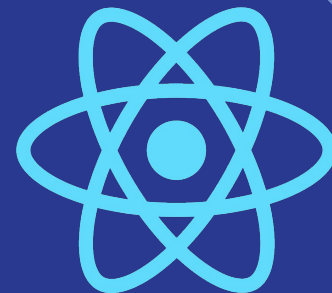


Formation React-Native Navigation



Bastien NICOLAS
bastien.nicolas@edu.esiee-it.fr

CODING
FACTORY
by ESIEE[tech]

Introduction

- Les modules de navigations
- React Navigation
- Naviguer entre les pages
- Autres méthodes de navigation
- Exercices

Les modules de navigations

- Il n'y a pas de navigation intégrée à React-native, il faut utiliser des modules externes
- Il existe plusieurs modules de navigation :
 - React Navigation (<https://reactnavigation.org/docs/getting-started/>)
 - React Native Navigation (<https://github.com/wix/react-native-navigation>)
 - React Router (<https://github.com/remix-run/react-router>)
 - Et d'autres ...
- Nous allons utiliser React Navigation (qui est un module React-native uniquement) car c'est le plus utilisé et le plus complet



React Navigation

- Il utilise en partie la navigation native Android et iOS et simule le reste en JS
- Il est très complet et paramétrable (modes de navigation, transitions des pages, ...)
- Il dispose d'une grosse communauté
- Toute l'application est contenue dans un composant `NavigationContainer` qui gère les différents mode de navigation



React Navigation

```
import * as React from 'react';

import {NavigationContainer} from '@react-navigation/native';

const App = () => {
  return (
    <NavigationContainer>
      {
        //... La navigation de l'application
      }
    </NavigationContainer>
  );
};

export default App;
```

Naviguer entre les pages

- Nous allons utiliser StackNavigator, c'est un composant qui permet de gérer les pages sous forme de pile, la page en haut de la pile est celle qui s'affiche
- Chaque page est un composant qui doit être déclarée dans le StackNavigator
- Pour afficher une nouvelle page :
 - `navigation.push(screenName, data)` -> créer une nouvelle page et l'ajoute en haut de la pile
 - `navigation.navigate(screenName, data)` -> si une page du même type existe il la met en haut de la pile sinon il fait la même chose que push
- Pour revenir à la page précédente :
 - `navigation.goBack()` -> retire la page en haut de la pile (la page actuelle)



Naviguer entre les pages

```
import * as React from 'react';
import {createStackNavigator} from '@react-navigation/stack';

const Stack = createStackNavigator();

import HomeScreen from '../screens/HomeScreen/HomeScreen';
import Page1Screen from '../screens/Page1Screen/Page1Screen';
import Page2Screen from '../screens/Page2Screen/Page2Screen';

const AppNavigator = () => {
  return (
    <Stack.Navigator initialRouteName="Home">
      <Stack.Screen name="Home" component={HomeScreen} />
      <Stack.Screen name="Page1" component={Page1Screen} />
      <Stack.Screen name="Page2" component={Page2Screen} />
    </Stack.Navigator>
  );
};

export default AppNavigator;
```

```
import * as React from 'react';

import {NavigationContainer} from '@react-navigation/native';

import AppNavigator from 'src/components/navigators/AppNavigator';

const App = () => {
  return (
    <NavigationContainer>
      <AppNavigator />
    </NavigationContainer>
  );
};

export default App;
```

Naviguer entre les pages

```
const Page1Screen = props => {
  const {navigation} = props;

  const goToPage2 = useCallback( () => {
    //navigation.push('Page2', {title: 'toto', quantity: 10});
    navigation.navigate('Page2', {title: 'toto', quantity: 10});
  }, [navigation]);

  const goBack = useCallback( () => {
    navigation.goBack();
  }, [navigation]);

  return (
    <SafeAreaView style={styles.screen}>
      <TouchableOpacity style={styles.btn} onPress={goToPage2}>
        <Text>Ouvrir la page 2</Text>
      </TouchableOpacity>
      <TouchableOpacity style={styles.btn} onPress={goBack}>
        <Text>Revenir à la page précédente</Text>
      </TouchableOpacity>
      <Increment />
    </SafeAreaView>
  );
};
```

```
const Page2Screen = ({route, navigation}) => {
  const {title, quantity} = route.params;

  return (
    <SafeAreaView>
      <Text>Title : {title}</Text>
      <Text>Quantity : {quantity}</Text>
    </SafeAreaView>
  );
};
```


Naviguer entre les pages

- Les pages déclarées en tant que `Stack.Screen` dans le `StackNavigator` reçoivent un props `"navigation"` qui permet de naviguer entre les pages, ainsi qu'un props `"route"` qui permet entre autre de récupérer les données passées par la page précédente
- Pour les autres composants (les enfants des pages par exemple), ils n'ont pas directement accès à la navigation via leurs props
- React navigation met à disposition un hook `useNavigation` qui permet de récupérer l'objet navigation depuis n'importe quel composant



Naviguer entre les pages

```
const Page1Screen = ({navigation}) => {  
  const goToPage2 = useCallback( callback: () => {  
    navigation.navigate('Page2', {title: 'toto', quantity: 10});  
  }, [navigation]);  
  
  const goBack = useCallback( callback: () => {  
    navigation.goBack();  
  }, [navigation]);
```

```
import {useNavigation} from '@react-navigation/native';  
  
const MyComponent = props => {  
  const navigation = useNavigation();  
  
  const goToPage2 = useCallback( callback: () => {  
    navigation.navigate('Page2', {title: 'toto', quantity: 10});  
  }, [navigation]);  
  
  const goBack = useCallback( callback: () => {  
    navigation.goBack();  
  }, [navigation]);
```

Autres méthodes de navigation - TabNavigator

- Il permet de créer une barre de navigation en bas de l'écran avec des tabs
- Il peut être couplé à un autre Navigator comme StackNavigator
- Il peut être complètement customisé

```
import React from 'react';
import {createBottomTabNavigator} from '@react-navigation/bottom-tabs';

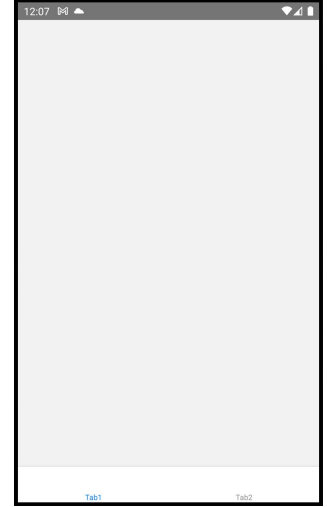
import Tab1Screen from '../screens/Tab1Screen/Tab1Screen';
import Tab2Screen from '../screens/Tab2Screen/Tab2Screen';

const Tab = createBottomTabNavigator();

const TabNavigator = () => {
  return (
    <Tab.Navigator>
      <Tab.Screen name="Tab1" component={Tab1Screen} />
      <Tab.Screen name="Tab2" component={Tab2Screen} />
    </Tab.Navigator>
  );
};
```

```
const App = () => {
  return (
    <NavigationContainer>
      <TabNavigator />
    </NavigationContainer>
  );
};
```

```
const StackNavigator = () => {
  return (
    <Stack.Navigator>
      <Stack.Screen name="TabNav" component={TabNavigator} />
      <Stack.Screen name="Page1" component={Page1Screen} />
      <Stack.Screen name="Page2" component={Page2Screen} />
    </Stack.Navigator>
  );
};
```



Autres méthodes de navigation - DrawerNavigator

- Il permet de créer un panneau latéral
- Il peut être couplé à un autre Navigator comme StackNavigator
- Il peut être complètement customisé

```
import React from 'react';
import {createDrawerNavigator} from '@react-navigation/drawer';

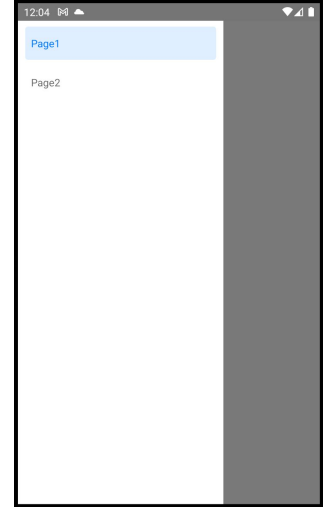
import Page1Screen from '../screens/Page1Screen/Page1Screen';
import Page2Screen from '../screens/Page2Screen/Page2Screen';

const Drawer = createDrawerNavigator();

const DrawerNavigator = () => {
  return (
    <Drawer.Navigator>
      <Drawer.Screen name="Page1" component={Page1Screen} />
      <Drawer.Screen name="Page2" component={Page2Screen} />
    </Drawer.Navigator>
  );
};
```

```
const App = () => {
  return (
    <NavigationContainer>
      <DrawerNavigator />
    </NavigationContainer>
  );
};
```

```
const StackNavigator = () => {
  return (
    <Stack.Navigator>
      <Stack.Screen name="DrawerNav" component={DrawerNavigator} />
      <Stack.Screen name="Page1" component={Page1Screen} />
      <Stack.Screen name="Page2" component={Page2Screen} />
    </Stack.Navigator>
  );
};
```



Exercice 1 - Todo list avec page de détail

A partir de la TodoList du cours précédent :

- Quand on appuie sur un item de la liste, une nouvelle page de détail s'ouvre
- Cette page ne doit pas avoir le header par défaut (à paramétrer au niveau du StackNavigator)
- La page doit afficher le texte de la Todo
- Un bouton retour permet de revenir à la liste des Todo



Ma Todo :

Faire les courses

Exercice 2 - TabBar avec tous les exercices

- Faire une navigation par TabBar qui permet d'avoir un onglet pour chaque exercice fait pendant la formation
- Ajouter une icône différente pour chaque onglet
- La couleur du label et de l'icône doit changer quand la page est focus
- Bonus : Faire la même chose avec une navigation par Drawer

