

index.js

```
require("dotenv").config();
const express = require("express");
const path = require("path");

const API_KEY = process.env.WEATHER_API_KEY;
if (!API_KEY) {
  console.log("could not find api key, exiting.");
  process.exit(1);
}
const API_BASE_URL = "https://api.openweathermap.org/data/2.5/";
const SERVER_PORT = 5501;
const DEFAULT_UNITS = "metric";

const app = express();
const axios = require("axios").default;
const cors = require("cors");
console.log("starting");
app.use(cors()); //Allow cross origin requests

function getWeatherForLocation(location) {
  return axios
    .get(API_BASE_URL + "forecast", {
      params: {
        q: location,
        APPID: API_KEY,
        units: DEFAULT_UNITS,
      },
    })
    .then((res) => res.data)
    .catch((err) => err.response);
}

function getAirPollutionForLocation(lat, lon) {
  return axios
    .get(API_BASE_URL + "air_pollution/forecast", {
      params: {
        lat: lat,
        lon: lon,
        APPID: API_KEY,
      },
    })
    .then((res) => res.data)
    .catch((err) => err.response);
}

app.use(express.static(path.join(__dirname, "public")));

app.get("/5dayweather", (req, res) => {
  getWeatherForLocation(req.query.location).then((weatherRes) => {
    // console.log(weatherRes);
    if (weatherRes.data?.message) {
      //send error message if there is one
      res.json({
        error:
          weatherRes.data.message.charAt(0).toUpperCase() +
          weatherRes.data.message.slice(1),
      });
    }
  });
});
```

```

    } else {
      let weatherData = weatherRes.list.map((element) => {
        //reorder and send the data
        return {
          timeString: element.dt_txt,
          temp: Math.round(element.main.temp),
          rain: element.rain?.["3h"] ?? 0,
          wind: element.wind,
        };
      });
      res.json({ weather: weatherData, city: weatherRes.city });
    }
  });
});

app.get("/airqual", (req, res) => {
  let lat = req.query.lat;
  let lon = req.query.lon;

  getAirPollutionForLocation(lat, lon).then((airRes) => {
    if (airRes) {
      if (airRes.data?.message) {
        res.json({
          error:
            airRes.data.message.charAt(0).toUpperCase() +
            airRes.data.message.slice(1),
        });
      } else {
        // console.log(airRes);
        let airData = airRes.list.map((element) => {
          return {
            timeString: element.dt,
            aqi: element.main.aqi,
          };
        });
        res.json({ airData });
      }
    } else {
      res.json({ error: "something borked" });
    }
  });
});

app.listen(SERVER_PORT, () =>
  console.log(`Server running on port: ${SERVER_PORT}`)
);

```