**React-Native**

**Multiple platform support**

1. Create a react native project:

npx react-native init basic

NOTE: React native macOS supports RN 0.71 right now.

1. **For web** **support,** run these commands:
   * npm install react-dom react-native-web --legacy-peer-deps
   * npm install --save-dev babel-plugin-react-native-web babel-plugin-module-resolver --legacy-peer-deps
   * npm i -D babel-plugin-react-native-web webpack webpack-cli webpack-dev-server html-webpack-plugin react-dom babel-loader url-loader @svgr/webpack --legacy-peer-deps
   * Add these lines into your package.json.

"jest": {

"preset": "react-native-web",

"moduleFileExtensions": [

"ts",

"tsx",

"js",

"jsx",

"json",

"node"

]

}

* + Include these two scripts into package.json.

"build": "rm -rf dist/ && webpack --mode=production --config webpack.config.js",

"web": "webpack serve --mode=development --config webpack.config.js"

* + Add 4 files into your project root folder.

**App.web.tsx**

|  |  |
| --- | --- |
| import React, {useState} from 'react'; |  |
|  | import {View, Text, TouchableOpacity, StyleSheet} from 'react-native'; |
|  |  |
|  | const App = () => { |
|  | const [count, setCount] = useState(0); |
|  | return ( |
|  | <View style={styles.container}> |
|  | <Text style={styles.title}>Hello from {'\n'}React Native Web!</Text> |
|  | <TouchableOpacity |
|  | onPress={() => setCount(count + 1)} |
|  | style={styles.button}> |
|  | <Text>Click me!</Text> |
|  | </TouchableOpacity> |
|  |  |
|  | <Text>You clicked {count} times!</Text> |
|  | </View> |
|  | ); |
|  | }; |
|  |  |
|  | const styles = StyleSheet.create({ |
|  | container: { |
|  | flex: 1, |
|  | backgroundColor: '#C3E8BD', |
|  | paddingTop: 40, |
|  | paddingHorizontal: 10, |
|  | }, |
|  | button: { |
|  | backgroundColor: '#ADBDFF', |
|  | padding: 5, |
|  | marginVertical: 20, |
|  | alignSelf: 'flex-start', |
|  | }, |
|  | title: { |
|  | fontSize: 40, |
|  | }, |
|  | }); |
|  |  |
|  | export default App; |

**index.html**

|  |  |
| --- | --- |
| <!DOCTYPE html> |  |
|  | <html> |
|  | <head> |
|  | <meta charset="UTF-8" /> |
|  | <meta name="viewport" content="width=device-width, initial-scale=1.0" /> |
|  | <meta http-equiv="X-UA-Compatible" content="ie=edge" /> |
|  | <title>RN Web</title> |
|  | <style> |
|  | #app-root { |
|  | display: flex; |
|  | flex: 1 1 100%; |
|  | height: 100vh; |
|  | } |
|  | </style> |
|  | </head> |
|  | <body> |
|  | <div id="app-root"></div> |
|  | </body> |
|  | </html> |

**index.web.js**

|  |  |
| --- | --- |
| import {AppRegistry} from 'react-native'; |  |
|  | import {name as appName} from './app.json'; |
|  | import App from './App'; |
|  | if (module.hot) { |
|  | module.hot.accept(); |
|  | } |
|  | AppRegistry.registerComponent(appName, () => App); |
|  | AppRegistry.runApplication(appName, { |
|  | initialProps: {}, |
|  | rootTag: document.getElementById('app-root'), |
|  | }); |

**webpack.config.js**

|  |  |
| --- | --- |
| const path = require('path'); |  |
|  |  |
|  | const webpack = require('webpack'); |
|  | const HtmlWebpackPlugin = require('html-webpack-plugin'); |
|  |  |
|  | const appDirectory = path.resolve(\_\_dirname); |
|  | const {presets} = require(`${appDirectory}/babel.config.js`); |
|  |  |
|  | const compileNodeModules = [ |
|  | // Add every react-native package that needs compiling |
|  | // 'react-native-gesture-handler', |
|  | ].map((moduleName) => path.resolve(appDirectory, `node\_modules/${moduleName}`)); |
|  |  |
|  | const babelLoaderConfiguration = { |
|  | test: /\.js$|tsx?$/, |
|  | // Add every directory that needs to be compiled by Babel during the build. |
|  | include: [ |
|  | path.resolve(\_\_dirname, 'index.web.js'), // Entry to your application |
|  | path.resolve(\_\_dirname, 'App.web.tsx'), // Change this to your main App file |
|  | path.resolve(\_\_dirname, 'src'), |
|  | ...compileNodeModules, |
|  | ], |
|  | use: { |
|  | loader: 'babel-loader', |
|  | options: { |
|  | cacheDirectory: true, |
|  | presets, |
|  | plugins: ['react-native-web'], |
|  | }, |
|  | }, |
|  | }; |
|  |  |
|  | const svgLoaderConfiguration = { |
|  | test: /\.svg$/, |
|  | use: [ |
|  | { |
|  | loader: '@svgr/webpack', |
|  | }, |
|  | ], |
|  | }; |
|  |  |
|  | const imageLoaderConfiguration = { |
|  | test: /\.(gif|jpe?g|png)$/, |
|  | use: { |
|  | loader: 'url-loader', |
|  | options: { |
|  | name: '[name].[ext]', |
|  | }, |
|  | }, |
|  | }; |
|  |  |
|  | module.exports = { |
|  | entry: { |
|  | app: path.join(\_\_dirname, 'index.web.js'), |
|  | }, |
|  | output: { |
|  | path: path.resolve(appDirectory, 'dist'), |
|  | publicPath: '/', |
|  | filename: 'rnw\_blogpost.bundle.js', |
|  | }, |
|  | resolve: { |
|  | extensions: ['.web.tsx', '.web.ts', '.tsx', '.ts', '.web.js', '.js'], |
|  | alias: { |
|  | 'react-native$': 'react-native-web', |
|  | }, |
|  | }, |
|  | module: { |
|  | rules: [ |
|  | babelLoaderConfiguration, |
|  | imageLoaderConfiguration, |
|  | svgLoaderConfiguration, |
|  | ], |
|  | }, |
|  | plugins: [ |
|  | new HtmlWebpackPlugin({ |
|  | template: path.join(\_\_dirname, 'index.html'), |
|  | }), |
|  | new webpack.HotModuleReplacementPlugin(), |
|  | new webpack.DefinePlugin({ |
|  | // See: <https://github.com/necolas/react-native-web/issues/349> |
|  | \_\_DEV\_\_: JSON.stringify(true), |
|  | }), |
|  | ], |
|  | }; |

1. **For macOS & Windows support:**

Follow https://microsoft.github.io/react-native-windows/docs/rnm-getting-started