1. Create the project
2. Create an app folder
3. Inside the app folder, create the following folder structure.
   1. assets
      1. All images, icons and other app needed files
   2. config
      1. appsettings.js – Define all your API urls here
      2. routes.js – Define all your scenes and routes
   3. modules
      1. YourModuleName
         1. Components – Define all your reusable module components here
            1. YourComponentName

Index.js

Styles.js

* + - 1. Scenes – Define all your scenes/pages here
         1. YourSceneName

index.js

* + - 1. actions.js – Define all your actions that will dispatch to the reducers
      2. constants.js – Define all constants that will be used by actions/reducers
      3. index.js – Export all actions, constants, reducers and theme files
      4. reducer.js – Define your reducer (one or many) with their respective states that will catch the actions
  1. redux
     1. rootReducer.js – Define all your reducers in the application and combine them all
     2. store.js – Where you create the Store and apply all your middleware
  2. styles
     1. theme.js – Define your application styling

**COMMON ISSUES AND DEBUGGING**

1)Unable to resolve module …. Does not exist in the haste module map.

Fixes: try npm install and then react-native link

Then if the issue persists on iOS, without trying to run from xCode directly, from the command line, type npm run ios and then it should build and run. Thereafter you can run on xCode like before

2) If you get an error saying... Unfortunately <APP\_NAME> has crashed Generate the logs for that session from Genymotion and search for keyword – FATAL EXCEPTION and see what the cause is..

If it’s an error related to URL, saying cannot find localhost or something, delete the APK and try running the application again

**DOING A DEPLOYMENT IN iOS (XCODE)**

1. Change the build and version number accordingly of your application and click on Archive from Product option on top menu
2. After app is archived, a window will appear, click Distribute app
3. Select iOS App Store -> Export -> Manage signings auto/manually -> And then click Export
4. Go to xCode
5. On top menu bar select Xcode and go to open developer tools and choose Application Loader
6. Choose the exported IPA file and click Next. Wait till the process is completed.
7. Wait for a few mins until the application gets approved by Apple
8. Go to itunes connect and select My Apps
9. Go to Test Flight tab and you should see your App with the build number under the Builds tab on iOS (Processing/Missing Compliance)
10. If Missing Compliance is shown, click on that and provide compliance information
11. And then submit it.
12. Add external testers to your build if you wish.

**DOING A DEPLOYMENT IN ANDROID**

**DEBUGGING LOCALLY IN WINDOWS MACHINE**

1. Cmd => ipconfig
2. Record ip address
3. Project => .vs folder => config => applicationhost
4. Find the relative site. Replace localhost with ip address under binding information (\*:<portnumber>:localhost) with \*:<portnumber>:<ipaddress>
5. Run cmd with admin

netsh advfirewall firewall add rule name="IISExpressWeb" dir=in protocol=tcp localport=<portnumber> profile=public remoteip=localsubnet action=allow

1. Run project visual studio (admin)
2. Replace swagger with <ipaddress>:<portnumber>

https://johan.driessen.se/posts/Accessing-an-IIS-Express-site-from-a-remote-computer