



La Navigation à Onglets (Bottom Tabs) en React Native



1. Comprendre la navigation dans React Native

1.1 Pourquoi une bibliothèque de navigation ?

React Native ne gère **aucune navigation nativement**.

Tu dois donc utiliser une bibliothèque — **React Navigation** est la plus populaire, car elle :

- est simple à configurer,
 - supporte les animations,
 - fonctionne sur Android & iOS,
 - offre plusieurs types de navigation :
 - **Stack Navigation** (écrans empilés comme une pile de cartes),
 - **Tab Navigation** (barre d'onglets en bas),
 - **Drawer Navigation** (menu latéral),
 - **Nested Navigation** (navigation imbriquée).
-



2. Installation et configuration initiale

2.1 Installation des dépendances principales

```
npm install @react-navigation/native
```

```
npm install react-native-screens react-native-safe-area-context
```

2.2 Installation des onglets (Bottom Tabs)

```
npm install @react-navigation/bottom-tabs
```

2.3 Installation des stacks (utile pour navigation imbriquée)

```
npm install @react-navigation/native-stack
```

2.4 (Facultatif) Installation des icônes

Pour des icônes dans les onglets :

```
npm install react-native-vector-icons
```

✿ 3. Création d'une navigation à onglets basique

Structure du projet :

```
MyTabsApp/  
├── App.js  
├── screens/  
│   ├── HomeScreen.js  
│   ├── SearchScreen.js  
│   └── SettingsScreen.js  
└── package.json
```

🏠 Étape 1 : Créer les écrans

screens/HomeScreen.js

```
import React from 'react';  
import { View, Text, Button } from 'react-native';  
  
export default function HomeScreen({ navigation }) {  
  return (  
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>  
      <Text style={{ fontSize: 22 }}> 🏠 Écran d'Accueil</Text>  
      <Button  
        title="Aller vers Détails"  
        onPress={() => navigation.navigate('Détails')}  
      />  
    </View>  
  );  
}
```

screens/SearchScreen.js

```
import React from 'react';
```

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

export default function SearchScreen() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text style={{ fontSize: 22 }}> 🔍 Écran de Recherche</Text>
    </View>
  );
}
```

screens/SettingsScreen.js

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

export default function SettingsScreen() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text style={{ fontSize: 22 }}> ⚙️ Paramètres</Text>
    </View>
  );
}
```

🏠 Étape 2 : Configurer la navigation principale

App.js

```
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
import Ionicons from 'react-native-vector-icons/Ionicons';

import HomeScreen from './screens/HomeScreen';
import SearchScreen from './screens/SearchScreen';
import SettingsScreen from './screens/SettingsScreen';

const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <NavigationContainer>
      <Tab.Navigator>
```

```

screenOptions={({ route }) => ({
  headerShown: false,
  tabBarIcon: ({ focused, color, size }) => {
    let iconName;
    if (route.name === 'Accueil') {
      iconName = focused ? 'home' : 'home-outline';
    } else if (route.name === 'Recherche') {
      iconName = focused ? 'search' : 'search-outline';
    } else if (route.name === 'Paramètres') {
      iconName = focused ? 'settings' : 'settings-outline';
    }
    return <Ionicons name={iconName} size={size} color={color} />;
  },
  tabBarActiveTintColor: '#007AFF',
  tabBarInactiveTintColor: 'gray',
  tabBarStyle: { backgroundColor: '#fff', height: 60 },
  tabBarLabelStyle: { fontSize: 12 },
}})
>
<Tab.Screen name="Accueil" component={HomeScreen} />
<Tab.Screen name="Recherche" component={SearchScreen} />
<Tab.Screen name="Paramètres" component={SettingsScreen} />
</Tab.Navigator>
</NavigationContainer>
);
}

```

4. Navigation imbriquée (Stack + Tabs)

Souvent, tu veux que chaque onglet ait sa propre pile d'écrans.

Par exemple : l'onglet **Accueil** contient une page principale + une page **Détails**.

Étapes :

1. Crée un **Stack Navigator** pour l'onglet Accueil.
2. Imbrique ce stack dans le **Tab Navigator**.

Exemple :

```

import { createNativeStackNavigator } from '@react-navigation/native-stack';
const HomeStack = createNativeStackNavigator();

```

```
function HomeStackScreen() {
  return (
    <HomeStack.Navigator>
      <HomeStack.Screen name="Accueil" component={HomeScreen} />
      <HomeStack.Screen name="Détails" component={DetailScreen} />
    </HomeStack.Navigator>
  );
}
```

Puis, dans ton `Tab.Navigator` :

```
<Tab.Screen name="Accueil" component={HomeStackScreen} />
```



5. Personnalisation avancée

5.1 Style de la barre

```
tabBarStyle: {
  backgroundColor: '#121212',
  borderTopWidth: 0,
  height: 70,
  paddingBottom: 10,
}
```

5.2 Couleur dynamique des icônes

```
tabBarIcon: ({ focused, color, size }) => {
  return <Ionicons name={focused ? 'home' : 'home-outline'} size={30} color={color} />;
}
```

5.3 Badge de notification

```
<Tab.Screen
  name="Paramètres"
  component={SettingsScreen}
  options={{ tabBarBadge: 3 }}
/>
```



6. Mini-projet : “MyTabsApp”

🎯 Objectif :

Créer une application à 3 onglets avec :

- Navigation imbriquée dans Home,
- Icônes dynamiques,
- Badge de notifications,
- Thème clair/sombre.

📁 Structure :

```
MyTabsApp/  
├── App.js  
├── screens/  
│   ├── HomeScreen.js  
│   ├── DetailScreen.js  
│   ├── SearchScreen.js  
│   └── SettingsScreen.js  
└── components/  
    └── CustomHeader.js
```

⚡ Étapes :

1. Mettre en place le **Tab Navigator**.
2. Créer un **HomeStackNavigator**.
3. Ajouter un bouton dans Home pour aller vers Détails.
4. Ajouter un **badge** sur l'onglet Paramètres.
5. Personnaliser la barre d'onglets (couleurs, icônes, label).



7. Bonnes pratiques professionnelles

- ✓ Utilise des stacks séparés pour chaque onglet.
- ✓ Ne surcharge pas la barre d'onglets (max 5 onglets).
- ✓ Utilise des icônes cohérentes avec le thème.
- ✓ Garde des noms d'écrans explicites (Home, Profile, Settings).
- ✓ Teste sur Android et iOS (différences de rendu).

