

```

1  const { response } = require('express')
2  const e = require('express')
3  const express = require('express')
4  const https = require('https')
5  const cors = require("cors");
6  const API_KEY = '3e2d927d4f28b456c6bc662f34350957'
7  const DEFAULT_UNITS = 'metric'
8  const app = express()
9  const port = 8081
10 const path=require("path")
11
12
13 let publicPath= path.resolve(__dirname,"public")
14 app.use(express.static(publicPath))
15 app.use(cors());
16 app.use(express.static(publicPath))
17 app.listen(port, () => console.log(`Example app listening on port ${port}!`))
18
19 function getLocation(location){
20     let data = ''
21     let url = `https://api.openweathermap.org/data/2.5/forecast?
q=${location}&APPID=${API_KEY}&units=${DEFAULT_UNITS}`
22
23     let p = new Promise((resolve, reject)=>{
24         https.get(url, resp=>{
25             resp.on("data", (d)=>{
26                 data +=d
27             })
28             resp.on("end", ()=>{
29                 const parsedData = JSON.parse(data)
30                 resolve(parsedData)
31             })
32         }).on("error", (e)=>{
33             reject(e.message)
34         })
35     })
36     return p
37 }
38
39 function getAirPollution(lat, long){
40     let airRawData = ''
41     let airurl = `https://api.openweathermap.org/data/2.5/air_pollution/forecast?
lat=${lat}&lon=${long}&appid=${API_KEY}`
42
43     let p = new Promise((resolve, reject)=>{
44         https.get(airurl, resp=>{
45
46             resp.on("data", (d)=>{
47                 airRawData+=d
48             })
49             resp.on("end", ()=>{
50
51                 const airData = JSON.parse(airRawData)
52                 resolve(airData)
53             })
54         }).on("error", (e)=>{

```

```

55         reject(e.message)
56     })
57 })
58 return p
59 }
60
61
62 app.get('/5dayforecast/:location', (req, res)=>{
63     getLocation(req.params.location).then(result=>{
64         if(result.message){
65             res.json({error:result.message})
66         }else{
67             locationData = {
68                 long: result.city.coord.lon,
69                 lat: result.city.coord.lat
70             }
71             weatherData = (result.list).map(element => {
72                 if(element.rain){
73                     return {
74                         date: element.dt_txt,
75                         temp: Math.round(element.main.temp),
76                         wind: element.wind.speed,
77                         rain: element.rain['3h']
78                     }
79                 }else{
80                     return {
81                         date: element.dt_txt,
82                         temp: Math.round(element.main.temp),
83                         wind: element.wind.speed,
84                         rain: 0
85                     }
86                 }
87             })
88             res.json({weatherData:weatherData, locationData:locationData})
89         }
90     })
91 })
92 })
93
94 app.get('/5dayforecast/:lat/:long', (req, res)=>{
95     getAirPollution(req.params.lat, req.params.long).then(result=>{
96         if(result.message){
97             res.json({error:result.message})
98         }else{
99             airData = (result.list).map(element=>{
100                 return{
101                     pm25:element.components.pm2_5
102                 }
103             })
104             res.json({airData:airData})
105         }
106     })
107 })
108 })
109
110
111

```