

Prerequisites :

- Open Xcode
- Open Android studio
- Open vscode with slides
- Open new Vscode in the correct Folder from finder .
- Open two browser
- Make dir in MONOREPO called rnmonorepo with packages/shared/src/* files and open it in vs-code.
- Done *react-native init mobile* inside *packages* and ran: `rm -rf node_modules/ yarn.lock`
- Delete *Pods* and *Podfile.lock* *
- Clear Developer cache for ios for mobile app.

[Presentation: 01, 02, 03]

Step by Step process:

A. React Native Apps

1. enable workspaces for whole projects with packages and install dependencies as react-native (make sure it is the same version as in mobile)

```
"workspaces": {  
  "packages": [  
    "packages/*"  
  ],  
  "dependencies": {  
    "react-native": "0.61.5"  
  }  
}
```

2. install dependencies... from root folder
`yarn.` [Presentation: 04, 05]

3. install dependencies... from root folder
find and replace
`node_modules/react-native => ../../node_modules/react-native/`

4. Open `packages./mobile/metro.config.js` and set the `projectRoot` field on it as well so it looks like this:

```
const path = require('path')  
  
module.exports = {  
  projectRoot: path.resolve(__dirname, '../..'),  
  transformer: {
```

```
getTransformOptions: async () => ({
  transform: {
    experimentalImportSupport: false,
    inlineRequires: false,
  },
}),
},
}
```

5. For iOS:

- a. In Podfile , change the require_relative to be

```
require_relative '../../../../../node_modules/@react-native-community/cli-
platform-ios/native_modules'
```

- b. POD INSTALL: `yarn mobile:pod`s

- c. Open xcworkspace: `open packages/mobile/ios/mobile.xcworkspace`

- d. Open AppDelegate.m,
replace `index` with `packages/mobile/index`

- e. Click project-name on the left, and go to Build Phases > Bundle React Native code and Images. Replace its content with this:

```
export NODE_BINARY=node
export EXTRA_PACKAGER_ARGS="--entry-file packages/mobile/index.js"
../../../../node_modules/react-native/scripts/react-native-xcode.sh
```

- f. Run the metro bundler in a new TERMINAL: `yarn workspace mobile start`

- g. Run the app in the simulator from xCode 🍷 [Presentation: 06]

6. For Android:

- a. Open android studio: `studio packages/mobile/android/`

- b. Change settings.gradle

```
apply from: file("../../../../../node_modules/@react-native-community/cli-
platform-android/native_modules.gradle");
applyNativeModulesSettingsGradle(settings)
```

- c. In app/build.gradle:

```
project.ext.react = [
  entryFile: "packages/mobile/index.js",
  root: "../../../../",
```

```
enableHermes: true, // **imp clean and rebuild if changing  
]
```

... (inside if enableHermes)

```
def hermesPath = "../../../../../node_modules/hermes-engine/android/";
```

... (last line)

```
apply from: file("../../../../../node_modules/@react-native-community/cli-  
platform-android/native_modules.gradle");  
applyNativeModulesAppBuildGradle(project)
```

- d. In MainApplication.java, change the getJSMainModuleName() method to return **"packages/mobile/index"** not just **"index"**
 - e. Then run android app in android simulator. 🙏 [Presentation: 07, 07.5]
7. To show hot reloading, change App.js in /packages/mobile to have in line: 42

```
<View style={{ backgroundColor: 'red', padding: 16 }}>  
  <Text style={{ fontSize: 20, color: 'white' }}>  
    React monorepo rocks !  
  </Text>  
</View>
```

B. React-native-web Apps

1. `cd packages`
2. create a reactjs project : `npx create-react-app web` [Presentation: 08]
3. inside web folder : `cd web`
4. changed the version of react to be same as others : `"react": "16.9.0"`
5. add shared as dependency: `"@monorepo/shared": "1.0.0"`
6. `yarn add react-native-web react-art` [do steps 7, 8 , 9, 10, 11]
7. remove everything from src: `rm src/*`
8. create index.js inside packages/web/src :

```
import { AppRegistry } from 'react-native'
```

```
import App from '@monorepo/shared/src/App'

AppRegistry.registerComponent('web', () => App)
AppRegistry.runApplication('web', {
  rootTag: document.getElementById('root'),
})
```

9. create an index.css inside packages/web/public/index.css with:

```
html,
body,
#root,
#root > div {
  width: 100%;
  height: 100%;
}

body {
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
```

10. include the css in index.html :

```
<link rel="stylesheet" href="%PUBLIC_URL%/index.css" />
```

11. create .env with :

```
SKIP_PREFLIGHT_CHECK=true
```

12. install dev dependencies for rewired : `yarn add react-app-rewired babel-plugin-react-native-web --dev`

[Presentation: 09]

13. change the scripts for react-web in package.json to be:

```
"scripts": {
  "start": "react-app-rewired start",
  "build": "react-app-rewired build",
  "test": "react-app-rewired test",
  "eject": "react-app-rewired eject"
},
```

14. make config-overrides.js for react-app-rewired to run with babel transform:

```
const fs = require('fs')
const path = require('path')
const webpack = require('webpack')

const appDirectory = fs.realpathSync(process.cwd())
const resolveApp = relativePath => path.resolve(appDirectory, relativePath)

// our packages that will now be included in the CRA build step
```

```
const appIncludes = [
  resolveApp('src'),
  resolveApp('../shared'),
]

module.exports = function override(config, env) {
  // allow importing from outside of src folder
  config.resolve.plugins = config.resolve.plugins.filter(
    plugin => plugin.constructor.name !== 'ModuleScopePlugin'
  )
  config.module.rules[0].include = appIncludes
  config.module.rules[1] = null
  config.module.rules[2].oneOf[1].include = appIncludes
  config.module.rules[2].oneOf[1].options.plugins = [
    require.resolve('babel-plugin-react-native-web'),
  ].concat(config.module.rules[2].oneOf[1].options.plugins)
  config.module.rules = config.module.rules.filter(Boolean)
  config.plugins.push(
    new webpack.DefinePlugin({ __DEV__: env !== 'production' })
  )

  return config
}
```

15. *yarn workspace web start*

[Presentation: 10, 11, 12]

C. React-Js Apps

1. create a reactjs project : `npx create-react-app react-web`
2. changed the version of react to be same as others : `"react": "16.9.0"`
3. add shared as dependency: `"@monorepo/shared": "1.0.0"`
4. install dev dependencies for babel : `npm i @babel/core babel-preset-react babel-loader @babel/preset-env @babel/preset-react --save-dev`
5. install dev dependencies for rewired and transform: `yarn add react-app-rewired @babel/plugin-transform-react-jsx --save-dev`
6. change the scripts for react-web in package.json to be:

```
"scripts": {
  "start": "react-app-rewired start",
  "build": "react-app-rewired build",
  "test": "react-app-rewired test",
  "eject": "react-app-rewired eject"
},
```

7. make config-overrides.js for react-app-rewired to run with babel transform:

```
const fs = require('fs')
const path = require('path')
const webpack = require('webpack')

const appDirectory = fs.realpathSync(process.cwd())
const resolveApp = relativePath => path.resolve(appDirectory, relativePath)

// our packages that will now be included in the CRA build step
const appIncludes = [
  resolveApp('src'),
  resolveApp('..shared'),
]

module.exports = function override(config, env) {
  // allow importing from outside of src folder
  config.resolve.plugins = config.resolve.plugins.filter(
    plugin => plugin.constructor.name !== 'ModuleScopePlugin'
  )
  config.module.rules[0].include = appIncludes
  config.module.rules[1] = null
  config.module.rules[2].oneOf[1].include = appIncludes
  config.module.rules[2].oneOf[1].options.plugins = [
    require.resolve('@babel/plugin-transform-react-jsx'),
  ].concat(config.module.rules[2].oneOf[1].options.plugins)
  config.module.rules = config.module.rules.filter(Boolean)
  config.plugins.push(
    new webpack.DefinePlugin({ __DEV__: env !== 'production' })
  )

  return config
}
```

8. don't forget to add jest for rewired: `yarn add jest`
9. `yarn workspace react-web start`