

TP Expo : Créez une mini-app de navigation avec expo-router

Objectifs

Créer une application Expo avec plusieurs écrans, une barre d'onglets personnalisée, une navigation en pile (stack), une gestion de la Safe Area, et une route dynamique simulant une "fiche produit".

Prérequis

- Expo CLI installé
 - VS Code ou autre éditeur
 - `npx create-expo-app MonApp -e with-router`
-

Étape 1 — Initialisation du projet

1. Créez un nouveau projet :

```
npx create-expo-app TP-Navigation -e with-router
```

```
cd TP-Navigation
```

```
npm install
```

```
npx expo start ou npm run android
```

2. Testez que le projet s'affiche correctement sur un téléphone.
-

Étape 2 — Structure des dossiers

Organisez le dossier `app/` comme ceci :

`app/`

|— `_layout.tsx` ← Navigation Tabs

|— `accueil/`

| |— `_layout.tsx` ← Stack pour Accueil

| |— `index.tsx` ← Page d'accueil

| |— `details.tsx` ← Détails (Stack)

|— `utilisateurs/`

```
|   └─ [id].tsx      ← Route dynamique
|   └─ favoris/
|   └─ index.tsx
```

Étape 3 — Navigation Tabs (app/_layout.tsx)

► Objectif : Afficher 3 onglets avec icônes

```
import { Tabs } from 'expo-router';
import { Ionicons } from '@expo/vector-icons';
```

```
export default function Layout() {
  return (
    <<<< Votre code ici >>>>
  );
}
```

Étape 4 — Stack dans l'onglet Accueil (app/accueil/_layout.tsx)

► Objectif : Naviguer vers un écran "details" avec retour automatique

```
import { Stack } from 'expo-router';

export default function AccueilLayout() {
  return (
    <Stack
      screenOptions={{
        headerStyle: { backgroundColor: '#007aff' },
        headerTintColor: 'white',
      }}
    />
  );
}
```

```
}
```

Étape 5 — SafeAreaView + navigation Stack (app/accueil/index.tsx)

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

import { useRouter } from 'expo-router';

export default function Accueil() {

  const router = useRouter();

  return (

    <<<<<< Le code ici >>>>>>

  )
}
```

Étape 6 — Page details (app/accueil/details.tsx)

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

export default function Details() {

  return (

    <<<<<<< Le code ici >>>>>>>>>>

  )
}
```

Étape 7 — Route dynamique : /utilisateurs/[id].tsx

► Objectif : Afficher des informations différentes selon l’ID



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

import { useRouter } from 'expo-router';

export default function Utilisateurs() {

  <<<<<<<< Le code ici >>>>>>>>>>>> }
}
```

Objectifs BONUS

-  Créez un écran favoris/details.tsx avec Stack (ajoutez un _layout.tsx)
-  Ajoutez un thème clair/sombre dynamique avec useColorScheme()

Concepts couverts

Concept	Fichier d'exemple
Tabs	app/_layout.tsx
Stack	app/accueil/_layout.tsx
SafeAreaProvider (global) (inclus par expo-router)	
SafeAreaView	accueil/index.tsx,
useRouter	[id].tsx