

React Native

Navigation

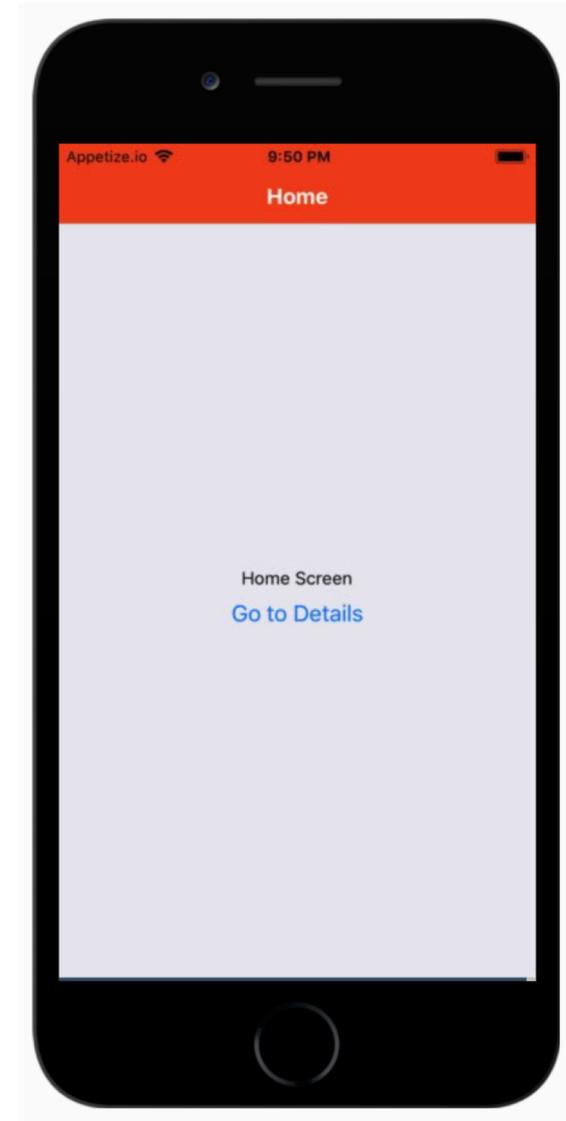
Configuring Headers

Adjusting Header Style

- There are three key properties to use when customizing the style of your header:
 - `headerStyle`,
 - `headerTintColor`, and
 - `headerTitleStyle`.
- **`headerStyle`**: a style object that will be applied to the View that wraps the header. If you set `backgroundColor` on it, that will be the color of your header.
- **`headerTintColor`**: the back button and title both use this property as their color. In the example below, we set the tint color to white (`#fff`) so the back button and the header title would be white.
- **`headerTitleStyle`**: if we want to customize the `fontFamily`, `fontWeight` and other Text style properties for the title, we can use this to do it.

```
class HomeScreen extends React.Component {
  static navigationOptions = {
    title: 'Home',
    headerStyle: {
      backgroundColor: '#f4511e',
    },
    headerTintColor: '#fff',
    headerTitleStyle: {
      fontWeight: 'bold',
    },
  };

  /* render function, etc */
}
```



Notes

- On iOS, the status bar text and icons are black, and this doesn't look great over a dark-colored background
 - configure the status bar to fit with your screen colors [as described in the status bar guide](#).
- **The configuration in the example only applies to the home screen;**
 - when we navigate to the details screen, the default styles are back.
 - Can share navigationOptions between screens. See <https://reactnavigation.org/docs/en/headers.html>

Sharing configurations

- can move the configuration up to the stack navigator.
- any screen that belongs to the RootStack will have our branded styles.
- Can override shared styles

```

class HomeScreen extends React.Component {
  static navigationOptions = {
    title: 'Home',
    /* No more header config here! */
  };
  /* render function, etc */
}
/* other code... */

```

```

const RootStack = createStackNavigator(
  {
    Home: HomeScreen,
    Details: DetailsScreen,
  },
  {
    initialRouteName: 'Home',
    /* The header config from HomeScreen
    is now here */
    navigationOptions: {
      headerStyle: {
        backgroundColor: '#f4511e',
      },
      headerTintColor: '#fff',
      headerTitleStyle: {
        fontWeight: 'bold',
      },
    },
  },
);

```

You might be wondering, why headerTitle when we provide a component and not title, like before? The reason is that headerTitle is a property that is specific to a stack navigator, the headerTitle defaults to a Text component that displays the title.

Overriding default styles

- more control than just changing the text and styles of your title
 - may want to render an image in place of the title,
 - or make the title into a button.
 - In these cases you can completely override the component used for the title and provide your own.

```

class DetailsScreen extends React.Component {
  static navigationOptions = ({ navigation, navigationOptions }) => {
    const { params } = navigation.state;

    return {
      title: params ? params.otherParam : 'A Nested Details Screen',
      /* These values are used instead of the shared configuration! */
      headerStyle: {
        backgroundColor: navigationOptions.headerTintColor,
      },
      headerTintColor: navigationOptions.headerStyle.backgroundColor,
    };
  };

  /* render function, etc */
}

```

Note where the function `navigationOptions` is placed

Must return the config options

This is an example of **overriding** the styles set in the rootstack.

Complete config guide

- <https://reactnavigation.org/docs/en/stack-navigator.html#navigationoptions-used-by-stacknavigator>