

HOW TO CREATE A WEATHER APP IN NODEJS USING OPEN WEATHER MAP API

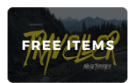
🕒 September 17, 2018 📁 JavaScript, NodeJS

Last updated: 7th November, 2018

How to create a weather app in nodejs

Nowadays there are many web services that provide the latest weather updates. **Open Weather Map** is a widely used [API](#) by developers. In this tutorial, you are going to create a weather app in NodeJS. A user enters a city name and clicks the **Fetch Weather** button, weather data returned from Open Weather API is displayed to the user.

READY TO USE CREATIVE ASSETS



envatoelements

START NOW

Weather App in NodeJS using Open Weather API

Enter City Name **Fetch Weather**

Weather Forecast

Longitude / Latitude

Pressure

Temperature

Humidity

Speed

Enter city name and click Fetch Weather button

Following tasks are performed in this tutorial

1. Register on **Open Weather Map** website.
2. Generate a weather app in nodejs using express command line tool
3. Create an HTML form and a table to display data
4. Create a **/weather** route to fetch weather for a city



Register for Open Weather Map API

Visit [Open Weather Map](#) website and [register](#) an account.

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using [Privacy Centre](#). You have the right to access your data held by us or to request your data to be deleted. For full details please see the [OpenWeather Privacy Policy](#).

☐ I am 16 years old and over


☐ I agree with [Privacy Policy](#), [Terms and conditions of sale](#) and [Websites terms and conditions of use](#)

I consent to receive communications from OpenWeather Group of Companies and their partners:

☐ System news (API usage alert, system update, temporary system shutdown, etc)

☐ Product news (change to price, new product features, etc)

☐ Corporate news (our life, the launch of a new service, etc)

☐ I'm not a robot 
reCAPTCHA
[Privacy](#) - [Terms](#)



Create Account

Login to your account after account creation. Click on [API Keys](#) and you can view the **API** Key. Copy this API Key to use in weather app in nodejs.

API keys

[Home](#)[Setup](#) [API keys](#) [My Services](#) [My Payments](#) [Billing plans](#) [Map editor](#) [Block logs](#) [History bulk](#)[Logout](#)

Activation of an API key for **Free** and **Startup accounts** takes **10 minutes**. For **other accounts** it takes from **10 to 60 minutes**. You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

| Key | Name | | Create key |
|-------------------|---------|---|--|
| 3598[REDACTED]95e | Default |   | <div><div>* Name</div><div></div><div>Generate</div></div> |

You can view different price plans by visiting this [link](#). In a **free plan**, there are following limits.

- **Threshold:** 7,200
- **Hourly forecast:** 5
- **Daily forecast:** 0
- **Calls 1 min:** 60

Using open weather map API you can get historical data, weather alerts, weather map layers, current weather data, the daily forecast for 16 days, and 3-hourly forecast or 5 days for your city.

Get city weather by name

To get a city weather, you can use the URL given below. The **cityName** and **appid** are used as placeholders.

1. `http://api.openweathermap.org/data/2.5/weather?q=CityName&appid=YOUR_API_KEY`

Cityname is replaced by user entered city name. Replace API KEY with your account's API Key. If you access above URL in the browser and add city name as London.

```
▼ coord:
  lon:      -0.13
  lat:      51.51
▼ weather:
  ▼ 0:
    id:      800
    main:    "Clear"
    description: "clear sky"
    icon:    "01n"
    base:    "stations"
  ▼ main:
    temp:    286.65
    pressure: 1021
    humidity: 77
    temp_min: 286.15
    temp_max: 288.15
    visibility: 10000
  ▼ wind:
    speed:    5.7
    deg:      250
  ▼ clouds:
    all:      0
    dt:      1536965400
  ▼ sys:
    type:    1
    id:      5091
    message: 0.5589
    country: "GB"
    sunrise: 1536903307
    sunset:  1536948924
  id:      2643743
  name:    "London"
  cod:     200
```

Install NodeJS

To install **NodeJS**, download NodeJS from the [download page](#) and install. **NPM** or **Node Package Manager** is also installed with nodejs.

Install Express Generator tool

Express tool is also installed with the express generator installation. Express is used to generate a nodejs express app skeleton.

```
1. npm install express-generator -g
```

Generate weather app in nodejs using express

Open the command line and go to the directory where you want to create the project. **Pug** is used as a template engine.

```
1. express --view=pug weather-app-nodejs
```

```
create : weather-app-nodejs
create : weather-app-nodejs/package.json
create : weather-app-nodejs/app.js
create : weather-app-nodejs/public
create : weather-app-nodejs/routes
create : weather-app-nodejs/routes/index.js
create : weather-app-nodejs/routes/users.js
create : weather-app-nodejs/views
create : weather-app-nodejs/views/index.pug
create : weather-app-nodejs/views/layout.pug
create : weather-app-nodejs/views/error.pug
create : weather-app-nodejs/bin
create : weather-app-nodejs/bin/www
create : weather-app-nodejs/public/javascripts
create : weather-app-nodejs/public/stylesheets
create : weather-app-nodejs/public/stylesheets/style.css

install dependencies:
> cd weather-app-nodejs && npm install

run the app:
> SET DEBUG=weather-app-nodejs:* & npm start

create : weather-app-nodejs/public/images
```

Install dependencies and modules

Next, install the required dependencies

```
1. cd weather-app-nodejs && npm install
```

Install nodemon module

Normally, if you have a running app in nodejs and a change is made to the code files, the app needs to be restarted. **nodemon** is a module that monitors any change in files and automatically restarts the server.

```
1. npm i nodemon
```

Run the application

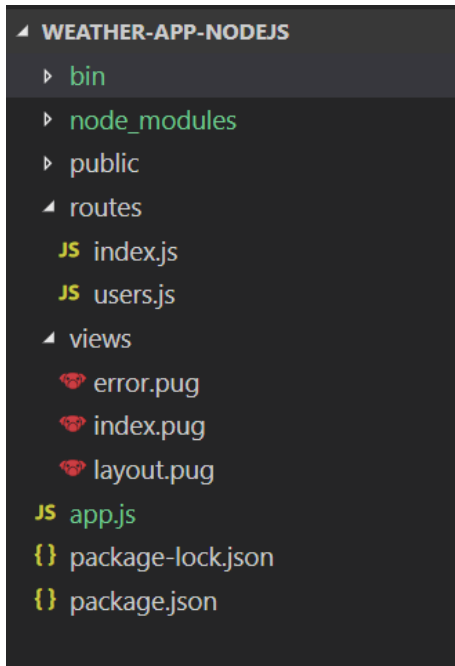
On the command line, type

```
1. nodemon
```

To view the running application, open browser, and type URL.

```
1. http://localhost:3000
```

Open the project in your favorite IDE like Atom, Sublime Text or Brackets. The project structure looks like.



Add bootstrap in layout.pug file

Open **layout.pug** file in **views** directory.

```
1. doctype html
2. html(lang="en")
3.   head
4.     title Weather App in NodeJS
5.     meta(charset="utf-8")
6.     meta(name="viewport" content="width=device-width, initial-scale=1")
7.     link(rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css")
8.     script(src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js")
9.     script(src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js")
10.   body
11.     block content
```

Below body tag, a block named **content** is added.

Create a form in index.pug file

Open **index.pug** file and add the code.

```
1. extends layout
2. block content
3.   .container
4.     h2
5.       p
6.         | Weather App in NodeJS using Open Weather API
7.       br
8.       br
9.       .row
10.        .col-md-6
11.          form.form-inline(method="post", action="weather")
12.            .form-group
13.              label.sr-only(for="inputEmail") City Name
14.              input.form-control(name="city" type="text" placeholder="Enter City Name" required='required')
15.            |
16.            button.btn.btn-primary(type="submit") Fetch Weather
17.
18.
19.   hr
```

Join Setapp

Ad Creating a r
informative min

Setapp.com

OPEN

After the **extends layout** command a form is created. HTTP **POST** method is used to send the data to the server. **Action** parameter is set to **weather** route. A text input named **city** with a submit button is added.

Add an HTML table to display records

Add a table below the form.

```

1.  .row
2.      .col-md-12
3.          p
4.              br
5.              br
6.              strong Weather Forecast #{forecast}
7.              table.table
8.                  thead
9.                      tr
10.                         th Longitude / Latitude
11.
12.                         th Pressure
13.                         th Temperature
14.                         th Humidity
15.                         th Speed
16.                  tbody
17.                      if body
18.                          tr
19.                              td
20.                                  #{body.coord.lon} / #{body.coord.lon}
21.                              td
22.                                  #{body.main.pressure}
23.                              td
24.                                  #{body.main.temp}
25.                              td
26.                                  #{body.main.humidity}
27.                              td
28.                                  #{body.wind.speed}
29.                      else
30.                          tr
31.                              td(colspan="6")
32.                                  | Enter city name and click Fetch Weather button

```

In table header, longitude/latitude, pressure, temperature, humidity and speed are added. If the body array has data then coordinates, pressure, temperature, humidity, and speed data is displayed.

Weather App in NodeJS using Open Weather API

Enter City Name

Weather Forecast

| Longitude / Latitude | Pressure | Temperature | Humidity | Speed |
|----------------------|----------|-------------|----------|-------|
|----------------------|----------|-------------|----------|-------|

Enter city name and click Fetch Weather button

Install request module in weather app in nodejs

To send a request to **open weather map** API, **request** module is installed.

```
1. npm i request --save
```

Add API credentials in index.js

Open **index.js** in **routes** directory. A URL variable is created that points to open weather map's API. Notice an URL parameter **?q=**. The value after param **q** is appended after the user's input. Add your account's API Key in **appid** variable. Units are used as metric.

Join Setapp

Ad Creating a r
informative mini

Setapp.com

OPEN

```
1. let url    = 'http://api.openweathermap.org/data/2.5/weather?q='
2. let appId = 'appid=YOUR API KEY';
3.
4. let units  = '&units=metric';
5.
6. var request = require('request');
```

Update index route

Replace the code in **'/'** route in **index.js**. Empty **body** and, **forecast** variables assigned to the template.

```
1. /* GET home page. */
2.
3. router.get('/', function(req, res, next) {
4.
5.   res.render('index', { 'body': '', forecast: '' });
6.
7. });
```

Create the Weather route

Open **index.js** in **routes** folder.

```
1. router.post('/weather', function(req, res, next){
2.
3.   let city = req.body.city;
4.
5.   url = url+city+"&" +appId;
6.
7.   request(url, function (error, response, body) {
8.
9.     body = JSON.parse(body);
10.
11.     if(error && response.statusCode !== 200){
12.
13.       throw error;
14.
15.     }
16.
17.     let country = (body.sys.country) ? body.sys.country : '' ;
18.
19.     let forecast = "For city "+city+', country '+country;
20.
21.     res.render('index', {body : body, forecast: forecast});
22.
23.   });
24.
25. });
```

Form data is posted to this route. The city name is assigned to the city variable using Nodejs's request object. **URL** is appended with city name and appID. Using **request** method to the URL, a request is sent to open weather map API.

This method returns the error, response and body parameters in the callback method. An error is thrown if an error is returned. Body parameter is returned as a JSON. JSON data is parsed. Forecast variable, created with country and city variable name and body variable is assigned to the template.

If you **console.log** the JSON data returned from open weather map API on the server.

Search for Toronto’s weather

```
1.  {
2.    "coord": {
3.      "lon": -79.39,
4.      "lat": 43.65
5.    },
6.    "weather": [
7.      {
8.        "id": 800,
9.        "main": "Clear",
10.       "description": "clear sky",
11.       "icon": "01d"
12.     }
13.   ],
14.   "base": "stations",
15.   "main": {
16.     "temp": 297.7,
17.     "pressure": 1022,
18.     "humidity": 83,
19.     "temp_min": 297.05,
20.     "temp_max": 298.15
21.   },
22.   "visibility": 14484,
23.   "wind": {
24.     "speed": 2.1,
25.     "deg": 180
26.   },
27.   "clouds": {
28.     "all": 1
29.   },
30.   "dt": 1537020000,
31.   "sys": {
32.     "type": 1,
33.     "id": 3721,
34.     "message": 0.0029,
35.     "country": "CA",
36.     "sunrise": 1537009056,
37.     "sunset": 1537053998
38.   },
39.   "id": 6167865,
40.   "name": "Toronto",
41.   "cod": 200
42. }
```

The JSON data contains coordinates, weather, temperature, and other information.

Search for London weather in weather app in nodejs

If the app is still running, enter **London** as city and click fetch weather button. Weather data such as longitude, latitude, pressure, temperature, humidity, and speed is displayed.

Weather App in NodeJS using Open Weather API

Weather Forecast For city London, country GB

| Longitude / Latitude | Pressure | Temprature | Humidity | Speed |
|----------------------|----------|------------|----------|-------|
| -0.13 / -0.13 | 1021 | 286.86 | 72 | 5.7 |

Search for New York's weather in weather app in nodejs

If user searches for **New York** City weather.

Weather App in NodeJS using Open Weather API

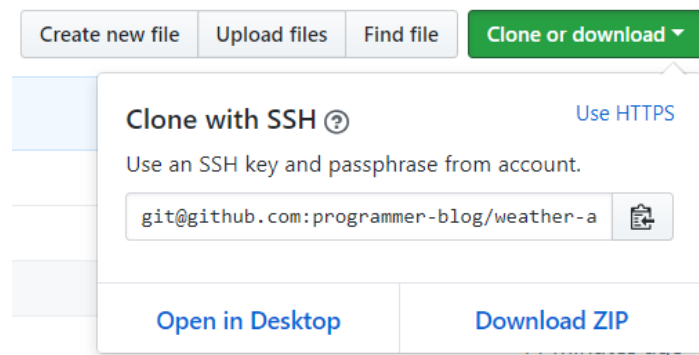
Weather Forecast For city New York, country US

| Longitude / Latitude | Pressure | Temperature | Humidity | Speed |
|----------------------|----------|-------------|----------|-------|
| -73.99 / -73.99 | 1023 | 296.05 | 71 | <4.6> |

Summary

To summarize, in this tutorial on creating a weather app in nodejs, an account is registered on Open weather map. A nodejs app is generated and a form with an HTML table is created. When the user enters the name of a city, a request is sent to Open Weather API. JSON data is sent back to the server and is displayed to the user.

You can clone or download the source code for this tutorial from this [GitHub repository](#).



Leave your valuable feedback. To stay updated on upcoming articles and tutorials, follow us on **twitter**, like us on **Facebook** or subscribe to our **newsletter**.

Related Articles

- [NodeJS MongoDB RESTful API](#)
- [Nodejs MongoDB Tutorial](#)
- [Nodejs File Upload With MongoDB – Photo Gallery](#)
- [Nodejs user registration tutorial](#)
- [Using socketio with nodejs](#)

Previous Articles

- [NodeJS Loopback RESTful API](#)

Next Article

- [Send SMS using nodejs](#)



3 Comments programmerblog

Login

Recommend 2 Tweet Share

Sort by Best



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name



BrainPundits • a year ago

Wow this certainly looks very interesting.
Keep up the good work

Do you have all the libraries on the Github?

^ | v • Reply • Share ›



jason Mod → BrainPundits • a year ago

Hi,
Thanks for the comment. Yes all code is present on GitHub. Create you account on OpenWeather Site and update index.js file. and that it.

Thanks

^ | v • Reply • Share ›



BrainPundits → jason • a year ago

Thank you

^ | v • Reply • Share ›

ALSO ON PROGRAMMERBLOG

How to generate RSS Feed using NodeJS and MySQL

3 comments • 2 years ago



János Mikle — Hi Jason,i've found the problem. The xml re-generate happening only in that case when you call index.js. In caseif you call only just the rssfeed.xml nothing change. Do you have any idea how

NodeJS Passport login script with MySQL database

27 comments • 2 years ago



ludakris4 — My desereailize user function is never called. I am having issues with the sessions/cookie part haven't been able to figure it out yet.

nodejs user registration tutorial with mongodb

6 comments • 2 years ago



jason — Hi Sateesh,Thanks for your comments.
Yes we can help you. We already have a tutorial on CRUD operations.
CRUD Tutorial: <https://programmerblog.net/...Other related>

How to generate xml files using php and mysql database

34 comments • 3 years ago



facebook-100000868941261 — Hi Jason,Thank you for the fast response.Yes, now works like a charm.
It's really strange because this morning I have tried the script at the

Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy