**React App to Send Messages to Service Bus**

**Steps to Create React App to send message**

1. Create ReactJs application using create-React-App with below commands in command prompt

npm install create-react-app -g

npx create-react-app “App Name”

1. Install azure service bus npm package using below command

npm install @azure/service-bus

1. Install Bootstrap and reactstrap using below command for styling page.

npm install --save bootstrap

npm install --save reactstrap react react-dom

1. Import bootstrap in index.js file

import 'bootstrap/dist/css/bootstrap.min.css';

1. Import form, button and all elements to design form that sends message to service bus
2. Replace content of App.js with below code

import React, { Component } from 'react';

import { Col, Button, Form, FormGroup, Label, Input } from 'reactstrap';

class App extends Component {

constructor(props){

super(props);

this.state={message:""}

// this.onChangeMessage = this.onChangeMessage.bind(this);

// this.onSubmit = this.onSubmit.bind(this);

}

onChangeMessage=(e)=>{

this.setState({

message: e.target.value

});

}

onSubmit=async()=>{

// implementation to send message to service bus

)

render() {

return (

<div>

<div>

<h2 style={{textAlign:"center",marginTop:"50px"}}>Publish Message</h2>

</div>

<div style={{marginLeft:"500px",marginTop:"50px"}} >

<Form >

<FormGroup row >

<Label for="exampleText" sm={1}><b>Message</b></Label>

<Col sm={6}>

<Input type="textarea" onChange={this.onChangeMessage} name="text" id="exampleText" />

</Col>

</FormGroup>

<FormGroup row >

<Col sm={3}>

<div ></div>

</Col>

<Button onClick={this.onSubmit} className="my-3 ml-5">Send</Button>

</FormGroup>

</Form>

</div>

</div>

);

}

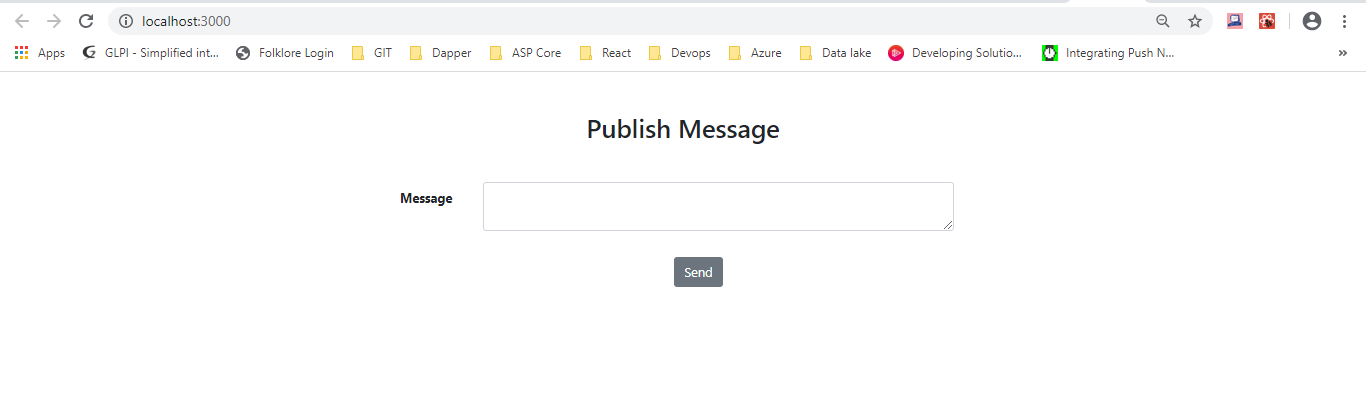
}

export default App;

1. Then run the Application using below command

npm start

1. Now browser will be showing page as below



1. Now let’s write code to send message in Message text box to service bus on click of send button. Replace below code with code and don’t forget to change **Endpoint**(primaryconnection string copied from shared access policy of Service bus service in azure portal ) and Topic Name

Please refer comments for code description.

onSubmit=async()=>{

console.log("-------submited--------")

//create service bus instance

const { ServiceBusClient } = require("@azure/service-bus");

// primary connection from shared access policy page of servicebus in //portal

const connectionString = "Endpoint=sb://msgbroker-servicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=Hz+9sofU/4UScgkKCEP/+2k9ylsVpd0UoNOBb7LtS1c=";

// Topic name created in service bus in azure portal

const topicName = "message-topic";

const sbClient = ServiceBusClient.createFromConnectionString(connectionString);

const topicClient = sbClient.createTopicClient(topicName);

const sender = topicClient.createSender();

try{

// message object

const message= {

body: this.state.message,

label: `test`,

};

console.log(`Sending message: ${message.body}`);

// async call to send message to service bus

await sender.send(message);

console.log("Message Sent");

this.setState({

message:""

});

}

finally {

await sbClient.close();

}

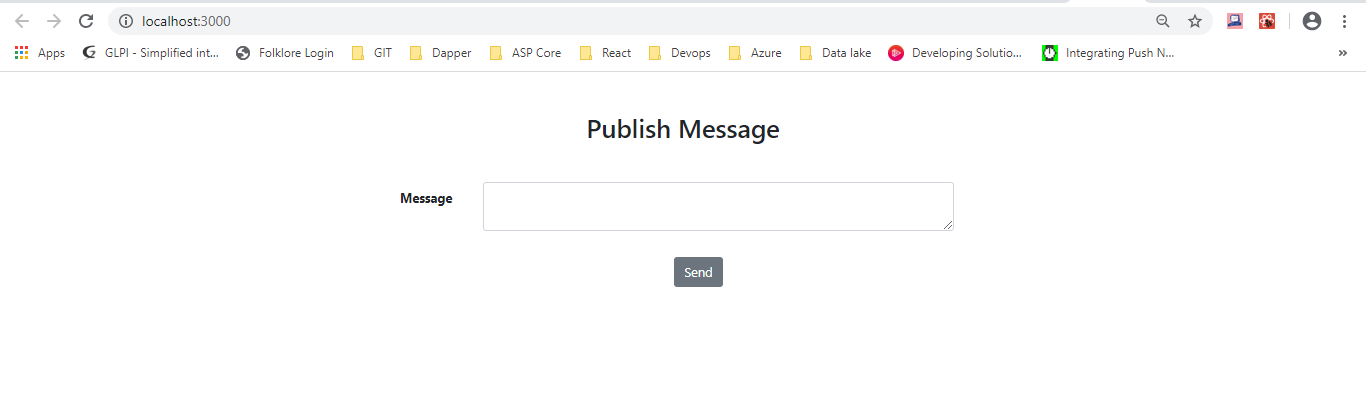
}

**Testing**

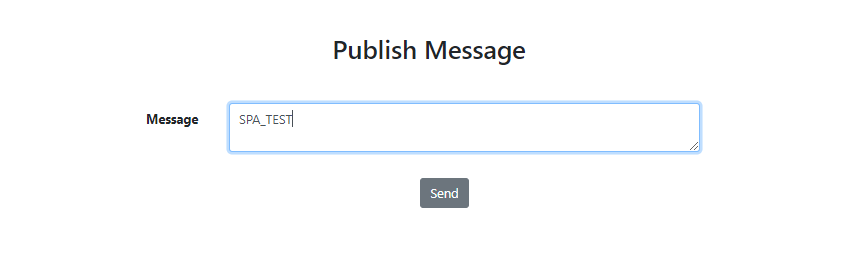
1. Then run the Application using below command

npm start

Below screen will be shown in browser**.**

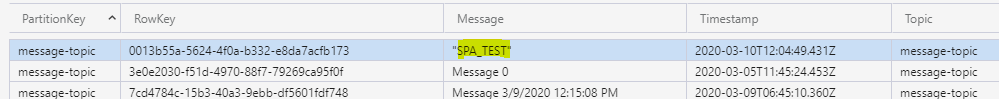


1. Enter “SPA\_TEST” in message text box and click **Send** button

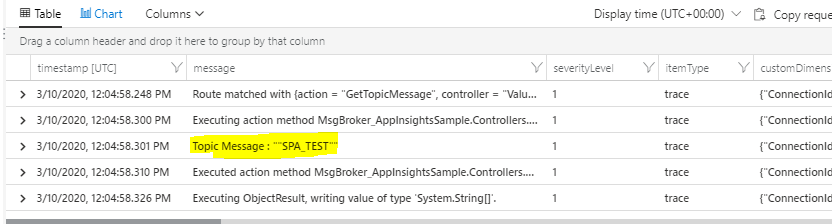


1. This send button sends message to service bus. This triggers 3 logic apps create in azure portal to send mail, store message in azure table storage and log message in application insights.

Check azure table storage table using storage explorer to see message was inserted or not



Checks logs of application insights to see the message logged or not



And check the inbox of the email registered in logic app, new mail should have been received with email body as “SPA\_Test”