

# 🎯 Navigation par Pile (Stack Navigation) en React Native

---

## ✳️ Introduction à la navigation

### ◆ Pourquoi une navigation ?

Une application mobile contient souvent plusieurs **écrans** : accueil, profil, paramètres, etc. Pour passer de l'un à l'autre, on a besoin d'un **système de navigation**.

---

### ◆ Qu'est-ce qu'une navigation par pile ?

La **Stack Navigation** fonctionne comme une **pile d'assiettes** 🍔 :

- Tu **empiles** un nouvel écran → `navigate()`
- Tu **retires** le dernier pour revenir → `goBack()`

⌚ Cela suit le principe **LIFO (Last In, First Out)** :

```
[HomeScreen]           ← premier écran
↓ navigate('Details')
[HomeScreen]
[DetailsScreen]       ← visible à l'écran
```

Quand tu appuies sur “Retour”, `DetailsScreen` est retiré et `HomeScreen` réapparaît.

---

## ⚙️ Mise en place du projet

### ◆ Installation (avec Expo)

Ouvre ton terminal et exécute :

```
npm install @react-navigation/native
npx expo install react-native-screens react-native-safe-area-context
npm install @react-navigation/native-stack
```

Ces librairies servent à :

- **@react-navigation/native** → cœur de React Navigation
  - **react-native-screens / safe-area-context** → gestion des zones sécurisées et optimisations natives
  - **@react-navigation/native-stack** → gestion spécifique des piles d'écrans
- 

## Structure du projet

```
/App.js
/screens/
  HomeScreen.js
  DetailsScreen.js
```

---

## Mise en place de la Stack Navigation

 App.js

```
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/native-stack';
import HomeScreen from './screens/HomeScreen';
import DetailsScreen from './screens/DetailsScreen';

const Stack = createStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={HomeScreen} options={{ title: 'Accueil' }} />
        <Stack.Screen name="Details" component={DetailsScreen} options={{ title: 'Détails' }} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

---

 HomeScreen.js

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

export default function HomeScreen({ navigation }) {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Text>Accueil</Text>
      <Button
        title="Aller aux détails"
        onPress={() => navigation.navigate('Details')}
      />
    </View>
  );
}
```

```
) ;  
}
```

---

### 📘 DetailsScreen.js

```
import React from 'react';  
import { View, Text } from 'react-native';  
  
export default function DetailsScreen() {  
  return (  
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>  
      <Text>Page de détails</Text>  
    </View>  
  );  
}
```

---

## 🗣 Navigation et paramètres

### ◆ Envoyer un paramètre :

```
navigation.navigate('Details', { productId: 42 })
```

### ◆ Lire un paramètre :

```
export default function DetailsScreen({ route }) {  
  const { productId } = route.params;  
  return (  
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>  
      <Text>Détails du produit # {productId}</Text>  
    </View>  
  );  
}
```

---

## ⬅ BACK Revenir en arrière

Automatique :

Le bouton “Retour” est affiché dans le header si `headerShown` est activé (par défaut).

Manuel :

```
<Button title="Retour" onPress={() => navigation.goBack()} />
```

Remplacer un écran :

```
navigation.replace('Home')
```

---

## 🛠 Personnalisation du header

Globalement :

```
<Stack.Navigator  
  screenOptions={ {
```

```
        headerStyle: { backgroundColor: '#2196f3' },
        headerTintColor: '#fff',
        headerTitleStyle: { fontWeight: 'bold' },
    } }
>
```

Par écran :

```
<Stack.Screen
  name="Details"
  component={DetailsScreen}
  options={{
    title: 'Infos du produit',
    headerStyle: { backgroundColor: '#4caf50' },
  }}
/>
```

Cacher le header :

```
options={{ headerShown: false }}
```

---



## Navigation imbriquée (Stack + Tabs)

Structure fréquente :

```
NavigationContainer
  └── Tab.Navigator
    └── HomeStack
      ├── Home
      └── Details
    └── Settings
```

→ Tu peux combiner un **Stack Navigator** avec un **Tab Navigator** pour créer une navigation complexe.

---



## Exercice pratique : mini-app à 3 écrans

🎯 Objectif :

Créer une app avec :

- Home → Profile (envoie un nom)
  - Home → Settings
  - Boutons de retour sur chaque page
- 

### App.js

```
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/native-stack';
import HomeScreen from './screens/HomeScreen';
```

```
import ProfileScreen from './screens/ProfileScreen';
import SettingsScreen from './screens/SettingsScreen';

const Stack = createNativeStackNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home">
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="Profile" component={ProfileScreen} />
        <Stack.Screen name="Settings" component={SettingsScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

---

### 🏠 HomeScreen.js

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

export default function HomeScreen({ navigation }) {
  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Text style={{ fontSize: 20, marginBottom: 20 }}>
        Bienvenue sur l'accueil 🌟
      </Text>

      <Button
        title="Aller au profil d'Alex"
        onPress={() => navigation.navigate('Profile', { name: 'Alex' })}
      />

      <View style={{ marginTop: 15 }}>
        <Button
          title="Aller aux paramètres"
          onPress={() => navigation.navigate('Settings')}
        />
      </View>
    </View>
  );
}
```

---

### 👤 ProfileScreen.js

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

export default function ProfileScreen({ route, navigation }) {
  const { name } = route.params;

  return (
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>
      <Text style={{ fontSize: 18 }}>Profil de {name}</Text>
      <Button title="Retour à l'accueil" onPress={() => navigation.goBack()} />
    </View>
  );
}
```

```
) ;  
}
```

---

### ⚙️ SettingsScreen.js

```
import React from 'react';  
import { View, Text, Button } from 'react-native';  
  
export default function SettingsScreen({ navigation }) {  
  return (  
    <View style={{ flex: 1, alignItems: 'center', justifyContent: 'center' }}>  
      <Text style={{ fontSize: 18 }}>Page des paramètres ⚙️</Text>  
      <Button title="Retour à l'accueil" onPress={() =>  
        navigation.goBack() } />  
    </View>  
  );  
}
```

---

## ✓ Résumé des notions clés

Concept	Description	Exemple
<b>NavigationContainer</b>	Conteneur principal pour gérer la navigation	<NavigationContainer> ...</NavigationContainer>
<b>Stack.Navigator</b>	Définit la pile d'écrans	<Stack.Navigator>...</Stack.Navigator>
<b>Stack.Screen</b>	Définit un écran dans la pile	<Stack.Screen name="Home" component={HomeScreen} />
<b>navigate()</b>	Aller à un autre écran	navigation.navigate('Profile')
<b>goBack()</b>	Revenir à l'écran précédent	navigation.goBack()
<b>replace()</b>	Remplacer l'écran courant	navigation.replace('Home')
<b>route.params</b>	Lire les paramètres transmis	const { name } = route.params;

---