

REACT-NATIVE

P3. PRACTISE



STRUCTURE

- Project practise:

<https://github.com/nhancv/nc-rct-demo>

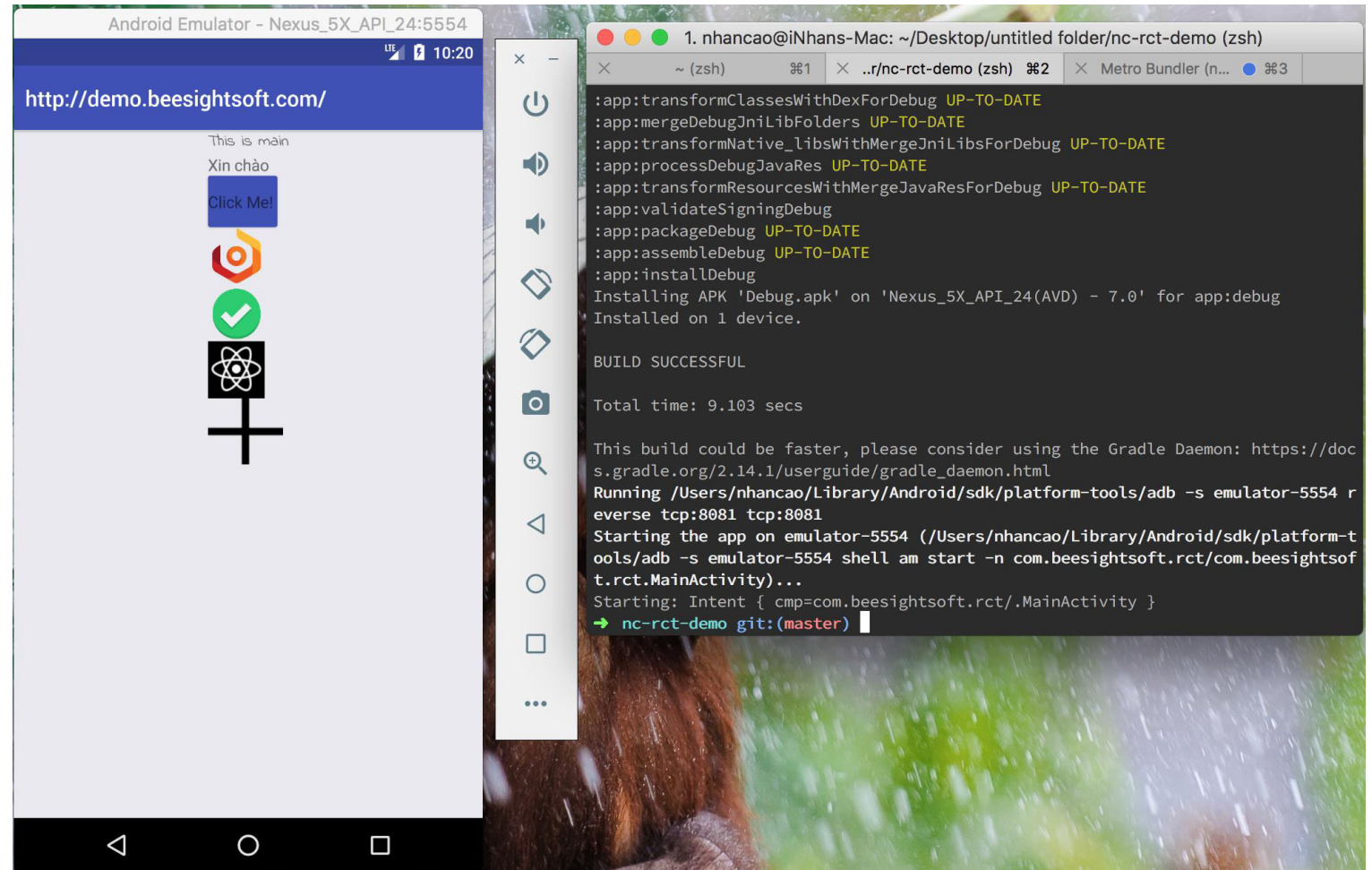
- Try

```
git clone  
https://github.com/nhancv/nc-  
rct-demo.git
```

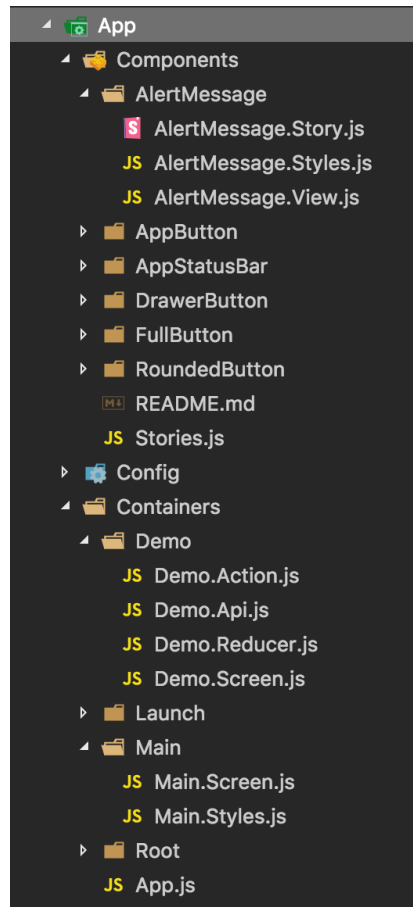
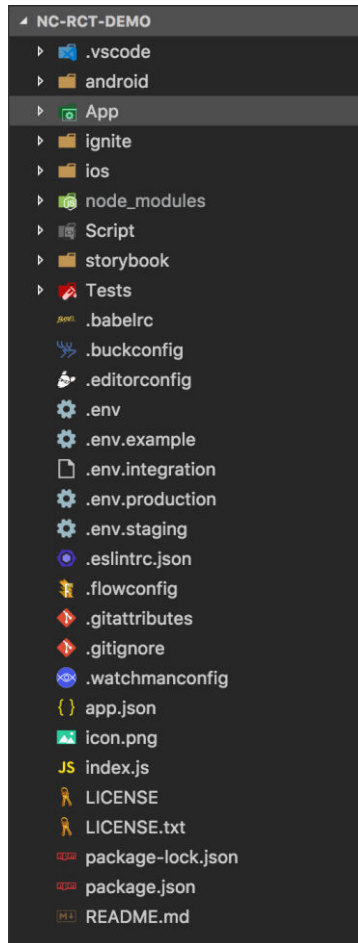
```
cd nc-rct-demo
```

```
npm i
```

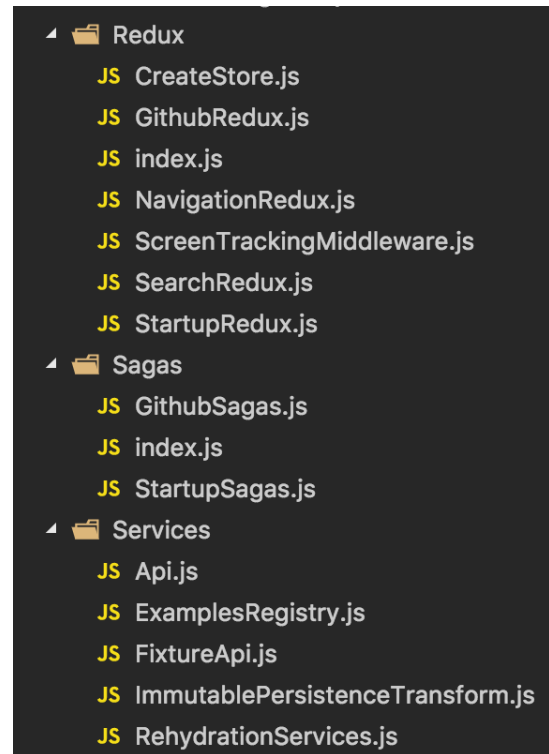
```
react-native run-android
```



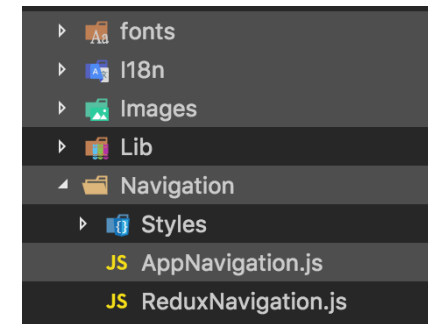
STRUCTURE



App



Redux



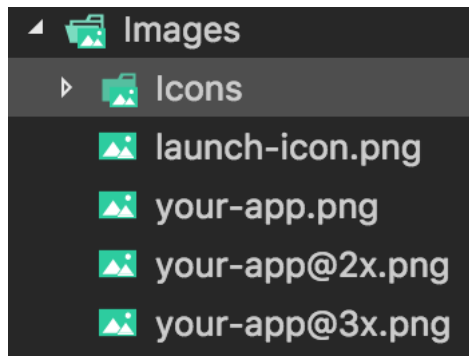
Fonts, i18n, images,
navigation

STYLES

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'blue',
  },
  text: {
    fontSize: 14,
    color: 'red'
  }
});
```

```
class Main extends Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.text}>I am red.</Text>
      </View>
    );
  }
}
```

IMAGES



JS Images.js x

```
1 // leave off @2x/@3x
2 const images = {
3   launch: require('../Images/launch-icon.png'),
4   ready: require('../Images/your-app.png'),
```

```
import { ScrollView, Text, Image, View } from 'react-native'
import { Images } from '../Themes/index'
import SplashScreen from 'react-native-splash-screen'

// Styles
import styles from './Launch.Styles'

export default class LaunchScreen extends Component {
  render() {
    return (
      <View style={styles.mainContainer}>
        <ScrollView style={styles.container}>
          <View style={styles.centered}>
            <Image source={Images.launch} style={styles.logo} />
          </View>
        </ScrollView>
      </View>
    )
  }
}
```

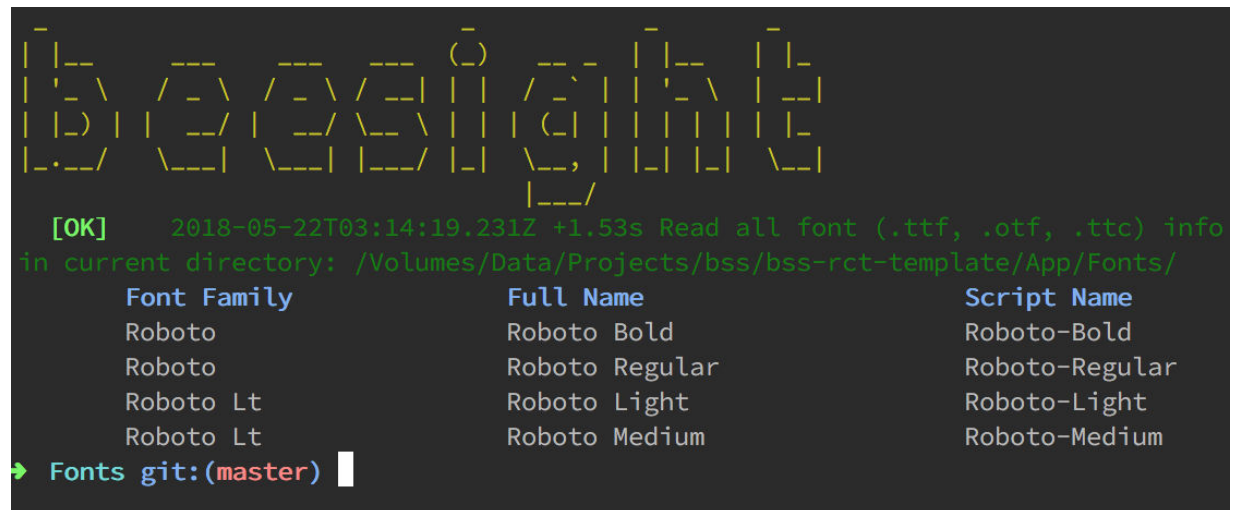
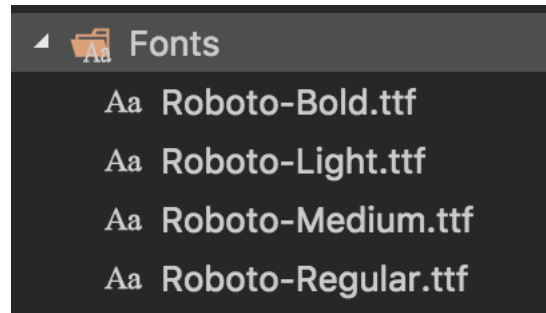
FONTS

Ensure font file name must be matched with Script Name.

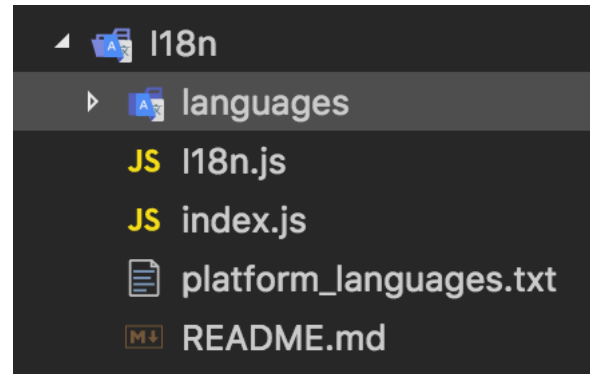
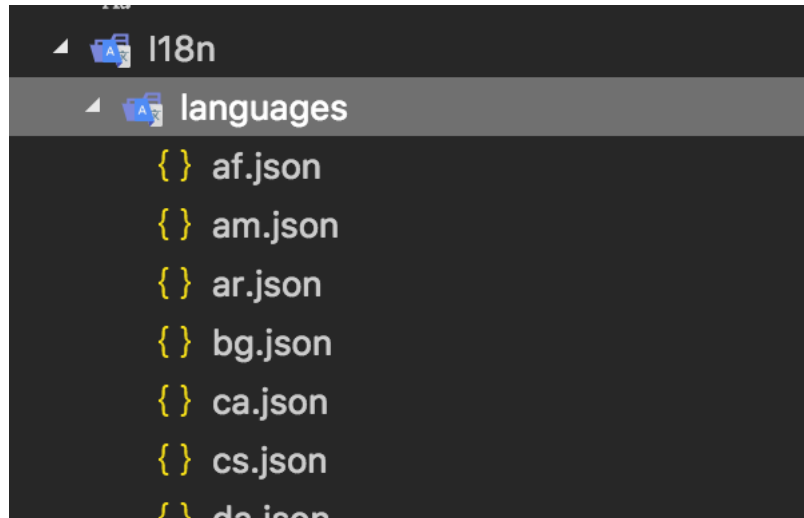
Read font info using: *beesight font*
Android will read from the filename.
iOS will read from the script name property.

- Put fonts to Fonts folder
- run: react-native link

```
<Text style={{
  fontFamily: "Roboto-Light"
}}>This is main</Text>
```

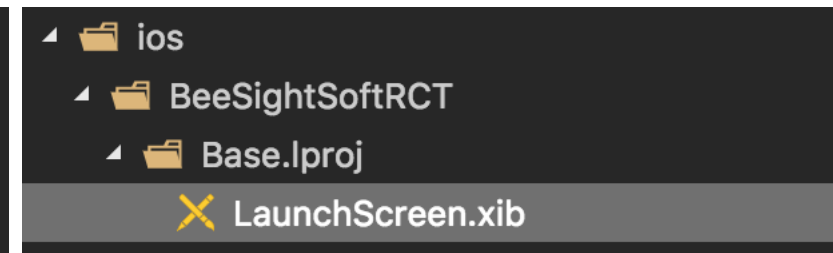
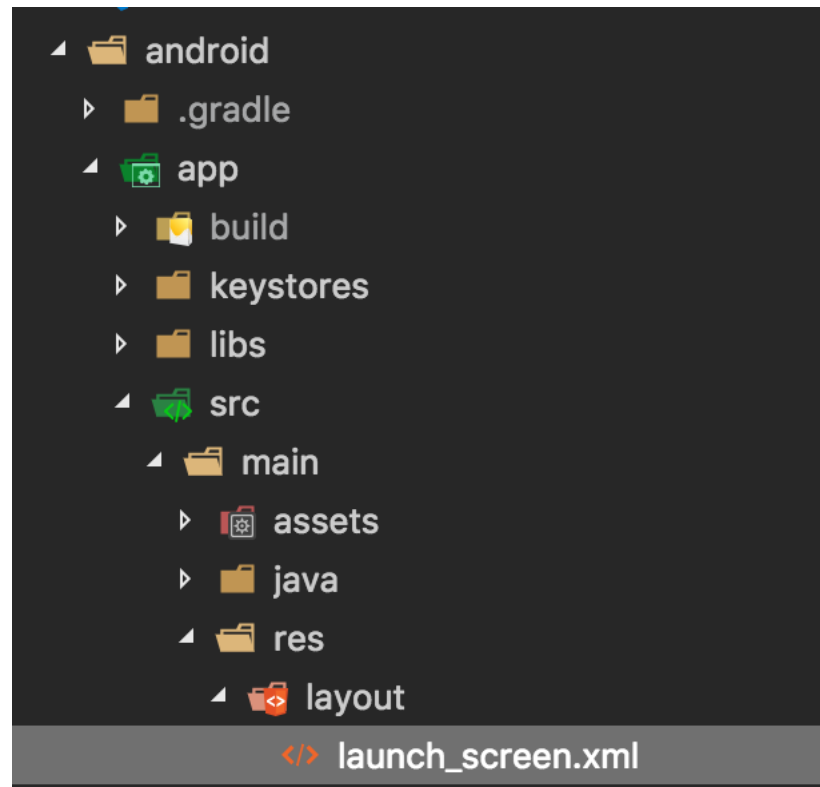


I18N



APP SPLASH SCREEN

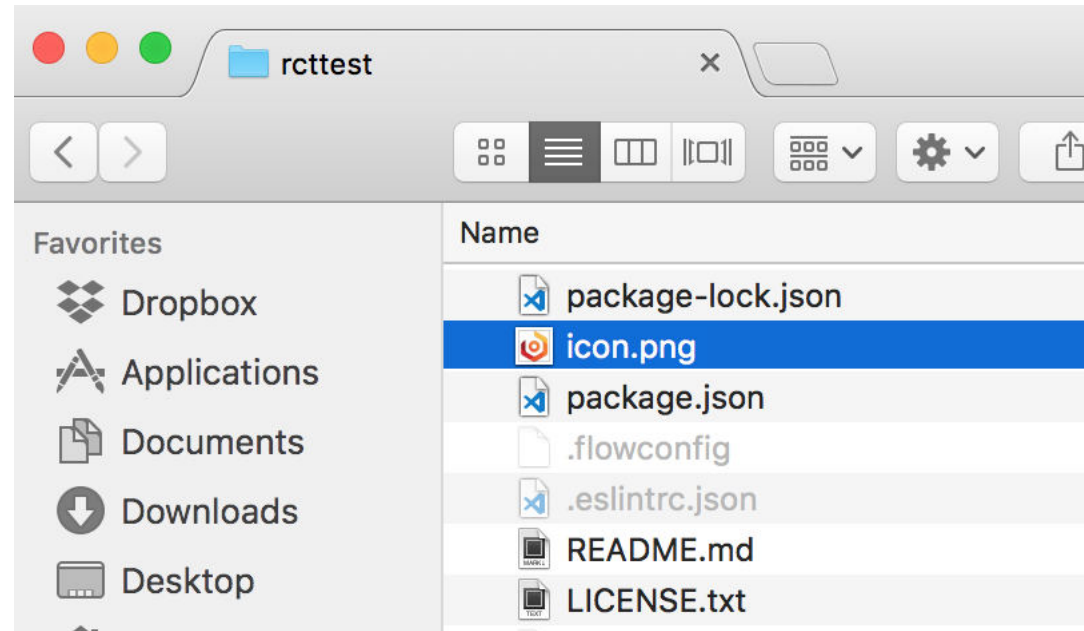
<https://github.com/crazycodeboy/react-native-splash-screen>



APP ICON

<https://github.com/dwmkerr/app-icon>

- Replace with a new single large **icon.png** file, at least **192px square**.
- Run `$ app-icon generate`



APP NAVIGATION

▼ added A App/Containers/Demo/Demo.Screen.js

New file with mode 100644 (Regular)
1 chunk, 13 insertions, 0 deletions

@@ -0,0 +1,13 @@

0	+1	import React, {Component} from 'react'
	+2	import { Dimensions, ScrollView, Text, Image, View, StyleSheet } from 'react-native'
	+3	
	+4	export default class DemoScreen extends Component {
	+5	
	+6	render() {
	+7	return (
	+8	<Text>Demo</Text>
	+9);
	+10	}
	+11	}
	+12	
	+13	

► modified M App/Containers/Main/Main.Screen.js

► modified M App/Navigation/AppNavigation.js

APP NAVIGATION

▶ added A App/Containers/Demo/Demo.Screen.js

▼ modified M App/Containers/Main/Main.Screen.js

1 chunk, 3 insertions, 0 deletions

@@ -23,14 +23,17 @@ export default class MainScreen extends Component {

23	23	style={{
24	24	fontFamily: 'IndieFlower'
25	25	}}
26	26	>
27	27	This is main
28	28	</Text>
29	29	<Text>{I18n.t('welcome')}</Text>
	+30	<Button onPress={() => {
	+31	this.props.navigation.navigate('DemoScreen', {})
	+32	}}><Text>Click Me! </Text></Button>
30	33	<Image style={{ width: 50, height: 50 }} source={require('../Images/lai
31	34	}

APP NAVIGATION

▼ modified M App/Navigation/AppNavigation.js

1 chunk, 3 insertions, 1 deletion

@@ -1,17 +1,19 @@

1	1	import { StackNavigator } from 'react-navigation'
2	2	import LaunchScreen from '../Containers/Launch/Launch.Screen'
3	3	import MainScreen from '../Containers/Main/Main.Screen'
	+4	import DemoScreen from '../Containers/Demo/Demo.Screen'
4	5	
5	6	import styles from './Styles/NavigationStyles'
6	7	
7	8	// Manifest of possible screens
8	9	const PrimaryNav = StackNavigator({
9	10	LaunchScreen: { screen: LaunchScreen },
-10		MainScreen: {screen: MainScreen}
	+11	MainScreen: {screen: MainScreen},
	+12	DemoScreen: {screen: DemoScreen}

PROPS & STATES

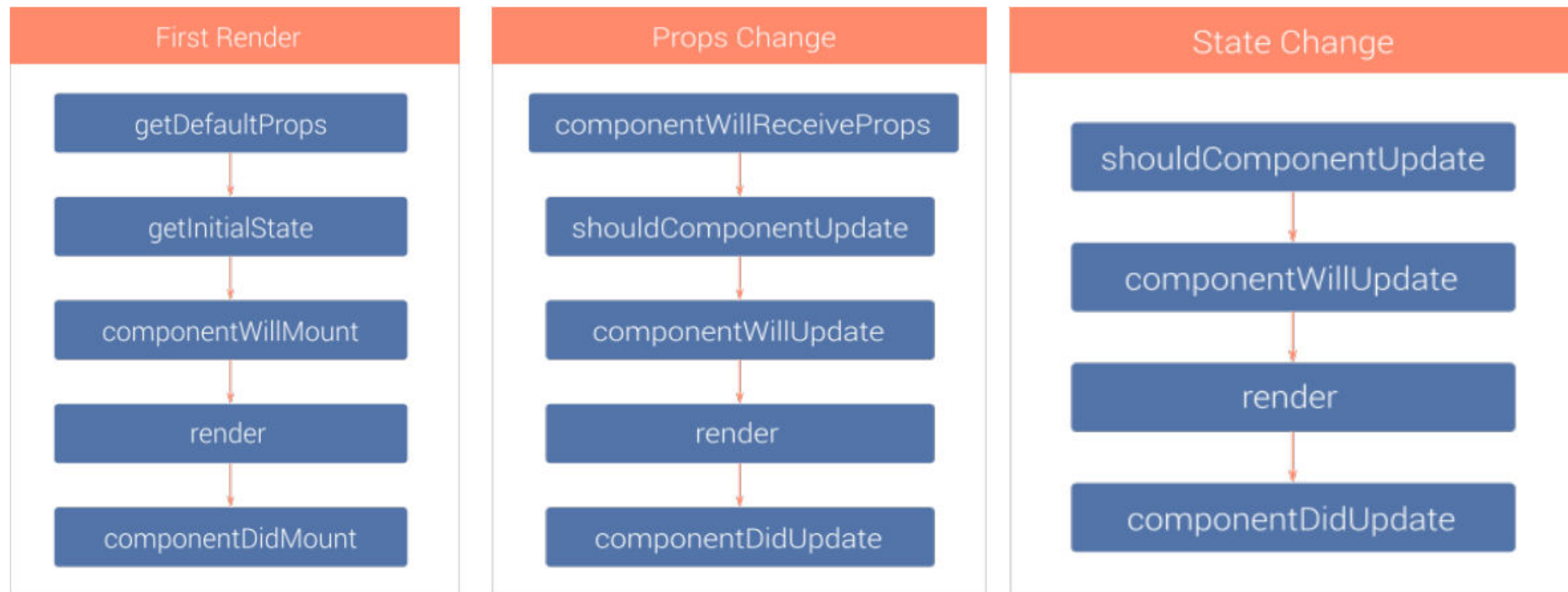
Changing *props* and *state*

	<i>props</i>	<i>state</i>
Can get initial value from parent Component?	Yes	Yes
Can be changed by parent Component?	Yes	No
Can set default values inside Component?*	Yes	Yes
Can change inside Component?	No	Yes
Can set initial value for child Components?	Yes	Yes
Can change in child Components?	Yes	No

- ✓ props are **read-only**
- ✓ props **can not be modified**

- ✓ state changes **can be asynchronous**
- ✓ state **can be modified** using **this.setState**

LIFECYCLE OF COMPONENT

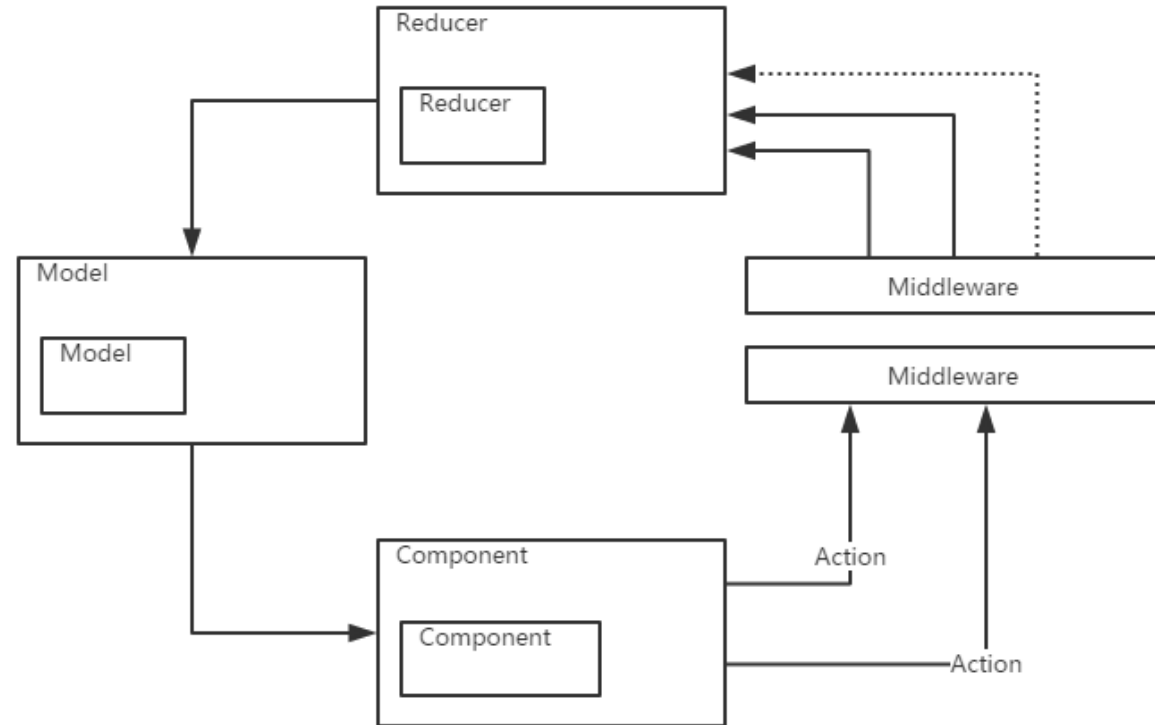


REDUX

Integrate redux
Add 3 file to Demo module
(Containers/ Demo)
```

Demo.Api.js  
Demo.Action.js  
Demo.Reducer.js  
```

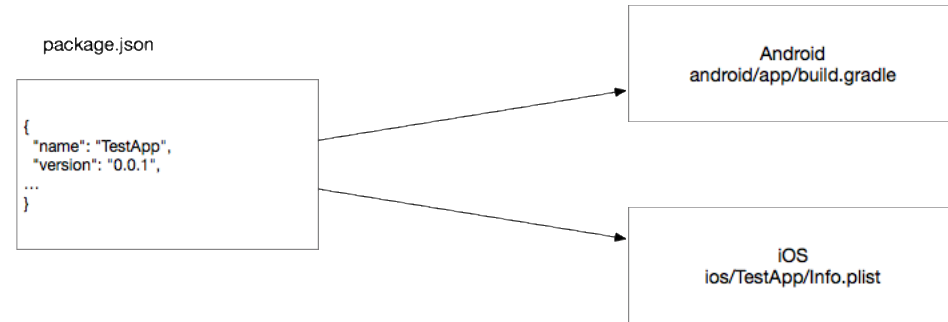
Assemble The Reducers at
(Redux/index.js)
Connect Types To Sagas at
(Sagas/index.js)



<https://github.com/nhancv/nc-rct-demo>

APP VERSION

<https://medium.com/p/94b70da7612f>



```
package.json x
1 {
2   "name": "BeeSightSoftRCT",
3   "version": "0.2.0",
4   "private": true,
```

\$ npm version

```
→ bss-rct-template git:(master) x npm version
{ BeeSightSoftRCT: '0.2.0',
  npm: '6.0.0',
  ares: '1.13.0',
  cldr: '32.0.1',
  http_parser: '2.8.0',
  icu: '60.2',
  modules: '59',
  napi: '2',
  nghttp2: '1.29.0',
  node: '9.10.1',
  openssl: '1.0.2o',
  tz: '2017c',
  unicode: '10.0',
  uv: '1.19.2',
  v8: '6.2.414.46-node.23',
  zlib: '1.2.11' }
```

\$ npm version <new_version>

\$ npm version 0.3.0

APP ENV

<https://github.com/luggit/react-native-config>

Usage

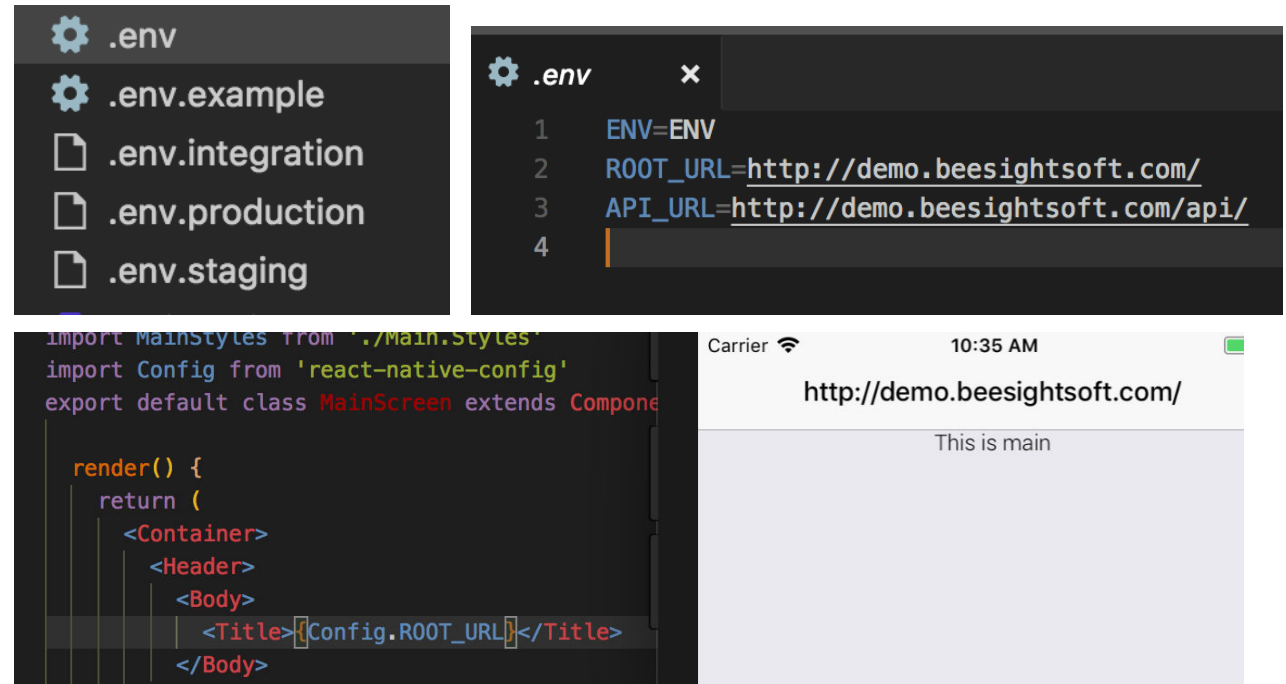
Create a new file `.env` in the root of your React Native app:

```
API_URL=https://myapi.com
GOOGLE_MAPS_API_KEY=abcdefgh
```

Then access variables defined there from your app:

```
import Config from 'react-native-config'

Config.API_URL // 'https://myapi.com'
Config.GOOGLE_MAPS_API_KEY // 'abcdefgh'
```



Android

```
export ENVFILE=.env.${ENVIRONMENT} && react-native run-android
# export ENVFILE=.env.staging && react-native run-android
```

iOS

```
echo ".env.${ENVIRONMENT}" && react-native run-ios
# echo ".env.staging" > /tmp/envfile && react-native run-ios
```

ADD RCT MODULE

```
$ npm install <library-with-native-dependencies> --save
```

Change Podfile to _Podfile

```
$ react-native link
```

Change _Podfile to Podfile

Android already configed.

iOs must be configed manually:

<https://facebook.github.io/react-native/docs/linking-libraries-ios.html>

ADD NATIVE MODULE

Android:

<https://facebook.github.io/react-native/docs/native-modules-android.html>

iOS:

<https://facebook.github.io/react-native/docs/native-modules-ios.html>

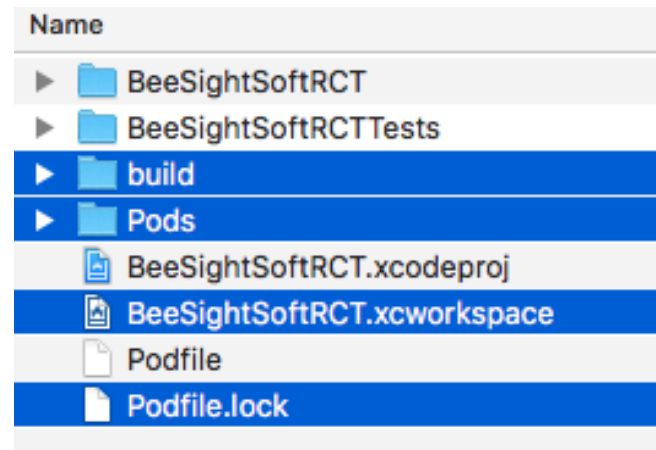
APP CLEAN CACHE

Full command: `rm -rf
$TMPDIR/react-* && rm -rf
$TMPDIR/metro-* && watchman
watch-del-all && rm -rf
ios/build && rm -rf
node_modules/ && npm cache
clean --force && npm i`

Short command:
`npm run newclear`

With iOs:

- Delete ios/build
 - Delete ios/Pods
 - Delete ios/*.xcworkspace
 - Delete ios/Podfile.lock
-
- Pod install again: `pod install`



BUILD RELEASE

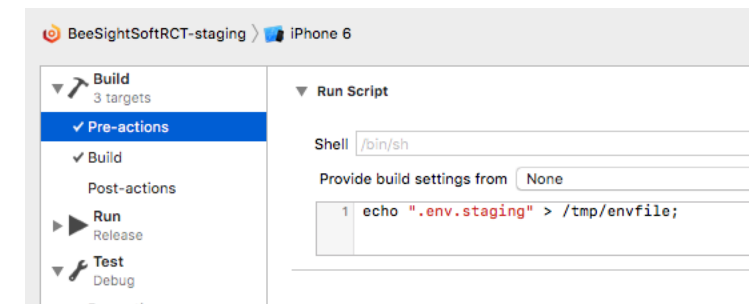
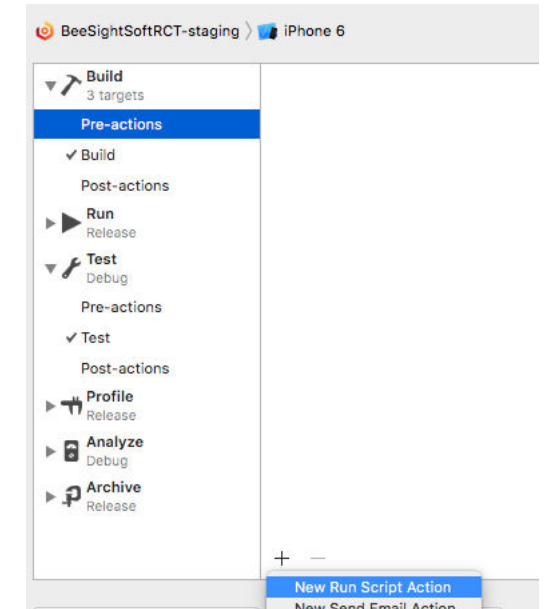
Android

```
mkdir -p
android/app/src/main/assets
rm -rf android/app/build
react-native bundle --platform
android --dev false --entry-file
index.js --bundle-output
android/app/src/main/assets/index.b
undle --assets-dest
android/app/src/main/res
export
ENVFILE=.env.${BUILD_ENVIRONMENT}
cd android && ./gradlew
assembleRelease
```

iOS

- Open .xcworkspace
- Duplicate schema from current app schema (ex: -staging)
- Change "Build Configuration" is "Release"
- Expand "Build" -> Pre-actions -> New "Run Script action"

```
echo ".env.${BUILD_ENVIRONMENT}" >
/tmp/envfile;
```



APP STORE ISSUES

React Native itself is safe. RN apps are all based on JavaScript and work in JavaScriptCore without a browser. However, applications using libraries:

- with the capability to execute external native code may not be compliant with the App Store guidelines. (some libraries for React Native were also considered “insecure”, because they were able to download and execute code which don’t allow for the execution of dynamically loaded native code.)
- has a specific configuration (build setting, signing) difference with configuration of Main App target. And we cannot change its config which added as libraries. Especially, about enable swift while the others just using ObjectC, it make the binary size increase too much and may face with a problem about swift structure libraries when you upload to itunesconnect.
- need setup a podfile correctly whenever install a new libraries.
- setup auto build on ci by Jenkins for development and appstore also.

==> Some problems need concern:

- + Signing
- + Dont have a unusual library in project.
- + Config libraries correctly

REFERENCES

- Example: <https://github.com/nhancv/nc-rct-demo>
- Home page: <https://facebook.github.io/react-native/>
- CLI for react native: <https://github.com/infinitered/ignite>
- Ui framework:
 - <https://nativebase.io/>
 - <https://akveo.github.io/react-native-ui-kitten>
- Awesome react native: <http://www.awesome-react-native.com/>
- Package: <https://devarchy.com/react>
- Fonts: <https://medium.com/p/ccc9aacf9e5e>
- Reference: <https://unbug.gitbooks.io/react-native-training/content/>

THANK YOU.