**Mapbox with PCF component Framework**

**1. PowerApps Component Framework Installation Steps**

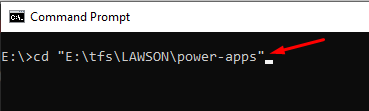
**Step 1.1**

Open a Command Prompt window.

**Step 1.2**

Go to your project folder.

cd <Your Project Folder>



**Step 1.3**

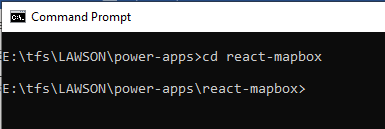
Create a new folder for the project using the following command:​

* mkdir react-mapbox

**Step 1.4**

Go into the component folder using the command:​

* cd react-mapbox

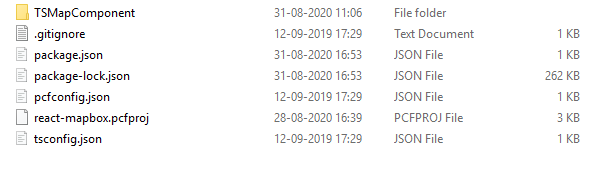


**Step 1.5**

Create a new component project by passing basic parameters using the command:​

* pac pcf init --namespace SampleNamespace --name TSMapComponent --template field

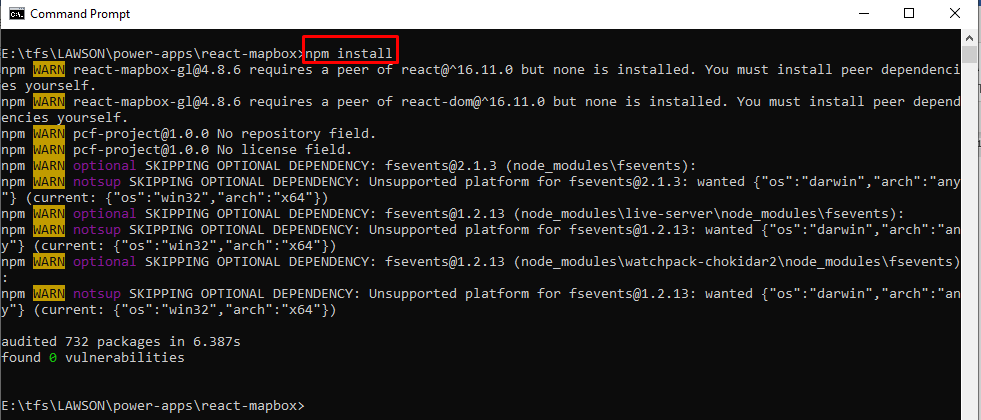
This installs the files into the directory as shown below:



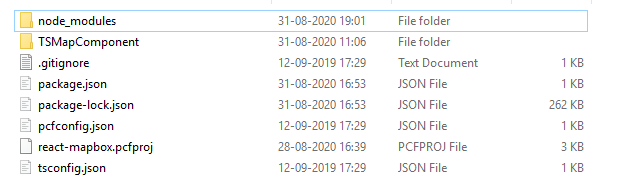
**Step 1.6**

Install the project build tools using the command:​

* npm install



We now have the node modules directory created with the dependencies:​

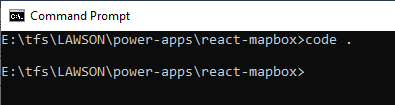


**Step 1.7**

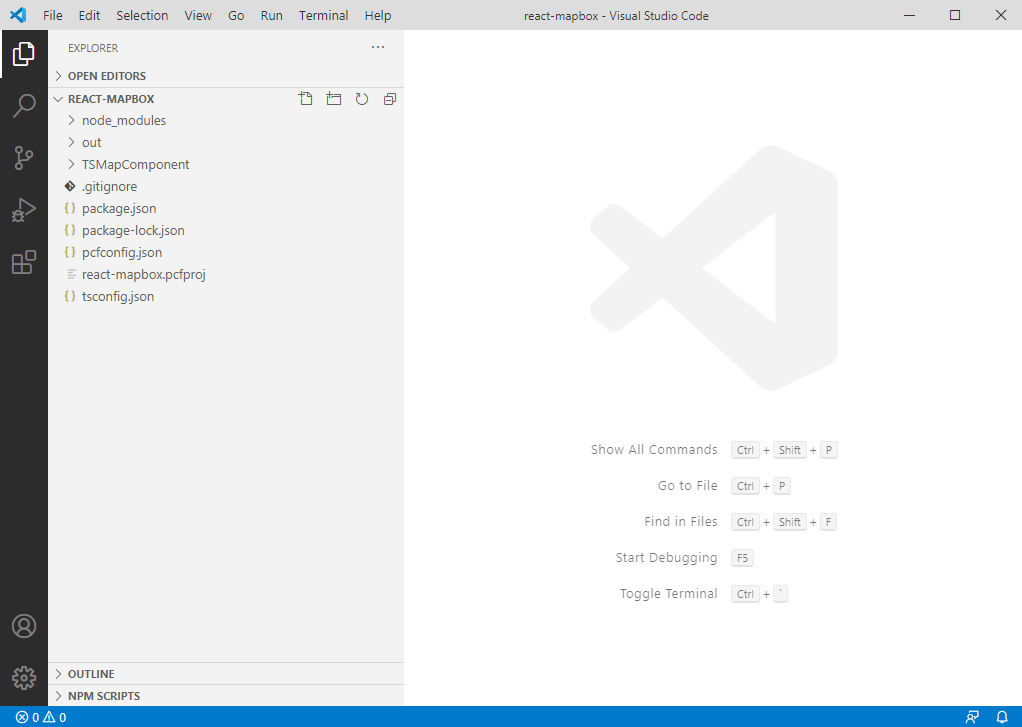
Open your project folder in a developer environment of your choice and get started with your code component development.​

To look at it, we let’s open Visual Studio Code.

There’s a couple of easy ways to do this, either type **code .** into the command prompt from the directory you want to open:​

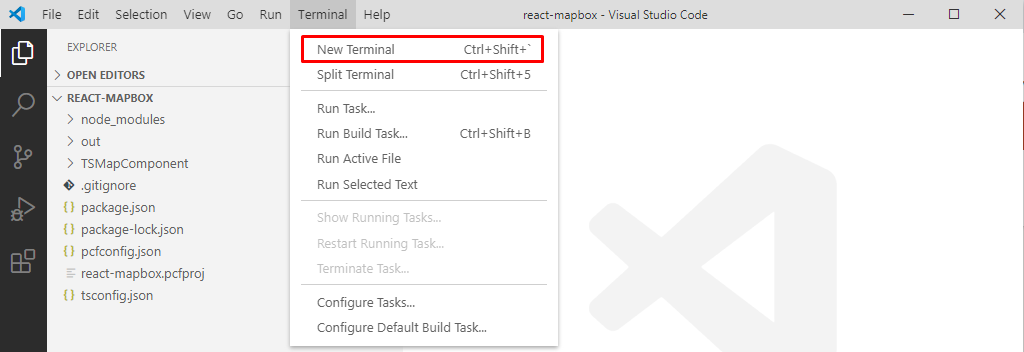


After the **code .** Command the project will open in the visual studio code as shown below.

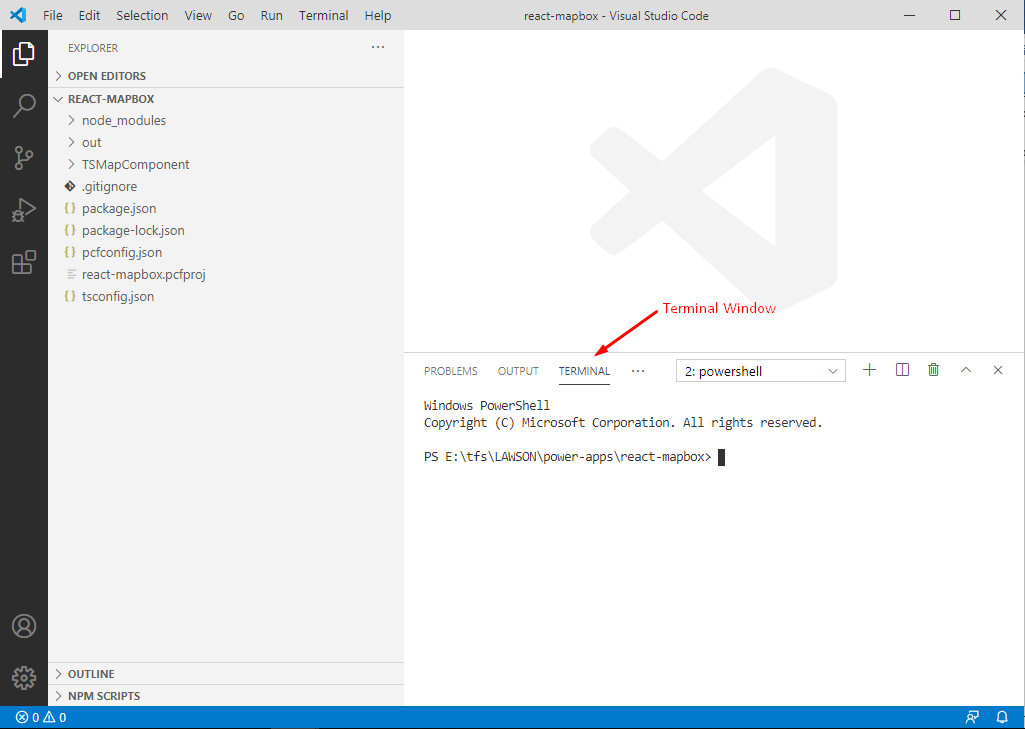


**Step 1.8**

Click on Terminal -> New Terminal



After clicking, it opens the below window.

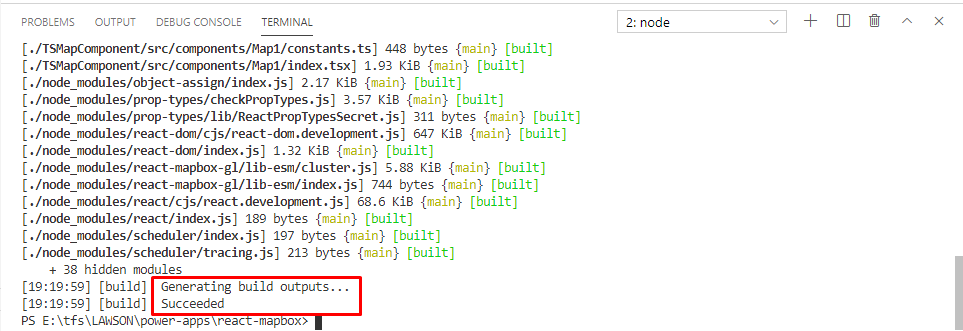


**Step 1.9**

Run the below command for build & build should got Succeeded.

* npm run build

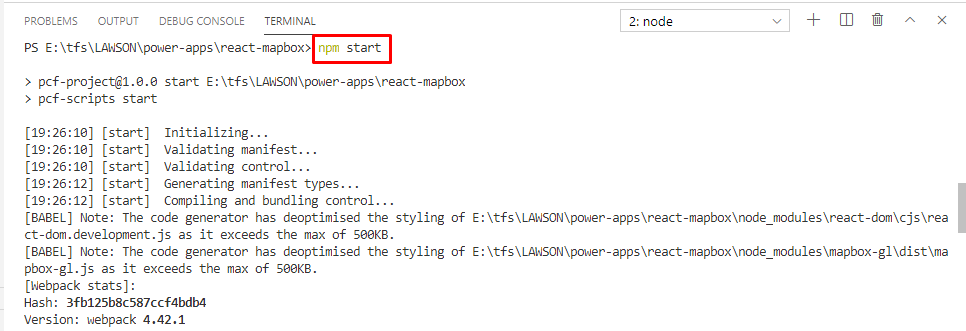


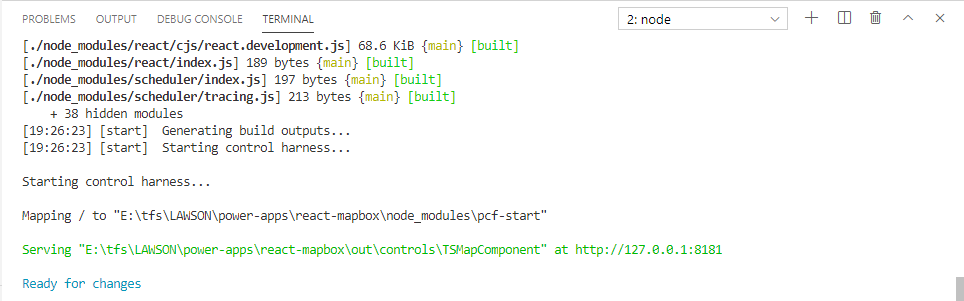


**Step 1.10**

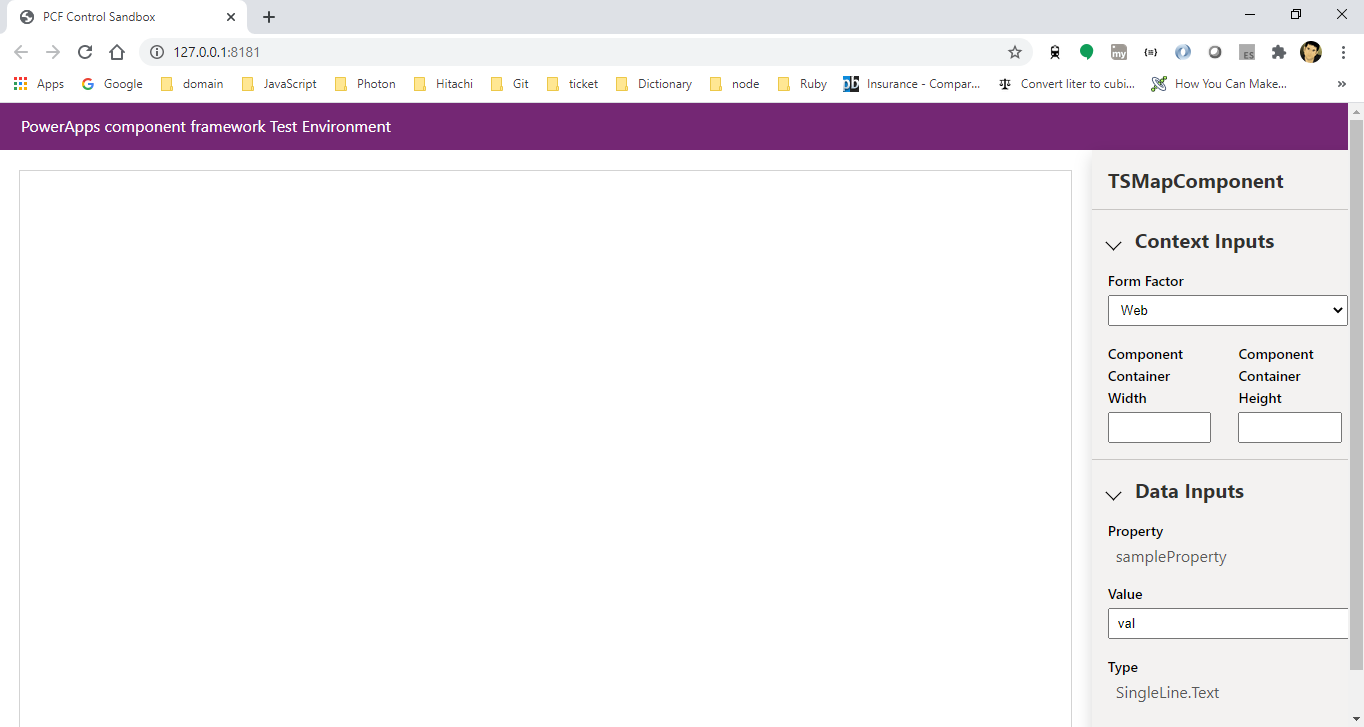
Then run the below command and task get executed on your system as shown below.

* npm start





This opens a web browser to a local IP address, in my case [**http://127.0.0.1:8181/**](http://127.0.0.1:8181/)



**2. Start creating Map Component**

**Step 2.1**

Now we have our basic project set up. Let’s overwrite the original code with our custom component code.

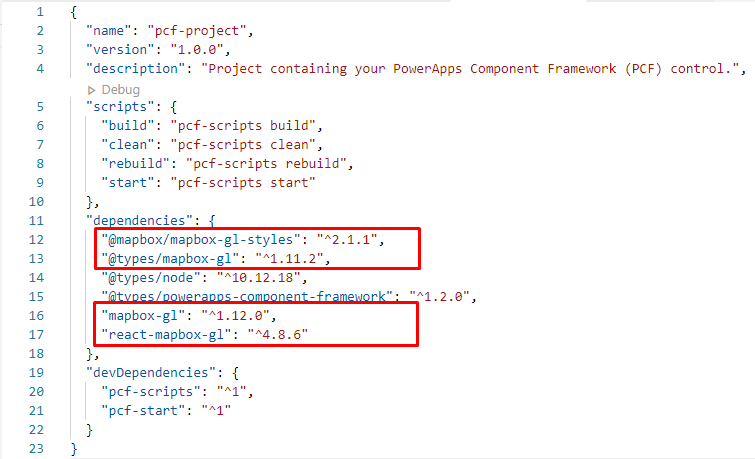
Overwrite the ControlManifest.Input.xml file with: [**Map Component Manifest**](https://docs.microsoft.com/en-us/powerapps/developer/component-framework/sample-controls/map-control)



**Step 2.2**

Install all the dependencies mentioned below.

* npm install --save mapbox-gl
* npm install --save react-mapbox-gl
* npm install --save @types/mapbox-gl
* npm install --save @mapbox/mapbox-gl-styles



**Step 2.3**

Create *TSMapComponent/index.css*

html {

    height: 100%;

}

body {

  margin: 0;

  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

    sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

  height: 100%;

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

div#root {

    height: 100%;

}

.TSMapComponent {

  height: 600px;

}

**Step 2.4**

Create *TSMapComponent/src/App.tsx*

import \* as React from 'react';

import ReactMapboxGl, { Layer, Feature, ZoomControl  } from 'react-mapbox-gl';

const Mapbox = ReactMapboxGl({

  accessToken: 'pk.eyJ1IjoibnVzcmF0aGphaGFuIiwiYSI6ImNrNzAwcHgwcDBkc2ozZXI0dGdtMDdyaWgifQ.\_w76wkW285SYjL4tw356gA',

});

export interface AppProps {

}

const App: React.FC<AppProps> = (props) => {

  const center: [number, number] = [-0.181747846041145, 51.5133379650232];

  return (

    <div className="App">

        <Mapbox

          style="mapbox://styles/mapbox/streets-v9"

          center={center}

          containerStyle={{

            height: '100%',

            width: '100%'

          }}

        >

            <ZoomControl></ZoomControl>

            <Layer id="symbolLayer" type="symbol" layout={{

                'icon-image': 'harbor-15',

            }}>

                <Feature coordinates={center} />

            </Layer>

        </Mapbox>

    </div>

  );

}

export default App;

**Step 2.4**

Create *TSMapComponent/src/App.css*

.App {

    height: 100%;

    position: relative;

}

**Step 2.5**

Add all CSS files into *ControlManifest.Input.xml* file



**Step 2.6**

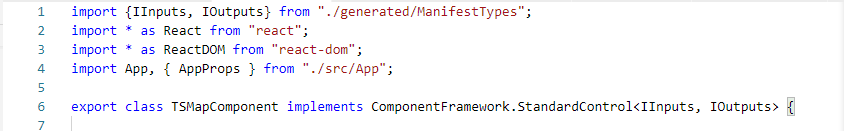
Overwrite the *TSMapComponent/index.ts* file with our custom react map component *TSMapComponent/src/App.tsx* file.

**Step 2.6.1** Import all the dependencies mentioned as below.

import \* as React from "react";

import \* as ReactDOM from "react-dom";

import App, { AppProps } from "./src/App";

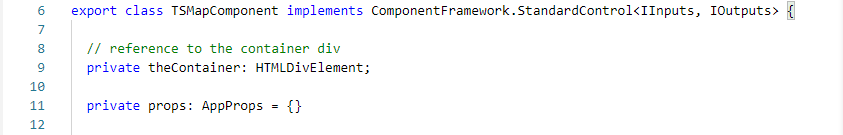


**Step 2.6.2** Add below static variable into class component

// reference to the container div

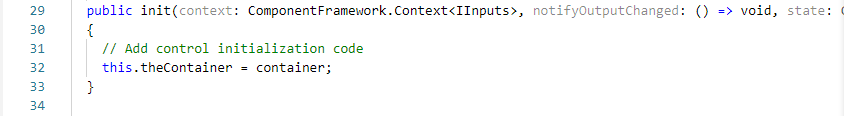
private theContainer: HTMLDivElement;

private props: AppProps = {}



**Step 2.6.3** Add below code into **init** method

this.theContainer = container;



**Step 2.6.4** Render React component into **updateView** method

// Render the React component into the div container

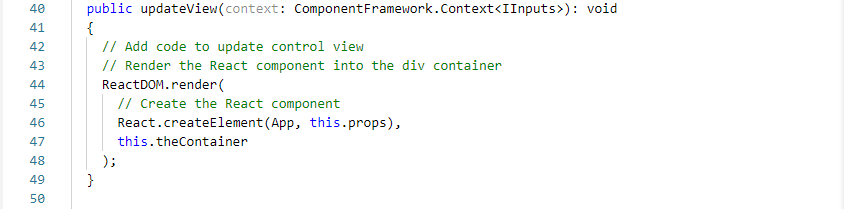
ReactDOM.render(

      // Create the React component

      React.createElement(App, this.props),

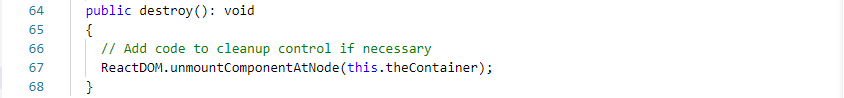
      this.theContainer

);



**Step 2.6.5** Cleanup the code into **destroy** method

ReactDOM.unmountComponentAtNode(this.theContainer);

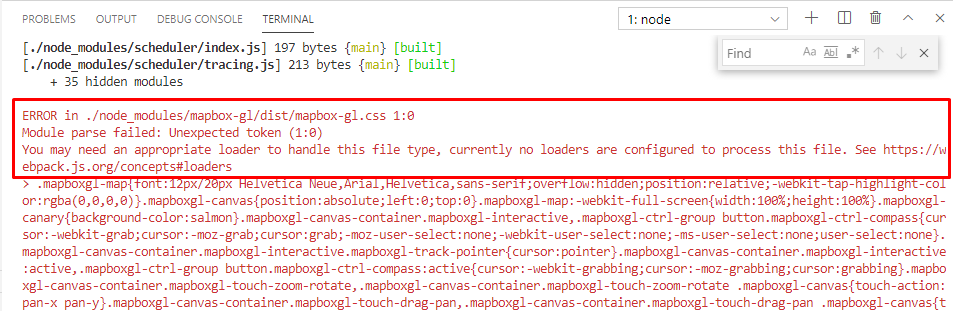


**Step 2.7**

Run the application using below command

* npm start

Get the below **error** after run this command.



**Step 2.8**

Resolve the CSS loader error issue into webpack config of pcf-scripts node package.

**Step 2.8.1** Add below lines into *node\_modules/pcf-scripts/webpackConfig.js* file

{

test: /\.css$/i,

    use: ['style-loader', 'css-loader'],

},



**Step 2.8.2** Install two dependencies in our application

* npm install --save css-loader
* npm install --save style-loader



**Step 2.9**

Now run the application using below command and started successfully.

* npm start

