

```

bin
  server.js
routes
  api
    weather.js
app.js
.env
.gitignore

```

```

package.json
"scripts": {
  "start": "node ./bin/server.js",
  "dev": "nodemon ./bin/server.js"
},
"dependencies": {
  "dotenv": "8.2.0",
  "express": "4.17.1",
  "express-validator": "6.9.2",
  "got": "11.8.1",
  "morgan": "1.10.0"
},
"devDependencies": {
  "nodemon": "2.0.7"
}

```

```

app.js
const express = require('express')
const logger = require('morgan')

const app = express()

const routesWeather = require('./routes/api/weather')

app.use(logger('dev'))
app.get('/', (req, res) => {
  res.send('Hello World!')
})
app.use('/weather', routesWeather)
app.use((_req, res) => {
  res.status(404).send({ message: 'Not found' })
})
app.use((err, _req, res, _next) => {
  res.status(500).send({ message: err.message })
})

module.exports = app

```

```

/bin/server.js
const app = require('../app')

const PORT = process.env.PORT || 3000

app.listen(PORT, () => {
  console.log(`Server start on: ${PORT}`)
})

```

```

/routes/api/weather.js
const express = require('express');
const got = require('got');
const { query, validationResult } = require('express-validator');

const router = express.Router();
require('dotenv').config();

const validator = (req, res, next) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) {
    return res.status(400).json({ message: errors.array() });
  }
  next()
}

// define the home page route
router.get("/",
  [
    query('lat').isNumeric(),
    query('lon').isNumeric()
  ],
  validator,
  async (req, res, next) => {
    const { lat, lon } = req.query;
    const apiKey = process.env.API_KEY;
    try {
      const response = await got(
        "http://api.openweathermap.org/data/2.5/weather",
        {
          searchParams: {
            lat,
            lon,
            appid: apiKey,
          },
        }
      )
      const {weather, wind, name} = JSON.parse(response.body)
      res.json({weather: weather[0], wind, name});
    } catch (e) {
      next(e);
    }
  }
);

module.exports = router;

```

```

.env
API_KEY=7d94f...24e49fcd639b

.gitignore
node_modules
.env

```