React Native
Module – Navigation

Peter Kassenaar –
info@kassenaar.com

Adding Navigation

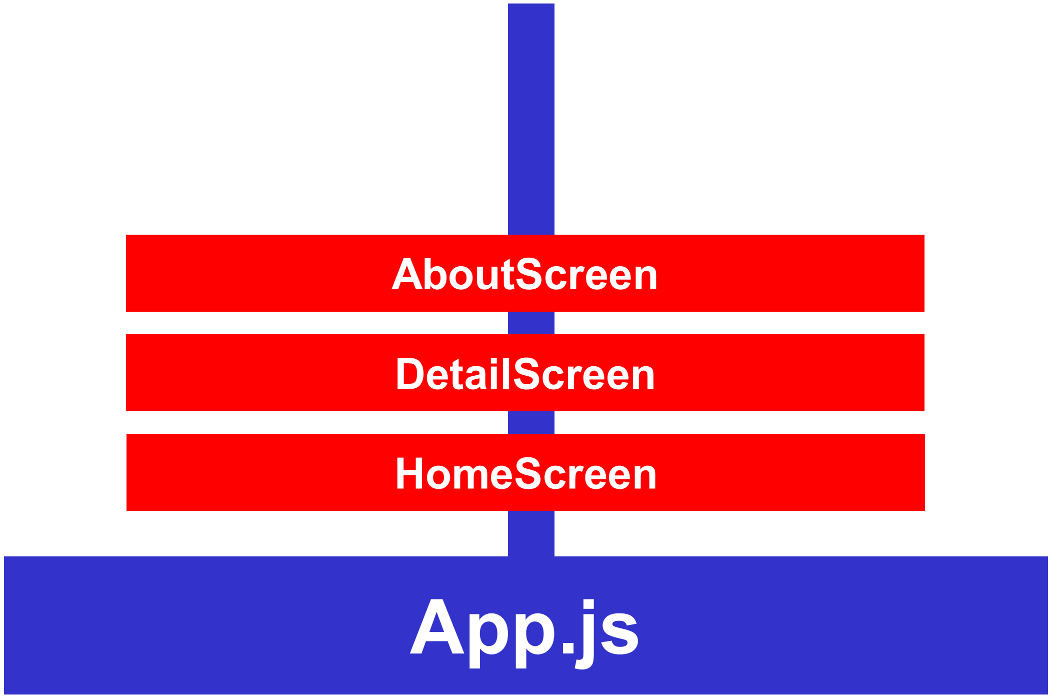
Navigating between screens in your app

Routing in React Native

- React Native DOES NOT HAVE default routing, like a web browser, or on the web.
- So NO `<a href="…">` tags available
- You have to install routing as separate packages
- There are different types of routing you can use
  - Stack navigation
  - Drawer navigation
  - Nested navigation (nested stacks)

Navigation in React Native

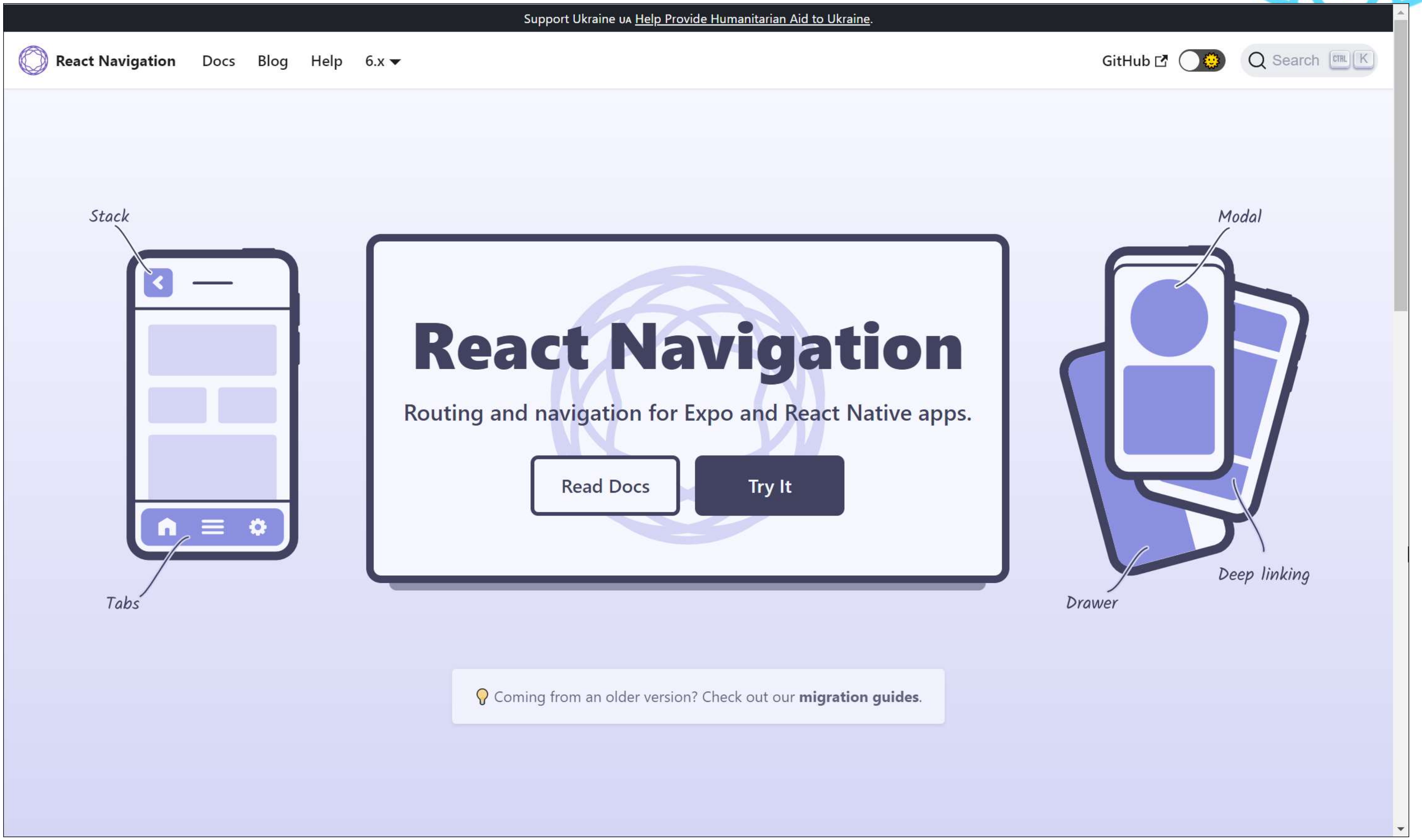
Navigation is like a ‘stack of screens’



Moving between screens

- You `push()` new screens on the stack
- You pop screens off the stack (by using `.goBack()`)
- You can pass data between screens, as `route` props
- Screens have an automatic `navigation` prop
- Your app lives inside a `<NavigationContainer>` component
- There typically is only one (1) NavigationContainer per app

Installing Routing



Support Ukraine 🇺🇦 Help Provide Humanitarian Aid to Ukraine.

React Navigation

Routing and navigation for Expo and React Native apps.

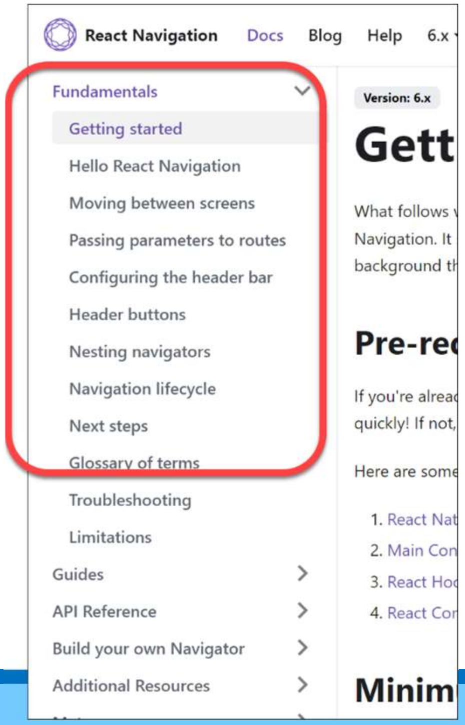
💡 Coming from an older version? Check out our **migration guides**.

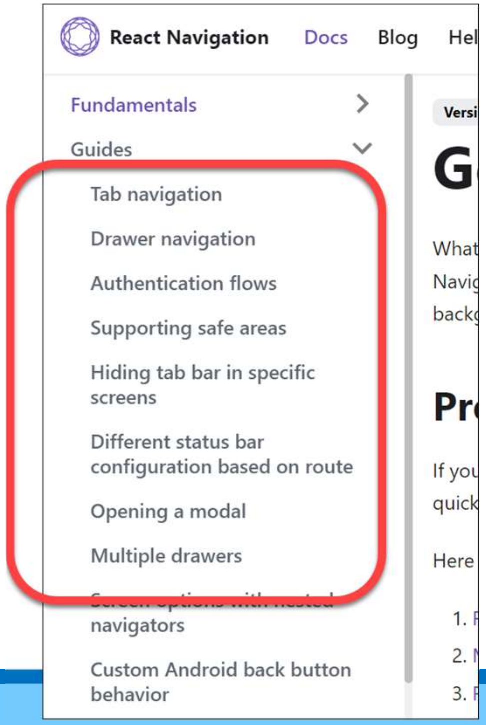
Warning: LOTS of outdated stuff online. Make sure to use `version 6.x`!!

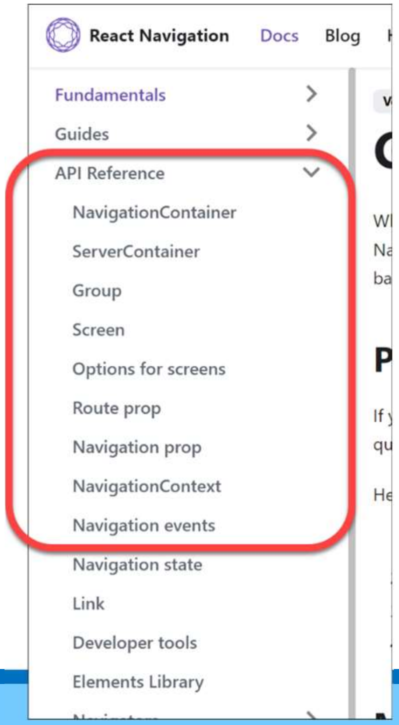
https://reactnavigation.org/

Recommended steps

- 1. Read Getting Started, to get familiar with routing
- 2. Read Guides, for different types of navigation
- 3. Read API docs per item that you use







Installing Routing

- In a new, or non-routing project:

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

- Also, add extra dependencies:

```
npx expo install react-native-screens react-native-safe-area-context
```

(in Expo project)

Wrapping the app

- Now, you need to wrap the whole app in `<NavigationContainer />`.
    - Usually, in your entry file such as `index.js` or `App.js`:

```
export default function App() {
    return (
        <NavigationContainer>
                {/*Rest of the app code…*/}
        </NavigationContainer>
    );
}
```

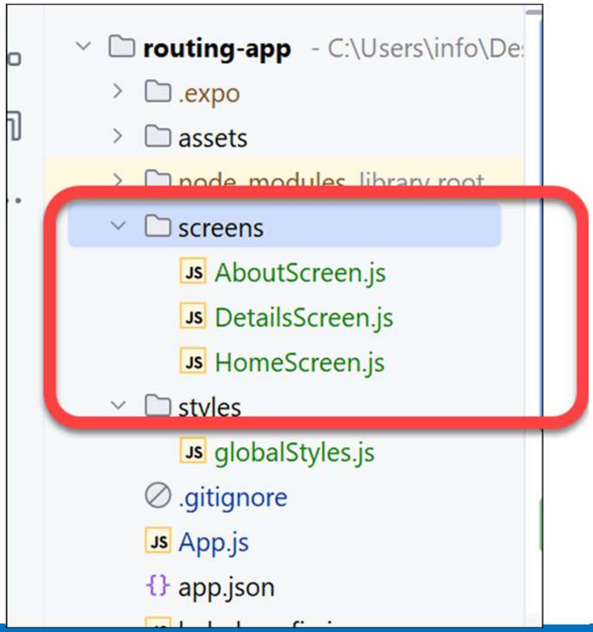
You typically have only 1 `<NavigationContainer />`, unless you have specific use cases

Multiple screens

“If your app uses only one stack navigator then it is conceptually similar to how a web browser handles navigation state - your app pushes and pops items from the navigation stack as users interact with it, and this results in the user seeing different screens”

Create additional screens

- We’ve put it in a `/screens` directory.
  - But of course this is convention. You can name it whatever you want
- Created 3 additional screens:



```
routing-app - C:\Users\info\De
  .expo
  assets
  node_modules library root
  screens
    AboutScreen.js
    DetailsScreen.js
    HomeScreen.js
  styles
    globalStyles.js
  .gitignore
  App.js
  app.json
```

Installing the stack navigator library

- Every navigator lives in its own package. So you also have to install that one (sigh…)

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

- Update `<App />` to use the StackNavigator:

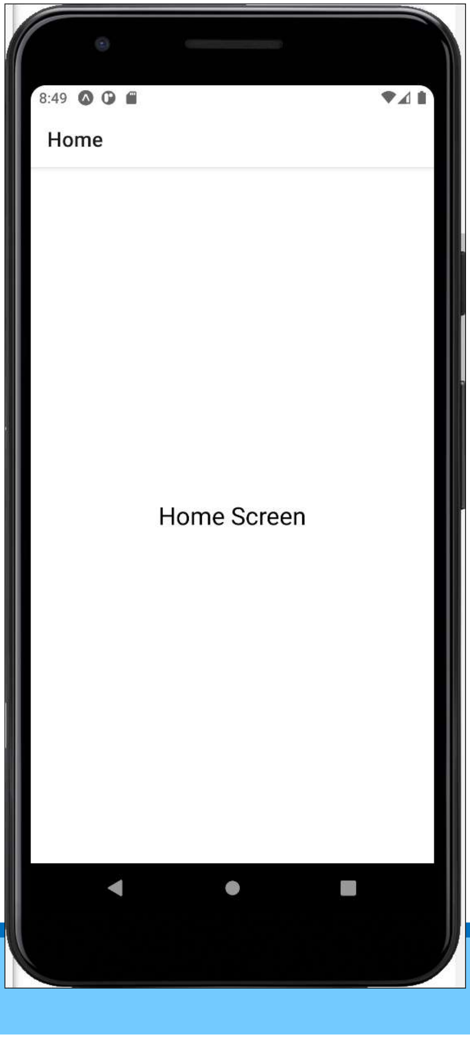
```
import {createNativeStackNavigator} from "@react-navigation/native-stack";
import …

const Stack = createNativeStackNavigator();
export default function App() {
    return (
        <NavigationContainer>
            <Stack.Navigator>
                <Stack.Screen name="Home" component={HomeScreen}/>
                 …more screens…
            </Stack.Navigator>
        </NavigationContainer>
    );
}
```

What is your ‘Home’ screen?

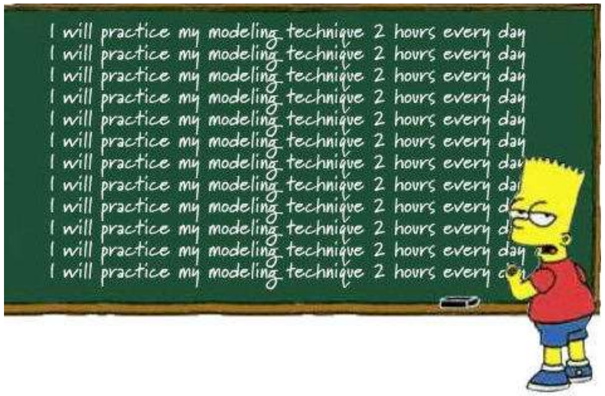
- The screen that is listed first, is by default the Home Screen.
- OR: set the property `initialRouteName`:

```
<Stack.Navigator
         initialRouteName="Home">
```



Workshop

- Create a new React app, or work from your existing app
- Add routing to the app, as described before
- Add three screens to the app. See if you can assign each screen as the ‘home’ screen
- (we’re going to navigate between screens next)
- Example: ../200-navigation-basics
- https://reactnavigation.org/docs/getting-started


I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every da
I will practice my modeling technique 2 hours every d
I will practice my modeling technique 2 hours every d
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every d

Checkpoint

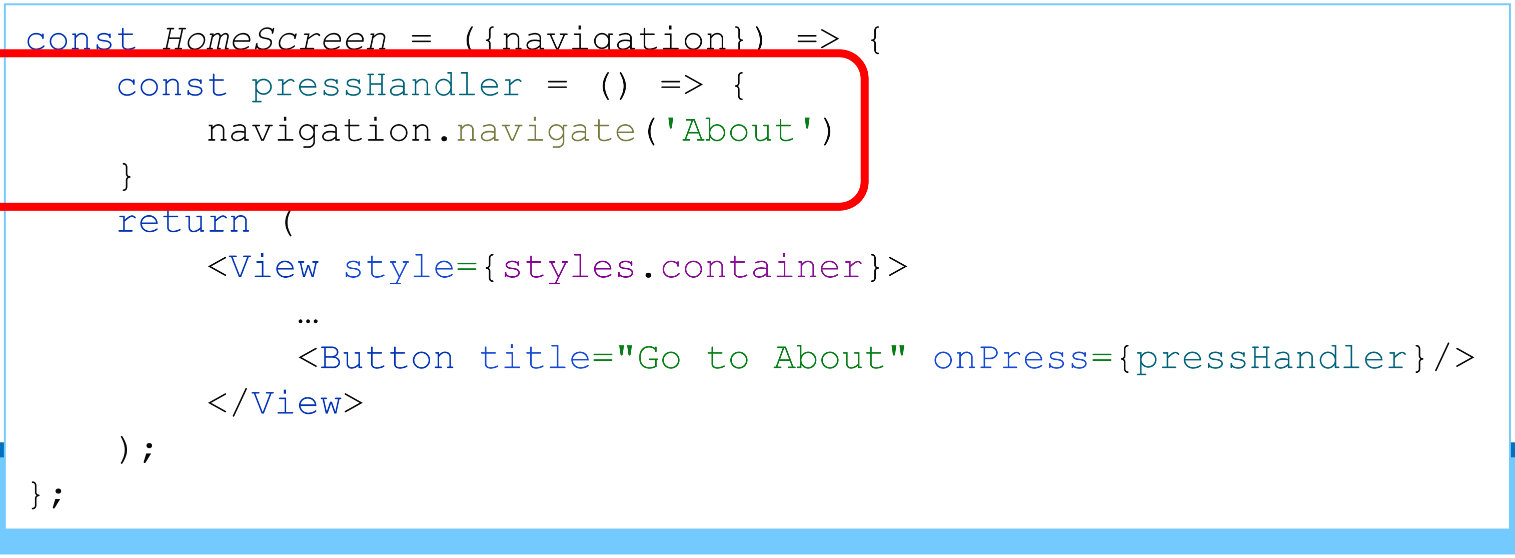
- React Native doesn't have a built-in API for navigation like a web browser does. React Navigation provides this for you.
- `Stack.Navigator` is a component that takes route configuration as its children
- There are props for configuration and rendering our content.
- Each `Stack.Screen` component takes a `name` which refers to the name of the route and `component` which specifies the component to render for the route.
- These are the 2 required props.

Navigating between screens

Moving from one screen to another and back to the home screen

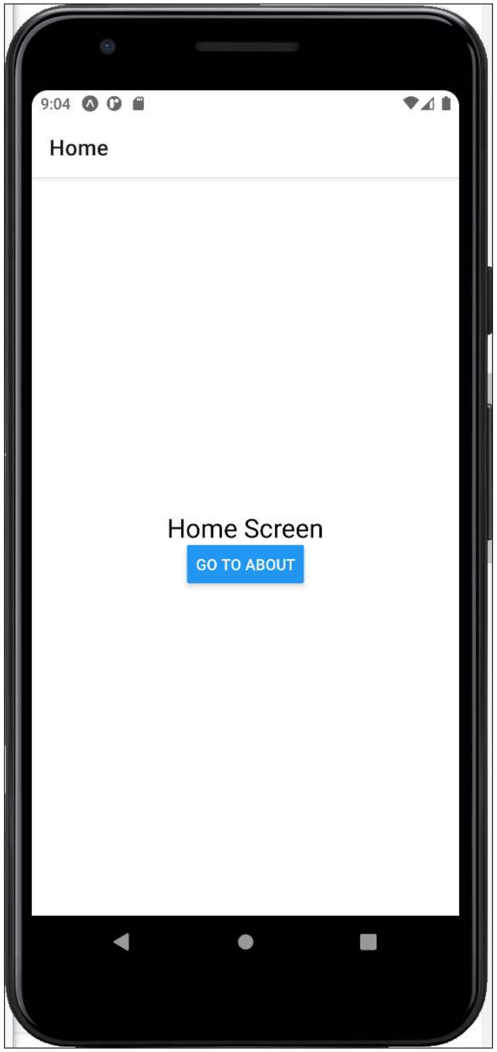
Navigation on the web

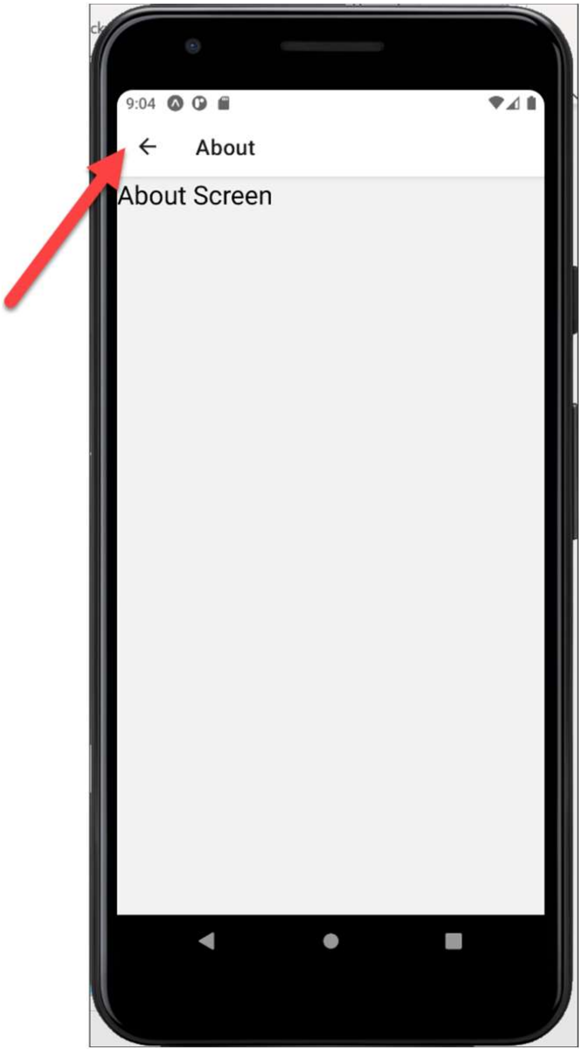
- `<a href="details.html">Go to Details</a>`
- NOT Valid in React Native
- You create a `Button` (or `Text`, or `Icon`, or `Pressable`) and handle the `onPress` event, for instance:



```
const HomeScreen = ({navigation}) => {
    const pressHandler = () => {
        navigation.navigate('About')
    }
    return (
        <View style={styles.container}>
            …
            <Button title="Go to About" onPress={pressHandler}/>
        </View>
    );
};
```

Result

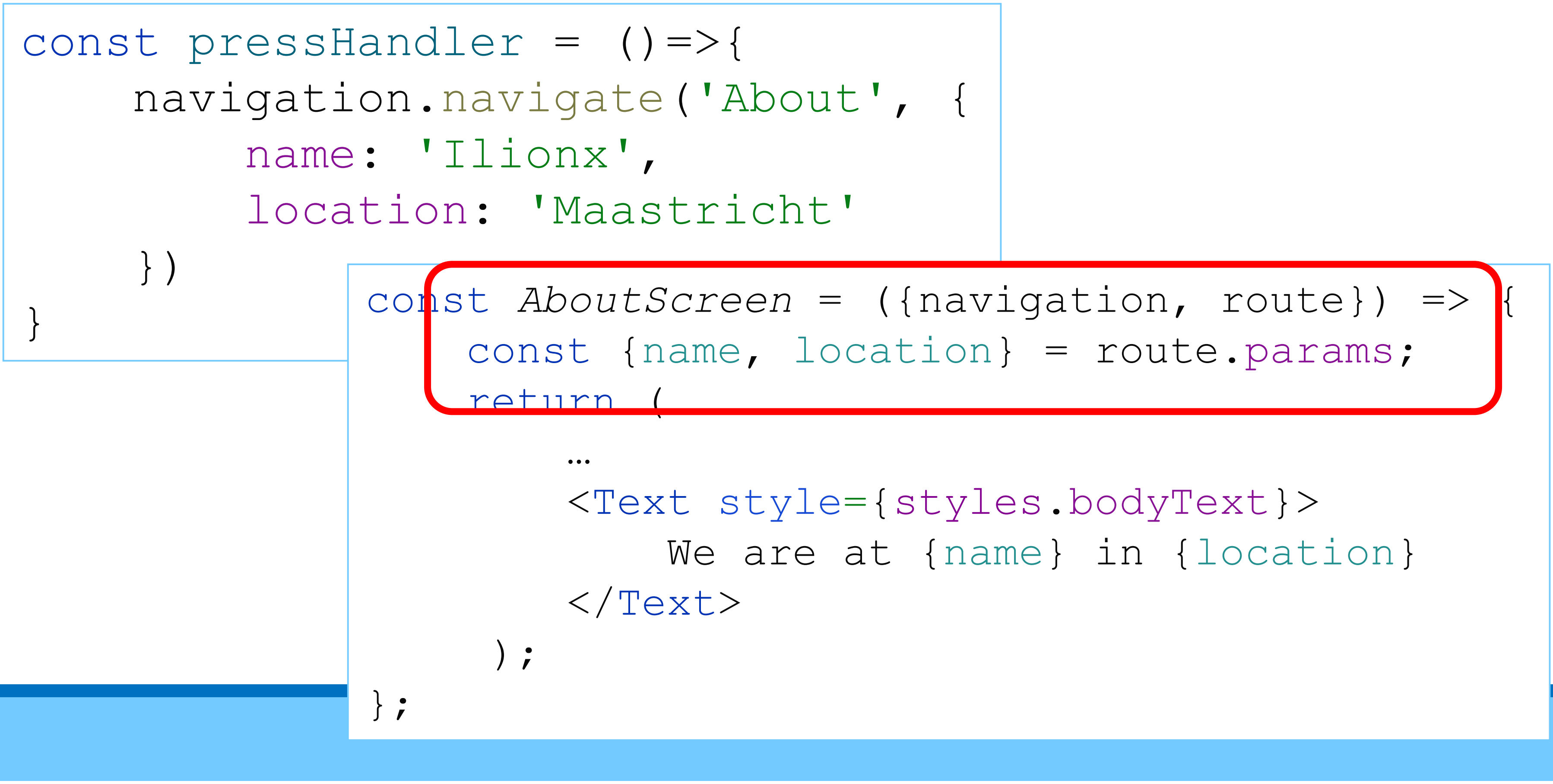




You get the Back button and navigation bar for free. These of course are highly configurable via props.

Passing data between screens

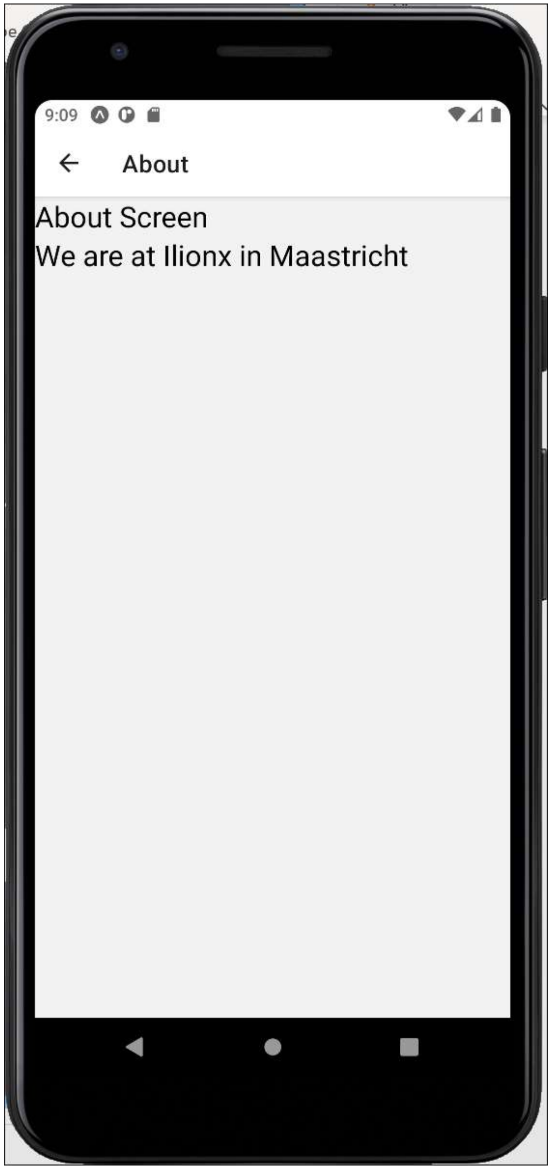
- We can send a configuration/params object to the next screen, upon navigation like so:



```
const pressHandler = ()=>{
    navigation.navigate('About', {
        name: 'Ilionx',
        location: 'Maastricht'
    })
}
```

```
const AboutScreen = ({navigation, route}) => {
    const {name, location} = route.params;
    return (
        …
        <Text style={styles.bodyText}>
            We are at {name} in {location}
        </Text>
    );
};
```

Result



About Screen
We are at Ilionx in Maastricht

Going back programmatically:

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

Going to the very first screen in the app:

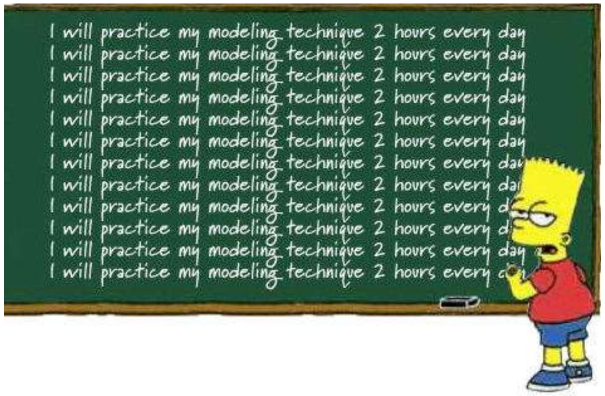
```
<Button title="Go back" onPress={() =>
       navigation.popToTop()} />
```

Checkpoint

- `navigation.navigate('RouteName')` pushes a new route to the native stack navigator if it's not already in the stack, otherwise it jumps to that screen.
- The header bar will automatically show a back button, but you can programmatically go back by calling `navigation.goBack()`.
- You can go back to the first screen in the stack with `navigation.popToTop()`.
- The `navigation` prop is available to all `Screen` components.

Workshop

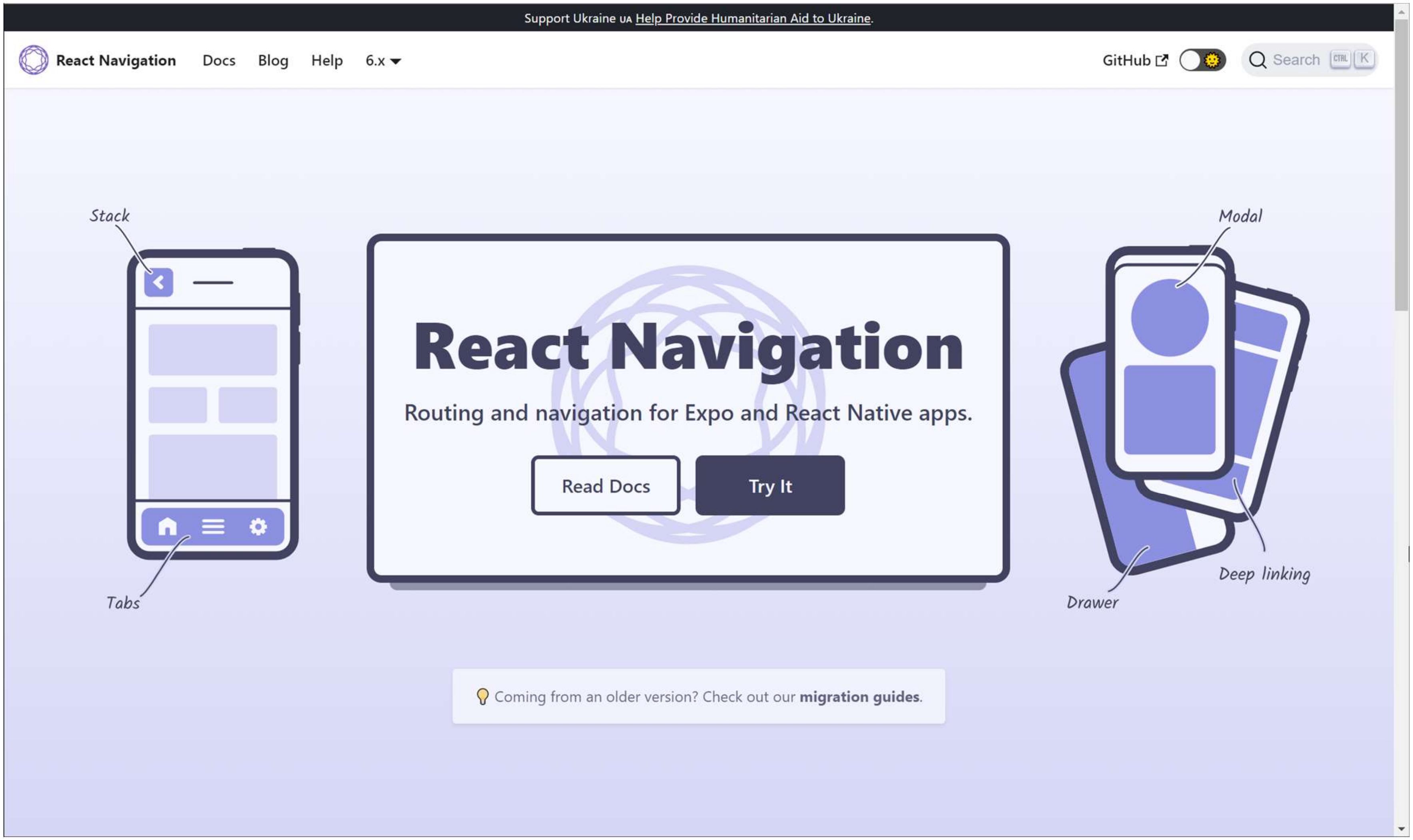
- Create a list in your navigation app. Load the list on the Home Screen
- When you tap an item in the list, navigate to a detail screen showing details for that item
- Experiment with `.goBack()` and `popToTop()`.
- Optional:
  - Send the tapped name, show in the Title Bar
  - Can you change the navigation animation?
- Example: `../210-navigation-data`
- https://reactnavigation.org/docs/navigating


I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every da
I will practice my modeling technique 2 hours every da
I will practice my modeling technique 2 hours every d
I will practice my modeling technique 2 hours every d
I will practice my modeling technique 2 hours every day
I will practice my modeling technique 2 hours every d

[more navigation to come…]

More info

More info on the topics in this module



Support Ukraine UA Help Provide Humanitarian Aid to Ukraine.

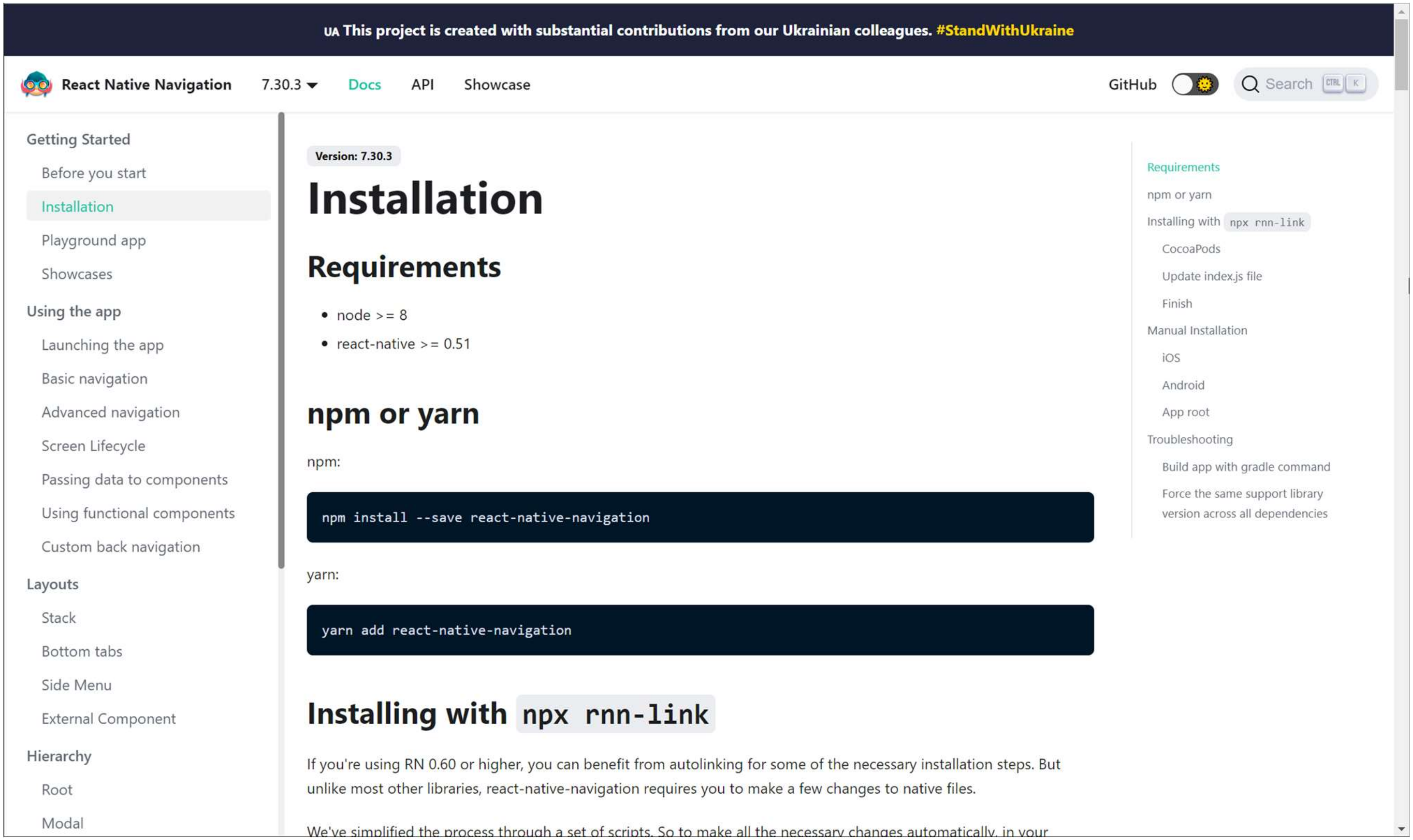
React Navigation

Routing and navigation for Expo and React Native apps.

Coming from an older version? Check out our **migration guides**.

https://reactnavigation.org/

Alternative navigation: WIX



Version: 7.30.3

Installation

Requirements

- node >= 8
- react-native >= 0.51

npm or yarn

npm:

```
npm install --save react-native-navigation
```

yarn:

```
yarn add react-native-navigation
```

Installing with `npx rnn-link`

If you're using RN 0.60 or higher, you can benefit from autolinking for some of the necessary installation steps. But unlike most other libraries, react-native-navigation requires you to make a few changes to native files.

We've simplified the process through a set of scripts. So to make all the necessary changes automatically, in your

https://wix.github.io/react-native-navigation/docs/before-you-start/