

React Native Expo Router Project Structure

Expo Router v3 file-based navigation. Next.js-style routing for React Native.

#react-native #expo #expo-router #mobile #typescript

PNG

PDF

Copy

</> Prompt

Project Directory

my-app/

- app.json Expo config
- package.json
- tsconfig.json
- babel.config.js
- .gitignore
- app/ File-based routing
 - _layout.tsx Root layout with navigation
 - index.tsx /
 - +not-found.tsx 404 page
- (auth)/ Auth route group
 - _layout.tsx
 - login.tsx
 - register.tsx
- (tabs)/ Tab navigation
 - _layout.tsx Tab bar configuration
 - index.tsx Home tab
 - explore.tsx
 - profile.tsx
- settings/ Stack routes
 - index.tsx
 - notifications.tsx
 - [id].tsx Dynamic route
- components/
 - Button.tsx
 - ThemedText.tsx
 - ThemedView.tsx
- hooks/
 - useAuth.ts
 - useColorScheme.ts
- services/
 - api.ts
- constants/
 - Colors.ts
- assets/
 - images/
 - fonts/

Why This Structure?

Expo Router brings Next.js-style file-based routing to React Native. Routes are defined by file structure in `app/`. Groups like `(tabs)` define navigation patterns. Dynamic routes use `[id].tsx`. Deep linking works automatically.

Key Directories

- `app/` - File path = route path, like Next.js
- `app/(tabs)/` - Grouped routes sharing tab navigation
- `app/(auth)/` - Grouped routes without affecting URL
- `app/[id].tsx` - Dynamic segments for params
- `app/_layout.tsx` - Layout components wrap nested routes

Tab Layout

```
// app/(tabs)/_layout.tsx
export default function TabLayout() {
  return (
    <Tabs>
      <Tabs.Screen
        name="index"
        options={{
          title: 'Home',
          tabBarIcon: ({ color }) => <IconHome color={color} />
        }}
      />
      <Tabs.Screen name="explore" options={{ title: 'Explore' }} />
    </Tabs>
  );
}
```

Getting Started

- `npx create-expo-app@latest my-app --template tabs`
- `cd my-app`
- `npx expo start`
- Add routes by creating files in `app/`

When To Use This

- Want Next.js-like routing for mobile
- Need automatic deep linking
- Prefer file-based over imperative navigation
- Building apps with shared navigation patterns
- Want typed routes with TypeScript

Trade-offs

- Learning curve** - File conventions different from React Navigation
- Expo dependency** - Expo Router is Expo-only
- Less control** - Some React Navigation features harder to access