

1. Sign in and get the api token

OpenWeather

Weather in your city

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business Saty... Support

New Products Services **API keys** Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
d9faf162c5f42425d5830a87090816eb	Default	Active	

Create key

API key name **Generate**

2. With the api token get the data

Home Workspaces API Network Explore

Search Postman

Invite Upgrade

pace1 New Import GET Weather

OpenWeather / Weather

Save

GET Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
appid	d9faf162c5f42425d5830a87090816eb	
q	Bhubaneswar	

Body Cookies Headers (9) Test Results (4/4)

Status: 200 OK Time: 181 ms Size: 794 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "coord": {
3     "lon": 85.8333,
4     "lat": 20.2333
5   },
6   "weather": [
7     {
8       "id": 721,
9       "main": "Haze",
10      "description": "haze",
11      "icon": "50n"
12    }
13  ],
14  "base": "stations",
15  "main": {
16    "temp": 304.27,
17    "feels_like": 311.27,
18    "temp_min": 304.27,
19    "temp_max": 304.27,
```

Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Make authorizing this easy for your users. [Set up authentication for your APIs](#)

3. Add tests with it

Params ● Authorization ● Headers (6) Body Pre-request Script **Tests ●** Settings

```
1
2 pm.test("Response status code is 200", function () {
3     pm.expect(pm.response.code).to.equal(200);
4 });
5
6
7 pm.test("Validate coord object", function () {
8     const responseData = pm.response.json();
9
10    pm.expect(responseData).to.be.an('object');
11    pm.expect(responseData.coord).to.exist.and.to.be.an('object');
12    pm.expect(responseData.coord.lon).to.exist.and.to.be.a('number');
13    pm.expect(responseData.coord.lat).to.exist.and.to.be.a('number');
14 });
15
16
17 pm.test("Weather is an array with at least one element", function () {
18     const responseData = pm.response.json();
19
20    pm.expect(responseData.weather).to.be.an('array').and.to.have.lengthOf.at.least(1);
21 });
22
23
24 pm.test("Visibility is a non-negative integer", function () {
25     const responseData = pm.response.json();
26
27    pm.expect(responseData.visibility).to.be.a('number');
28    pm.expect(responseData.visibility).to.be.at.least(0, "Visibility should be a
    non-negative integer");
29 });
30
31 |
```

Output:

Body Cookies Headers (9) Test Results (4/4)

Status: 200 OK Time: 816 ms

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "coord": {
3     "lon": 85.8333,
4     "lat": 20.2333
5   },
6   "weather": [
7     {
8       "id": 721,
9       "main": "Haze",
10      "description": "haze",
11      "icon": "50n"
12    }
13  ],
14  "base": "stations",
15  "main": {
16    "temp": 304.27,
17    "feels_like": 311.27,
18    "temp_min": 304.27,
19    "temp_max": 304.27,
20    "pressure": 1003,
21    "humidity": 74
22  },
23  "visibility": 5000,
24  "wind": {
25    "speed": 1.54,
26    "deg": 280
27  },
28  "clouds": {
29    "all": 75
30  }
31 }
```

Body Cookies Headers (9) Test Results (4/4)

All

Passed

Skipped

Failed



PASS Response status code is 200

PASS Validate coord object

PASS Weather is an array with at least one element

PASS Visibility is a non-negative integer