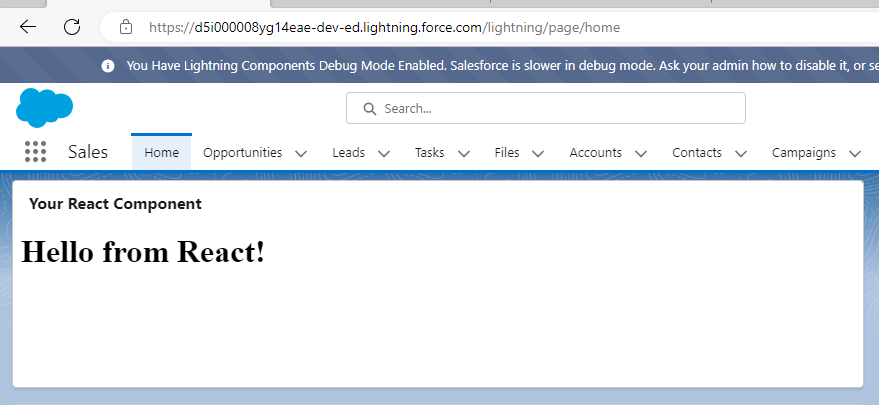
**Salesforce – ReactJs Data Transfer using LCC.Callapex**

**Step1**: Configure Webpack and Create Salesforce Project using the steps followed in below document.

****



**Step2**: Install Lightning-container using below command

npm i lightning-container

**Step3:** Create ReactJs project to display Account Details by providing Account name as input.

* Create App.js in the “Src” folder of our project (Delete Hello.js as we don’t want)



* Replace Index.Js file in the Src folder with below code,

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(

  <App />,

  document.getElementById('root')

);

* Create manifest.json file in “Public” folder and add below code,

{

    "landing-pages" : [

        {

            "path": "index.html",

            "apex-controller": "GetAccountRecordByName"

        }

    ]

}

* Replace Index.html file in the Public folder with below code,

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>React App</title>

  </head>

  <body>

    <div id="root"></div>

  </body>

</html>

* Create Apex Class “GetAccountRecordByName” for fetching account details,

global with sharing class GetAccountRecordByName {

public String accountName {get; set;}

public static Account account {get; set;}

public GetAccountRecordByName() {

}

@RemoteAction

global static Account getAccount(String accountName) {

account = [SELECT Id, Name, Phone, Type, NumberOfEmployees FROM Account WHERE Name = :accountName];

return account;

}

}

**Step4:** Create Aura Component for displaying the Details, (Replace Static resource name by your static resource name)

Cmp -> <https://github.com/developerforce/LightningContainerExamples/blob/master/ReactJS/Javascript/apex-controller/metadata/aura/ApexController/ApexController.cmp>

Js ->

<https://github.com/developerforce/LightningContainerExamples/blob/master/ReactJS/Javascript/apex-controller/metadata/aura/ApexController/ApexControllerController.js>

**Step5:** For Webpack – Manifest Configuration

Install Webpack-Manifest plugin by running below command in the Terminal

npm i webpack-manifest-plugin

In **webpack.config.js** file add the below lines at the beginning,

const { WebpackManifestPlugin } = require('webpack-manifest-plugin');

const options = { publicpath:null, filename: 'manifest.json'};

Add below lines in the Plugin property after HTMLWebpackPlugin,

new WebpackManifestPlugin({

        options

    }),

**Step6**: Now build Static resource by running below Command

npm run build

Check whethere Manifest.json file created in the Static Resource,

Replace Manfiest.json by below Code

{

  "landing-pages" : [

    {

        "path": "index.html",

        "apex-controller": "GetAccountRecordByName"

    }

  ]

}

After Deploying the changes, we can see the output like this

