur cliquable sans effet
librairie: react-native
import:

```
import { TouchableWihoutFeedback } from ‘react-native’;
```

syntax:

```
<TouchableWihoutFeedback onPress={function}>
{ composantes react native }
</TouchableWihoutFeedback>
```

React Native

Image : afficher une image
librairie: react-native
import:

```
import { Image } from ‘react-native’;
```

syntax:

```
<Image source={source} />
```

source: require(“chemin/local”) ou {uri: “lien”}

React Native

ImageBackground : afficher une image de fond
librairie: react-native
import:

```
import { ImageBackground } from ‘react-native’;
```

syntax:

```
<ImageBackground source={source} />
```

source: require(“chemin/local”) ou {uri: “lien”}

React Native

Switch : une switch avec 2 positions (on/off)
librairie: react-native
import:

```
import { Switch } from ‘react-native’;
```

syntax:

```
<Switch value={value} onValueChange={function} />
```

React Native

FlatList: conteneur qui contient d'autre éléments React Native (eq **button** en HTML, ListView en Android)
librairie: react-native
import:

```
import { FlatList } from 'react-native';
```

syntax:

```
<FlatList data={array} renderItem={itemComponent}/>
```

React Native

StatusBar : la barre des status (eq AppCompatActivity en Android)
librairie: react-native
import:

```
import { StatusBar } from ‘react-native’;
```

syntax:

```
<StatusBar />
```

React Native

SafeAreaView: conteneur pour éviter le notch de l'iPhone
librairie: react-native
import:

```
import { SafeAreaView } from 'react-native';
```

syntax:

```
<SafeAreaView>
{composantes react native}
</SafeAreaView>
```

React Native

StyleSheet: créer et valider le syntax de style
librairie: react-native
import:

```
import { StyleSheet } from ‘react-native’;
```

syntax:

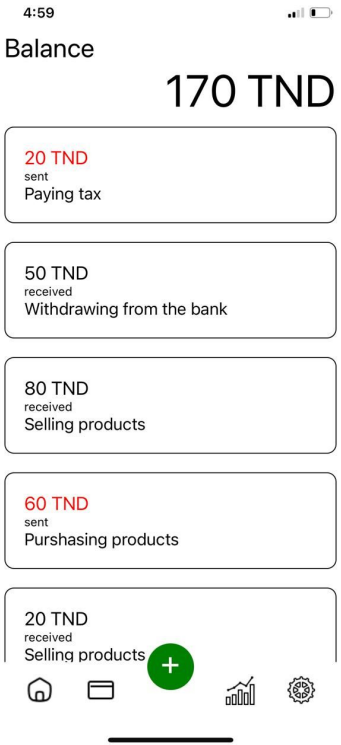
```
const styles = StyleSheet.create(obj);
```

obj: style object

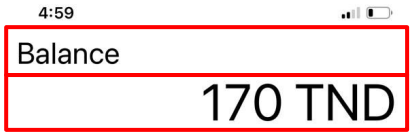
Appliquer le style:

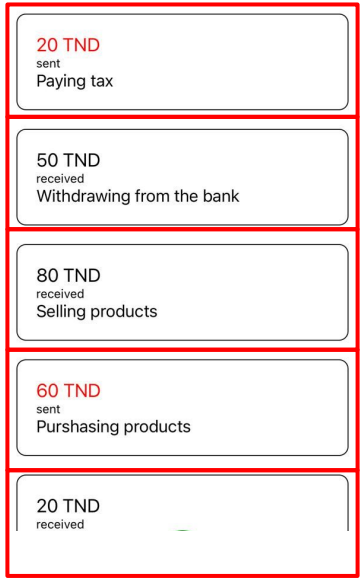
```
<View style={styles.container}></View>
```

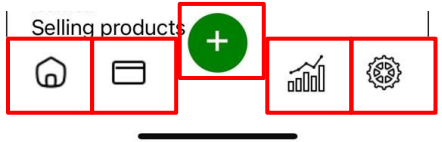
MyBank!


4:59
Balance
170 TND
20 TND
sent
Paying tax
50 TND
received
Withdrawing from the bank
80 TND
received
Selling products
60 TND
sent
Purshasing products
20 TND
received
Selling products

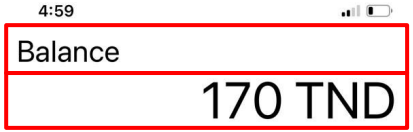

4:59
Header
Balance
170 TND
20 TND
sent
Paying tax
50 TND
received
Withdrawing from the bank
Liste
80 TND
received
Selling products
60 TND
sent
Purshasing products
20 TND
received
Selling products
Footer


4:59
Balance
170 TND


20 TND
sent
Paying tax
50 TND
received
Withdrawing from the bank
80 TND
received
Selling products
60 TND
sent
Purshasing products
20 TND
received


Selling products

View

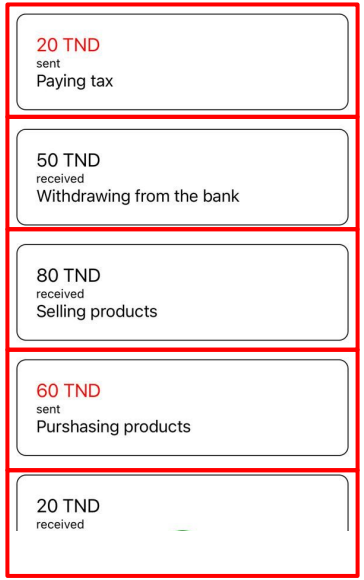


Text

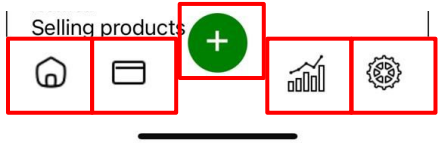
Text

Component

FlatList



View



Button

Button

Button

Button

Button

Header

./screens/HomeScreen.jsx

```
export function HomeScreen() {
    return <View>
        <View>
            <Text>Balance</Text>
            <Text>100 TND</Text>
        </View>
    </View>
}
```

./App.js

```
export default function App() {
    return <View style={styles.container}>
        <HomeScreen />
    </View>
}
const styles = StyleSheet.create({
  container: {
    flex: 1, backgroundColor: '#fff', alignItems: 'center',
    justifyContent: 'center',
  },
});
```

./App.js

```
export default function App() {
    return <SafeAreaView style={styles.container}>
        <HomeScreen />
    </SafeAreaView>
}
const styles = StyleSheet.create({
  container: {
    flex: 1, backgroundColor: '#fff', alignItems: 'center',
    justifyContent: 'center',
    marginTop: StatusBar.currentHeight || 0,
  },
});
```

Un peu de style

```
// ./screens/HomeScreen.jsx
const styles = StyleSheet.create({
    container: {
        width: '100%', height: '100%', display: 'flex',
        flexDirection: 'column', alignItems: 'stretch',
        paddingHorizontal: 10
    },
    header: {
        width: '100%', display: "flex", flexDirection: "column",
        alignItems: "flex-start", justifyContent: "center",
        fontSize: 48,
```

Un peu de style

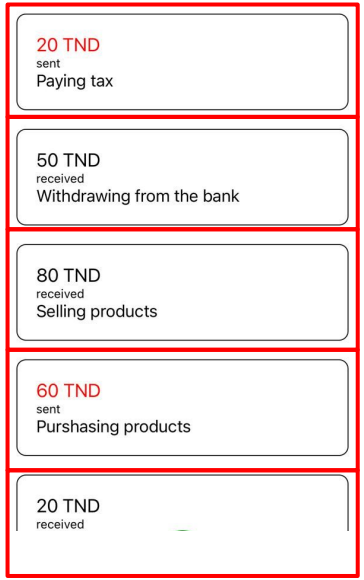
```
// ./screens/HomeScreen.jsx
const styles = StyleSheet.create({
    …
    title: {
        fontSize: 28,
    },
    balance: {
        alignSelf: "flex-end",
        fontSize: 48,
    }
}
```

Un peu de style

```
<View style={styles.container}>
    <View style={styles.header}>
        <Text style={styles.title}>Balance</Text>
        <Text style={styles.balance}>100 TND</Text>
    </View>
</View>
```

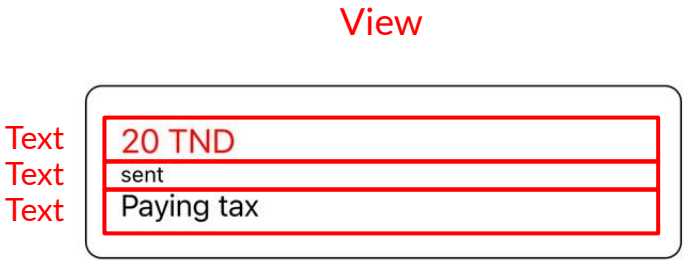
FlatList

Component



20 TND
sent
Paying tax


20 TND
sent
Paying tax


View
Text
Text
Text
20 TND
sent
Paying tax

```
// ./components/PaymentItem.jsx
export function PaymentItem(){
    return <View style={styles.container}>
        <Text style={styles.amount}>50 TND</Text>
        <Text style={styles.type}>Received</Text>
        <Text style={styles.description}>Pocket money</Text>
    </View>
}
const styles = StyleSheet.create({
    container: {
        width: '100%', display: 'flex', alignItems: 'flex-start',
        padding: 20, marginVertical: 10, borderWidth: 1,
```

```
// ./components/PaymentItem.jsx
const styles = StyleSheet.create({
    …
        borderRadius: 10,
    },
    amount: { fontSize: 20 },
    type: { fontSize: 13 },
    description: { fontSize: 18 }
});
```

```
// ./App.js
export default function App() {
    const [payments, setPayments] = useState([{amount: “50”,
        type: “received”, description: “Pocket money”},
        {amount: “20”, type: “sent”, description: “Grocery”}]);
    return <SafeAreaView>
        <HomeScreen />
    </SafeAreaView>
}
```

```
// ./screens/HomeScreen.jsx
export function HomeScreen({ payments }) {
    return <View>
        {/* Header */}
        <FlatList data={payments} renderItem={PaymentItem} />
    </View>
}
```

```
// ./App.js
export default function App() {
    const [payments, setPayments] = useState([{“50”, “received”,
        “Pocket money”}, {“20”, “sent”, “Grocery”]);
    return <SafeAreaView>
        <HomeScreen payments={payments} />
    </SafeAreaView>
}
```

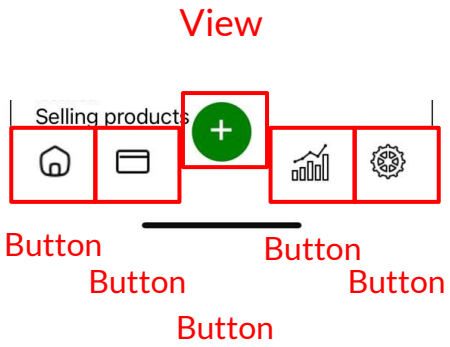
```
// ./components/PaymentItem.jsx
export function PaymentItem({ item }){
    return <View style={styles.container}>
        <Text style={styles.amount}>{item.amount} TND</Text>
        <Text style={styles.type}>{item.type}</Text>
        <Text style={styles.description}>{item.description}</Text>
    </View>
}
```

```
// ./components/PaymentItem.jsx
<Text style={[styles.type, {color: item.type === PAYMENT_TYPE_SEND ?
'red' : 'black'}]}>{item.type}</Text>
```

```
// ./constants.js
export const PAYMENT_TYPE_SEND = "sent";
export const PAYMENT_TYPE_RECEIVE = "received";
```

```
// ./App.js
const [payments, setPayments] = useState([{id: 0, amount: “50”,
    type: PAYMENT_TYPE_RECEIVE, description: “Pocket money”}, {id: 1,
    amount: “20”, type: PAYMENT_TYPE_SEND, description: “Grocery”}]);
```

```
// ./components/PaymentItem.jsx
export function PaymentItem({ item }){
    return <TouchableOpacity onPress={() => console.log("go to
        item" + item.id)}>
        …
    </TouchableOpacity>
}
```


View
Selling products
Button
Button
Button
Button
Button

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View>
    <Text>Home</Text>
    <Text>Pay</Text>
    <Text>+</Text>
    <Text>Stats</Text>
    <Text>Settings</Text>
</View>
```

```
// ./screens/HomeScreen.jsx
const styles = StyleSheet.create({
    …
    footer: {
        width: '100%',
        padding: 5,
        display: 'flex',
        flexDirection: 'row',
        alignItems: 'center',
        justifyContent: 'space-around',
    },
});
```

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    <Text>Home</Text>
    <Text>Pay</Text>
    <Text>+</Text>
    <Text>Stats</Text>
    <Text>Settings</Text>
</View>
```

TouchableOpacity

Image

TouchableOpacity

Text

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    <TouchableOpacity onPress={()=>console.log("go home")}>
        <Text>Home</Text>
    </TouchableOpacity>
    …
</View>
```

```
// ./screens/HomeScreen.jsx
const styles = StyleSheet.create({
    …
    logo: {
        width: 32,
        height: 32,
        color: 'black',
    },
});
```

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    <TouchableOpacity onPress={()=>console.log("go home")}>
        <Image style={styles.logo}
            source={require('../assets/home.png')} />
    </TouchableOpacity>
    …
</View>
```

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    …
    <TouchableOpacity onPress={()=>console.log("go pay")}>
        <Image style={styles.logo}
            source={require('../assets/carte-bancaire.png')} />
    </TouchableOpacity>
    …
</View>
```

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    …
    <TouchableOpacity onPress={()=>console.log("go receive")}>
        <Text>+</Text>
    </TouchableOpacity>
    …
</View>
```

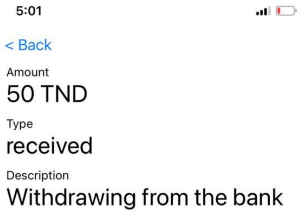
```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    …
    <TouchableOpacity onPress={()=>console.log("go stats")}>
        <Image style={styles.logo}
            source={require('../assets/chart.png')} />
    </TouchableOpacity>
    …
</View>
```

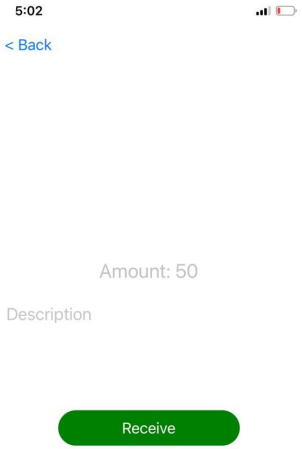
```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    …
    <TouchableOpacity onPress={()=>console.log("go settings")}>
        <Image style={styles.logo}
            source={require('../assets/settings.png')} />
    </TouchableOpacity>
</View>
```

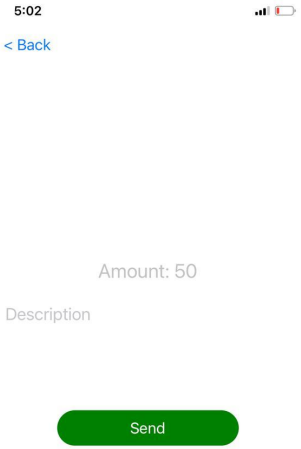
```
// ./screens/HomeScreen.jsx
const styles = StyleSheet.create({
    …
    receivePayment: {
        position: 'relative',
        top: -25,
        width: 50,
        height: 50,
        backgroundColor: 'green',
        borderRadius: 25,
    },
});
```

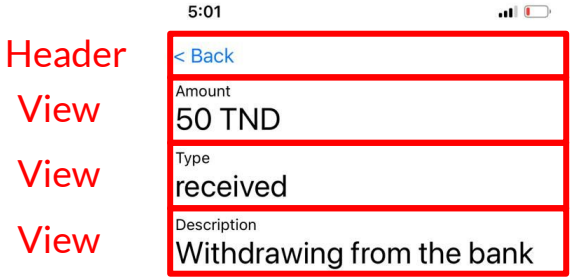
```
// ./screens/HomeScreen.jsx
const styles = StyleSheet.create({
    …
    plus: {
        height: '100%',
        width: '100%',
        textAlign: "center",
        textAlignVertical: 'center',
        fontSize: 35,
        color: 'white',
    },
});
```

```
// ./screens/HomeScreen.jsx
{/* Header */}
{/* FlatList */}
<View style={styles.footer}>
    …
    <TouchableOpacity onPress={()=>console.log("go receive")}
         style={styles.receivePayment}>
        <Text style={styles.plus}>+</Text>
    </TouchableOpacity>
    …
</View>
```


5:01
< Back
Amount
50 TND
Type
received
Description
Withdrawing from the bank


5:02
< Back
Amount: 50
Description
Receive


5:02
< Back
Amount: 50
Description
Send


5:01
Header
< Back
View
Amount
50 TND
View
Type
received
View
Description
Withdrawing from the bank

Text

Text

Amount

50 TND

View

< Back

Button

```
// ./components/Header.jsx
const styles = StyleSheet.create({
    container: {
        width: '100%',
        display: 'flex',
        alignItems: 'flex-start',
    },
});
```

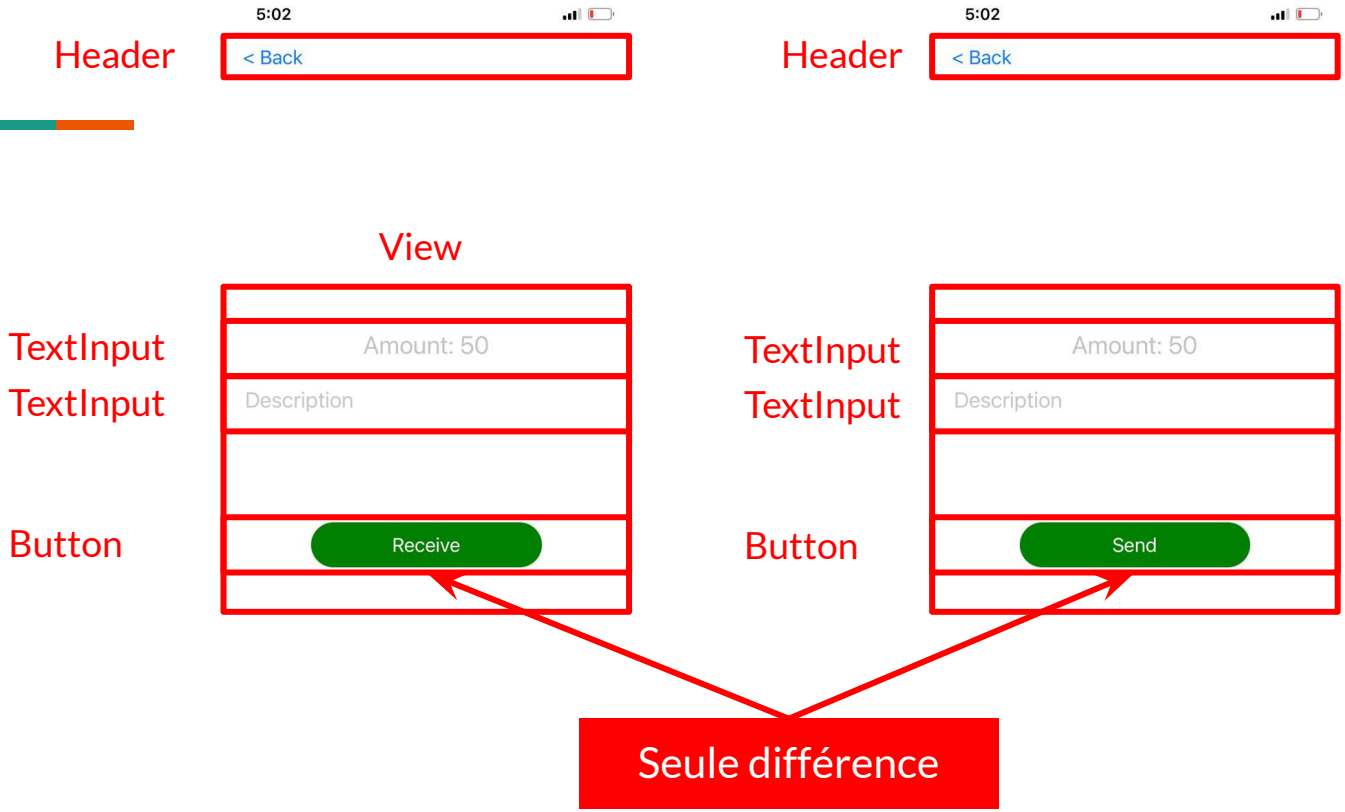
```
// ./components/Header.jsx
export function Header({ goToHome }) {
    return <View style={styles.container}>
        <Button title="< Back" onPress={goToHome} />
    </View>
}
```

```
// ./screens/PaymentScreen.jsx
const styles = StyleSheet.create({
    container: {
        width: '100%',
        flex: 1,
        paddingHorizontal: 10,
    },
    attrContainer: {
        marginVertical: 5,
    },
    value: {fontSize: 28}
});
```

```
// ./screens/PaymentScreen.jsx
export function PaymentScreen({ payment, goToHome }) {
    return <View style={styles.container}>
        <Header goToHome={goToHome} />
        <View style={styles.container}>
            <View style={styles.attrContainer}>
                <Text style={styles.title}>Amount</Text>
                <Text style={styles.value}>{payment.amount}
TND</Text>
            </View>
            …
        </View>
```

```
// ./screens/PaymentScreen.jsx
<View style={styles.container}>
    …
    <View style={styles.attrContainer}>
        <Text style={styles.title}>Type</Text>
        <Text style={styles.value}>{payment.type}</Text>
    </View>
    …
</View>
```

```
// ./screens/PaymentScreen.jsx
<View style={styles.container}>
    …
    <View style={styles.attrContainer}>
        <Text style={styles.title}>Description</Text>
        <Text style={styles.value}>{payment.description}</Text>
    </View>
</View>
```


5:02
Header
< Back
View
TextInput
Amount: 50
TextInput
Description
Button
Receive
5:02
Header
< Back
TextInput
Amount: 50
TextInput
Description
Button
Send
Seule différence

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    container: {
        flex: 1,
        width: '100%',
        display: 'flex',
        paddingHorizontal: 10,
    },
    form: {
        width: '100%',
        flex: 1,
        display: 'flex',
```

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    container: {
        flex: 1,
        width: '100%',
        display: 'flex',
        paddingHorizontal: 10,
    },
    …
})
```

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    …
    form: {
        width: '100%',
        flex: 1,
        display: 'flex',
        alignItems: 'center',
        justifyContent: 'center',
    },
    …
})
```

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    …
    amount: {
        fontSize: 23,
        margin: 10,
    },
    …
})
```

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    …
    description: {
        width: '100%',
        minHeight: 100,
        padding: 10,
        margin: 10,
        fontSize: 20,
    },
    …
})
```

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    …
    button: {
        width: '60%',
        padding: 10,
        margin: 20,
        backgroundColor: 'green',
        borderRadius: 25,
        display: 'flex',
        alignItems: 'center',
    },
```

```
// ./screens/NewPaymentScreen.js
const styles = StyleSheet.create({
    …
    buttonText: {
        color: 'white',
        fontSize: 18,
    },
})
```

```
// ./screens/NewPaymentScreen.js
export function NewPaymentScreen({ appendPayment, type, goToHome }) {
    const [ amount, setAmount ] = useState();
    const [ description, setDescription ] = useState();

    function savePayment() {
        appendPayment({amount, description, type});
        goToHome();
    }

    return <TouchableWithoutFeedback onPress={() =>
Keyboard.dismiss()}>
```

```
// ./screens/NewPaymentScreen.js
export function NewPaymentScreen({ appendPayment, type, goToHome }) {
return <TouchableWithoutFeedback onPress={() => Keyboard.dismiss()}>
    <View style={styles.container}>
        <Header goToHome={goToHome} />
        <View style={styles.form}>
            <TextInput style={styles.amount} value={amount}
        onChangeText={setAmount} placeholder="Amount: 50"
        keyboardType="numeric"/>
            …
        </View>
    </View>
```

```
// ./screens/NewPaymentScreen.js
export function NewPaymentScreen({ appendPayment, type, goToHome }) {
    …
    <View style={styles.form}>
        …
        <TextInput style={styles.description} value={description}
        onChangeText={setDescription} multiline={true}
        placeholder="Description" />
        <TouchableOpacity style={styles.button}
        onPress={savePayment}>
            <Text style={styles.buttonText}>{type ===
            PAYMENT_TYPE_SEND ? 'Send': 'Receive'}</Text>
```

Navigation entre les écrans différentes

```
// ./screens/index.js
export { HomeScreen } from './HomeScreen';
export { PaymentScreen } from './PaymentScreen';
export { NewPaymentScreen } from './NewPaymentScreen';

const SEND_PAYMENT_SCREEN = "send";
const RECEIVE_PAYMENT_SCREEN = "receive";
const PAYMENT_SCREEN = "payment";
…
```

```
// ./screens/index.js
…
export const screens_constants = {
    HOME_SCREEN: "home",
    SEND_PAYMENT_SCREEN: "send",
    RECEIVE_PAYMENT_SCREEN: "receive",
    PAYMENT_SCREEN: "payment"
}
```

```
// ./App.js
export default function App() {
    const [payments, setPayments] = useState([]);
    const [balance, setBalance] = useState(0);
    const [currentScreen, setCurrentScreen] = useState({name:
constants.HOME_SCREEN, params: ""});
    …
}
```

```
// ./App.js
    …
    const navigation = {
    goToHome: () => setCurrentScreen({name: constants.HOME_SCREEN,
    params: ""}),
    goToSendPayment: () => setCurrentScreen({name:
constants.SEND_PAYMENT_SCREEN, params: ""}),
    goToReceivePayment: () => setCurrentScreen({name:
constants.RECEIVE_PAYMENT_SCREEN, params: ""}),
    goToPayment : id => () => setCurrentScreen({name:
constants.PAYMENT_SCREEN, params: id}),
  }
```

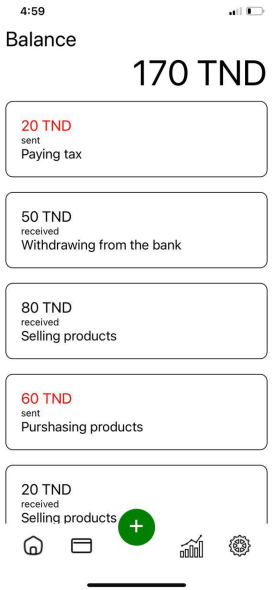
```
// ./App.js
    …
function appendPayment(payment) {
    setPayments(p => {
      try{
        let amount = parseInt(payment.amount);
        payment.type === PAYMENT_TYPE_RECEIVE ? setBalance(b => b +
amount) : setBalance(b => b - amount);
        try{
          payment.id = p.reduce((prev, curr) => curr.id > prev.id ?
curr : prev).id + 1;
        } catch(e) {payment.id = 0;}
```

```
// ./App.js
    …
return (
    <SafeAreaView style={styles.container}>
      { currentScreen.name === constants.HOME_SCREEN && <HomeScreen
balance={balance} payments={payments} navigation={navigation} />}
      { currentScreen.name === constants.SEND_PAYMENT_SCREEN &&
<NewPaymentScreen goToHome={navigation.goToHome}
appendPayment={appendPayment} type={PAYMENT_TYPE_SEND} />}
      …
    </SafeAreaView>
    );
```

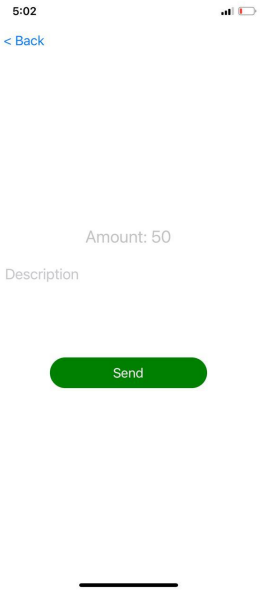
```
// ./App.js
    …
return (
<SafeAreaView style={styles.container}>
      …
      { currentScreen.name === constants.RECEIVE_PAYMENT_SCREEN &&
<NewPaymentScreen goToHome={navigation.goToHome}
appendPayment={appendPayment} type={PAYMENT_TYPE_RECEIVE} />}
      { currentScreen.name === constants.PAYMENT_SCREEN &&
<PaymentScreen payment={payments.filter(e => e.id ===
currentScreen.params)[0]} goToHome={navigation.goToHome} /> }
</SafeAreaView>
```

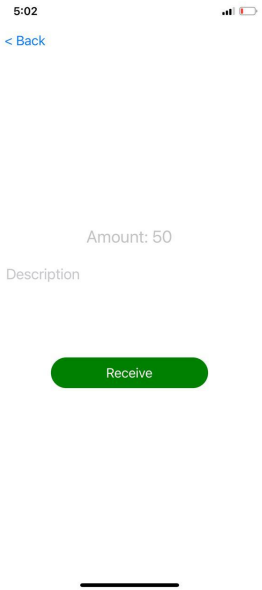
```
// ./screens/HomeScreen.js
export function HomeScreen({ balance, payments, navigation }) {
    …
    <Text>{balance} TND</Text>
    …
    <FlatList data={[...payments].reverse()}
renderItem={PaymentItem} />
    …
    <TouchableOpacity onPress={navigation.goToSendPayment}>
        …
    </TouchableOpacity>
    <TouchableOpacity onPress={navigation.goToReceivePayment}
```

Et voilà!


4:59
Balance
170 TND
20 TND
sent
Paying tax
50 TND
received
Withdrawing from the bank
80 TND
received
Selling products
60 TND
sent
Purshasing products
20 TND
received
Selling products


5:01
< Back
Amount
50 TND
Type
received
Description
Withdrawing from the bank


5:02
< Back
Amount: 50
Description
Send


5:02
< Back
Amount: 50
Description
Receive

Repository github du projet MyBank
https://github.com/ystn/payment-application-react-native-2022

Pour aller plus profond:

- https://reactnative.dev/docs/components-and-apis
- https://reactnavigation.org/docs/getting-started
- https://redux.js.org/introduction/getting-started

A très bientôt