

```
1 // Author : Naman Arora
2 // Date: 3rd December 2021
3
4 //initializing same as the class slides example
5 const express = require('express')
6 const app = express()
7 const port = 3000
8 const path = require("path")
9 const cors = require('cors')
10 const fetch = require('node-fetch')
11 let publicPath = path.resolve(__dirname, "public")
12
13 app.use(express.static(publicPath))
14 app.listen(port, () => console.log(`Weather app listening on port ${port}!`))
15
16 app.use(cors()) //to allow cross-origin access across our server & client
17
18 //Key details - taken from slide
19 const APIkey = "3e2d927d4f28b456c6bc662f34350957";
20
21 //Route for initial display page
22 app.get("/", function (req, res) {
23     res.sendFile(path.join(__dirname + "/weatherClient.html"))})
24
25 //Route for weather request
26 app.get("/weather/:location", weatherCheck)
27
28 //summary table for the next 5 days showing: Temperature, Wind Speed and
29 //Rainfall level
30 //returns JSON object of forecast to be printed on screen
31 async function weatherCheck(req, res) {
32     // Parse the city details and make api call to openweathermapAPI
33     let city = req.params.location
34     let APIurl = `https://api.openweathermap.org/data/2.5/forecast?
35 q=${city}&units=metric&appid=${APIkey}`
36     let response = await fetch(APIurl)
37     let dataOfWeather = await response.json()
38
39     // Define objects to contain data
40     let packStuff, rain, description, date, feels_like = ''
41     let temp, windSpeed, pm2_5 = 0
42     let allForecast = {
43         latitude : dataOfWeather.city.coord.lat,
44         longitude : dataOfWeather.city.coord.lon,
45         umbrella : 'No rain, no umbrella needed.',
46         temperatureAtPacking: '',
47         maskRequirement : 'Mask is not required. Air is clean.',
48         forecastList :[]
49     }
50     umbrella = 'No umbrella'
51     temperatureAtPacking = ' '
52     maskRequirement = 'No mask'
53
54     // Get the latitude & longitude of the location
55     let latitude = dataOfWeather.city.coord.lat
56     let longitude = dataOfWeather.city.coord.lon
57
58     // Define API url for pollution fetching data
```

```
57     let polUrl =
`http://api.openweathermap.org/data/2.5/air_pollution/forecast?
lat=${latitude}&lon=${longitude}&appid=${APIkey}`
58     let response2 = await fetch(polUrl)
59     let dataOfPollution = await response2.json()
60
61     //parse data from the fetched response json object
62     for(var index = 0; index < (dataOfWeather.list.length); index++){
63         date = dataOfWeather.list[index].dt_txt
64         description = dataOfWeather.list[index].weather[0].description
65         temp = dataOfWeather.list[index].main.temp
66         feels_like = dataOfWeather.list[index].main.feels_like
67         rain = rainCheck(dataOfWeather.list[index].rain, allForecast)
68         windSpeed = dataOfWeather.list[index].wind.speed
69         packStuff = packingCheck(dataOfWeather.list[1].main.temp,
allForecast)
70         pm2_5 = dataOfPollution.list[index].components.pm2_5
71         mask = maskCheck(pm2_5, allForecast)
72
73         allForecast.forecastList.push(
74             {
75                 Date: date,
76                 City: city,
77                 Description: description,
78                 Temp: temp,
79                 Feels: feels_like,
80                 Rainfall: rain,
81                 Windspeed: windSpeed,
82                 Packing: packStuff,
83                 PM2_5: pm2_5,
84                 mask: mask
85             }
86         )
87     }
88
89     res.json(allForecast)
90 }
91
92 function packingCheck(temp, allForecast){
93     //Determine the packing of user by temperature ranges
94     if(temp >= -10 && temp <= 10){
95         packStuff = "COLD"
96     }
97     else if(temp > 10 && temp <= 20){
98         packStuff = "WARM"
99     }
100     else{
101         packStuff = "HOT"
102     }
103     allForecast.temperatureAtPacking = "You should pack for " + packStuff + "
weather."
104     return packStuff
105 }
106
107 function rainCheck(dataOfWeather, allForecast){
108     //Get amount of rain and convert it to a string
109     //Determine if you should pack an umbrella
110     if(dataOfWeather != undefined){
111         if(JSON.stringify(dataOfWeather).substr(6,5) != ''){
112             rain = parseFloat((JSON.stringify(dataOfWeather)).substr(6, 5))
```

```
113     }
114     allForecast.umbrella = 'You should bring an umbrella.'
115 }
116 else{
117     rain = 'No rain, no umbrella needed.'
118 }
119 return rain
120 }
121
122 function maskCheck(maskR, allForecast){
123     //Determine the PM2_5 levels for mask requirements
124     if(maskR > 10){
125         mask = 'You should WEAR a mask !!'
126         allForecast.maskRequirement = 'You should WEAR a mask !!'
127     }
128     else{
129         mask = 'Mask is not required. Air is clean.'
130     }
131
132     return mask
133 }
```