Navigation

Navigation

The community solution to navigation is a standalone library that allows developers to set up the screens of an app with just a few lines of code.

introducing `react-navigation`

Navigation

Three main types of navigators

`StackNavigator`

`TabNavigator`

`DrawerNavigator`

Navigation

StackNavigator

Step 1: Install react-navigation

```
npm i react-navigation —save
```

Navigation

StackNavigator

Step 2: import StackNavigator into your component

```
import {
  StackNavigator,
} from 'react-navigation';
```

Navigation

StackNavigator

StackNavigator takes two arguments

```
StackNavigator(RouteConfigs: object, StackNavigatorConfig: object)
```

```
const MyHomeRoutes = StackNavigator({
  Home: { screen: Home },
  Info: { screen: Info }
},{
  initialRouteName: 'Info',
});
```

Navigation

StackNavigator

Step 3: Create navigation screens using components

```
const App = StackNavigator({
  Home: { screen: Home },
  Info: { screen: Info },
});
```

Step 4: Render Navigation

```
AppRegistry.registerComponent('App', () => App);
```

Navigation

StackNavigator - Basic Implementation

```
const Home = () => (
  <Text>Hello from Home</Text>
);

const Info = () => (
  <Text>Hello from Info</Text>
);

const App = StackNavigator({
  Home: { screen: Home },
  Info: { screen: Info }
});

AppRegistry.registerComponent('App', () => App)
```

Navigation

StackNavigator - Navigation Options

```
const Home = () => (
  <Text>Hello from Home</Text>
);
Home.navigationOptions = {
  title: 'Home'
};

// or

class Home extends React.Component {
  static navigationOptions = {
    title: 'Home'
  }
  render() {
    return (
      <Text>Hello from Home</Text>
    );
  }
}
```

Navigation

StackNavigator - navigation prop

Every scene receives a navigation prop (object) with these properties:

navigate - (helper) link to other screens
state - screen's current state/routes
setParams - (helper) make changes to route's params
goBack - (helper) close active screen and move back
dispatch - send an action to router

Navigation

StackNavigator - Pushing / Popping Routes

```
const Home = (props) => (
  <Text onPress={() => props.navigation.navigate(‘Info')}>Hello from Home</Text>
)

const Info = ({ navigation }) => (
  <Text onPress={() => navigation.goBack()}>Hello from Info</Text>
)
```

Navigation

TabNavigator

Step 1: import TabNavigator into your component

```
import {
  TabNavigator,
} from 'react-navigation';
```

Navigation

TabNavigator

TabNavigator also takes two arguments

```
TabNavigator(RouteConfigs: object, TabNavigatorConfig : object)

const App = TabNavigator({
  Home: { screen: Home },
  Info: { screen: Info }
},{
  tabBarOptions:{
    activeTintColor: ‘#ff9900’,
    // showIcon is only necessary for android
    showIcon: true,
  },
});
```

Navigation

TabNavigator - Basic Implementation

```
const Home = () => (
  <Text>Hello from Home</Text>
)

const Info = () => (
  <Text>Hello from Info</Text>
)

const App = TabNavigator({
  Home: { screen: Home },
  Info: { screen: Info }
},{
  tabBarOptions:{
    activeTintColor: ‘#ff9900’,
    // showIcon is only necessary for android
    showIcon: true,
  },
});

AppRegistry.registerComponent('App', () => App)
```

TabNavigator - Add Icons

```
const Home = () => (
  <Text>Hello from Home</Text>
)

Home.navigationOptions = {
  tabBarIcon: ({ tintColor }) => (
    <Image
      source={require('./homeicon.png')}
      style={{ width: 24, height: 24, tintColor }}
    />
  ),
}
```

Navigation

TabNavigator - Nested Navigators

First, move navigationOptions into screen configuration

```
const Nav = TabNavigator({
  Home: {
    screen: Home,
    navigationOptions: {
      tabBarIcon: ({ tintColor }) => (
        <Image
          source={require('./homeicon.png')}
          style={{ width: 24, height: 24, tintColor }}
        />
      ),
    }
  },
  Info: { screen: Info },
});
```

Navigation

TabNavigator - Nested Navigators

Next, update Home to have a link to navigate, and add a new About route

```
const Home = ({ navigation }) => (
  <View style={styles.container}>
    <Text>Home</Text>
    <Text onPress={() => navigation.navigate('About')}>Go To About</Text>
  </View>
)

const About = () => (
  <View style={styles.container}>
    <Text>About</Text>
  </View>
)
```

Navigation

TabNavigator - Nested Navigators

Now, create new route stack using StackNavigator

```
const HomeRouteStack = StackNavigator({
  Home: { screen: Home },
  About: { screen: About }
})
```

And replace existing Home route with new HomeRouteStack

```
const Nav = TabNavigator({
  Home: {
    screen: HomeRouteStack,
    ...
  },
  Info: { screen: Info },
});
```

Navigation

DrawerNavigator

Step 1: import DrawerNavigator into your component

```
import {
  DrawerNavigator,
} from 'react-navigation';
```

Navigation

DrawerNavigator

DrawerNavigator also takes two arguments

```
DrawerNavigator(RouteConfigs: object, DrawerNavigatorConfig : object)
```

```
const App = DrawerNavigator({
  Home: { screen: Home },
  Info: { screen: Info }
},{
  contentOptions:{
    activeTintColor: '#ff9900'
  },
});
```

Navigation

DrawerNavigator - Basic Implementation

```
const Home = ({ navigation }) => (
  <Text onPress={() => navigation.navigate('DrawerOpen')}>Open Drawer</Text>
)

const Info = ({ navigation }) => (
   <Text onPress={() => navigation.navigate('DrawerOpen')}>Open Drawer</Text>
)

const App = DrawerNavigator({
  Home: { screen: Home },
  Info: { screen: Info }
});
```

Navigation

DrawerNavigator - Adding Icons

```
const Home = ({ navigation }) => (
  <Text onPress={() => navigation.navigate('DrawerOpen')}>Open Drawer</Text>
)
Home.navigationOptions = {
  drawerIcon: ({ tintColor }) => (
    <Image
      source={require('./homeicon.png')}
      style={{ width: 24, height: 24, tintColor }}
    />
  ),
}
```