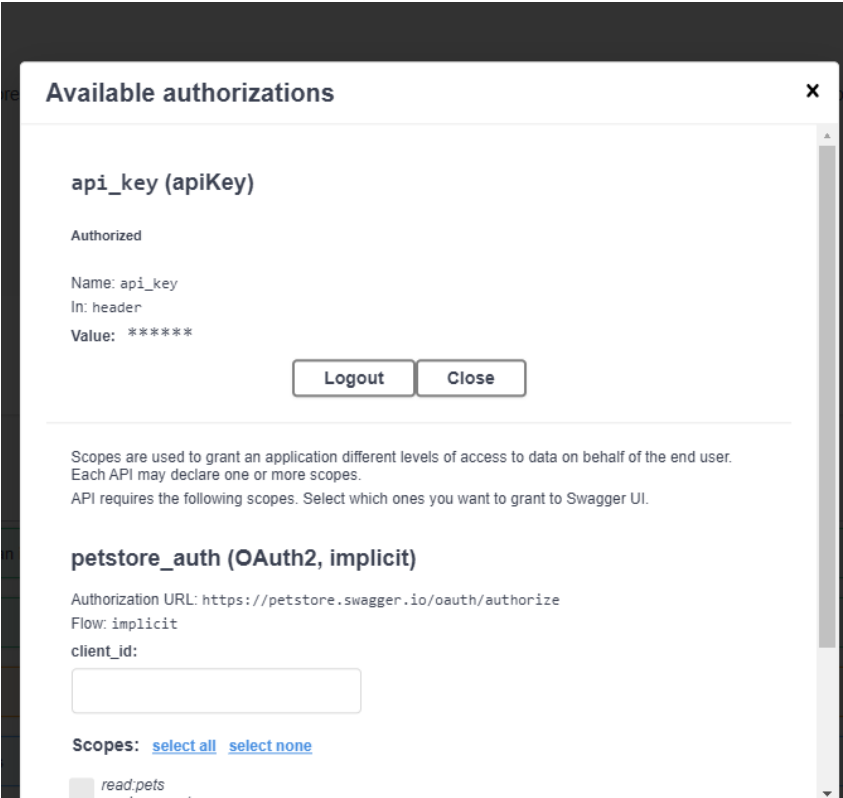
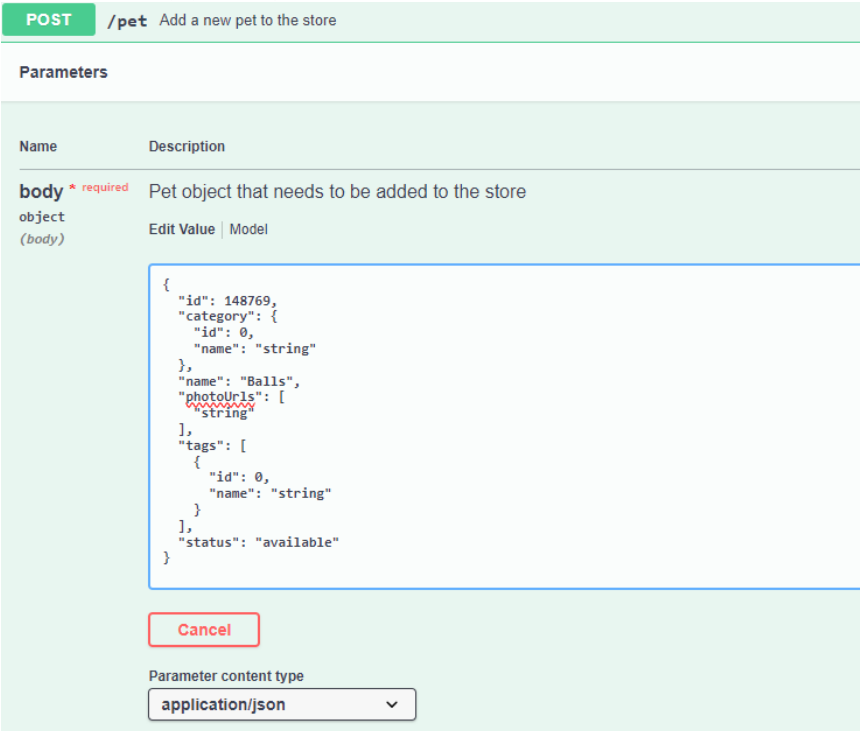


Тема Лабораторно й Работы	Lb_5.1
Выполняющи й	Коваленко Кирилл ИС221
Помогающий	Лавренов Александр ИС221
Ход работы	<div>1. “Логинимся” в swagger</div> <div></div> <div>2. редактируем /pet</div> <div></div>

3. Вывод при запуске кода 2

Curl

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 148769,
    "category": {
      "id": 0,
      "name": "string"
    },
    "name": "Balls",
    "photoUrls": [
      "string"
    ],
    "tags": [
      {
        "id": 0,
        "name": "string"
      }
    ],
    "status": "available"
  }'
```

Request URL

https://petstore.swagger.io/v2/pet

4. Так же получаем xml

Response body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Pet>
  <category>
    <id>0</id>
    <name>string</name>
  </category>
  <id>148769</id>
  <name>Balls</name>
  <photoUrls>
    <photoUrl>string</photoUrl>
  </photoUrls>
  <status>available</status>
  <tags>
    <tag>
      <id>0</id>
      <name>string</name>
    </tag>
  </tags>
</Pet>
```

5. В GET pet {petID} вводим айди нашего питомца и получаем информацию о нашем API и питомце

6. Настраиваем Swagger editor, пишем код на YAML

Swagger Editor
File Edit Insert Generate Server Generate Client About

```
1 info:
2   title: "OpenWeatherMap API"
3   description: "Get the current weather, daily forecast for 16 days, and a three-hour-interval forecast for 5
4     days for your city. Helpful stats, graphics, and this day in history charts are available for your reference.
5     Interactive maps show precipitation, clouds, pressure, wind around your location stations. Data is
6     available in JSON, XML, or HTML format. **Note:** This sample Swagger file covers the 'current' endpoint
7     only from the OpenWeatherMap API. Other endpoints have other parameters, but you must select at
8     least one parameter. Calling the API by city ID (using the 'id' parameter) will provide the most precise
9     location results."
10
11 version: "2.1"
12 termsOfService: "https://openweathermap.org/terms"
13 contact:
14   name: "OpenWeatherMap API"
15   url: "https://openweathermap.org/api"
16   email: "info@openweathermap.org"
17 license:
18   name: "CC Attribution-ShareAlike 4.0 International"
19   url: "https://openweathermap.org/pricing"
20
21 servers:
22   - url: "https://api.openweathermap.org/data/2.5/"
23
24 externalDocs:
25   description: "API Documentation"
26   url: "https://openweathermap.org/api"
27
28 paths:
29   /weather:
30     get:
31       tags:
32         - "Current Weather Data"
33       summary: "Get current weather data for one location"
34       description: "Access current weather data for any location on Earth including over 200,000 cities! Current
35         weather is frequently updated based on global models and data from more than 40,000 weather stations."
36       operationId: "GetCurrentWeatherData"
37       parameters:
38         - $ref: "#/components/parameters/q"
39         - $ref: "#/components/parameters/lat"
40         - $ref: "#/components/parameters/lon"
41         - $ref: "#/components/parameters/zip"
42         - $ref: "#/components/parameters/lang"
43         - $ref: "#/components/parameters/units"
44         - $ref: "#/components/parameters/appid"
45       responses:
46         200:
47           description: "Successful response"
48           content:
49             application/json:
50               schema:
51                 $ref: "#/components/schemas/Weather"
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
```

Try out now Editor

Current Weather Data

GET /weather Call current weather data for one location

Access current weather data for any location on Earth including over 200,000 cities! Current weather is frequently updated based on global models and data from more than 40,000 weather stations.

Parameters

Cancel

Name	Description
q	City name. Example: London. You can call by city name, or by city name and country code. The API responds with a list of results that match a searching word. For the query value, type the city name and optionally the country code divided by a comma, use ISO 3166 country codes.
id	City ID. Example: 21272797. You can call by city ID. The API responds with the exact result. The list of city IDs can be downloaded here . You can include multiple cities in this parameter — just separate them by commas. The limit of locations is 20. Note: A single ID counts as a one API call. So, if you have city IDs, it's treated as 3 API calls.
lat	Latitude. Example: 35. The latitude coordinate of the location of your interest. Must use with lon .
lon	Longitude. Example: 139. Longitude coordinate of the location of your interest. Must use with lat .
zip	Zip code. Search by zip code. Example: 95050,us. Please note that if the country is not specified, the search uses USA as a default.
lang	Language. Example: en. You can use lang parameter to get the output in your language. We support the following languages that you can use with the corresponded lang values: Arabic - 'ar', Bulgarian - 'bg', Catalan - 'ca', Czech - 'cz', German - 'de', Greek - 'el', English - 'en', Persian (Farsi) - 'fa', Finnish - 'fi', French - 'fr', Galician - 'gl', Croatian - 'hr', Hungarian - 'hu',
units	Units. Example: imperial. Possible values: 'standard', 'metric', and 'imperial'. When you do not use the 'units' parameter, the format is 'standard' by default.

Swagger Editor
File Edit Insert Generate Server Generate Client About

```
63   type: string
64   id:
65     name: id
66     in: query
67     description: "**City ID**. *Example: `21272797`*. You can call by city ID. The API responds with the exact
68       result. The List of city IDs can be downloaded [here](http://bulk.openweathermap.org/sample/). You can
69       include multiple cities in this parameter &mdash; just separate them by commas. The limit of locations
70       is 20. *Note: A single ID counts as a one API call. So, if you have city IDs, it's treated as 3 API
71       calls.*"
72     schema:
73       type: string
74
75   lat:
76     name: lat
77     in: query
78     description: "**Latitude**. *Example: 35*. The latitude coordinate of the location of your interest. Must
79       use with `lon`."
80     schema:
81       type: string
82
83   lon:
84     name: lon
85     in: query
86     description: "**Longitude**. *Example: 139*. Longitude coordinate of the location of your interest. Must
87       use with `lat`."
88     schema:
89       type: string
90
91   zip:
92     name: zip
93     in: query
94     description: "**Zip code**. Search by zip code. *Example: 95050,us*. Please note that if the country is
95       not specified, the search uses USA as a default."
96     schema:
97       type: string
98
99   units:
100     name: units
101     in: query
102     description: "**Units**. *Example: imperial*. Possible values: `standard`, `metric`, and `imperial`. When
103       you do not use the `units` parameter, the format is `standard` by default."
104     schema:
105       type: string
106       enum: [standard, metric, imperial]
107       default: "imperial"
108
109   lang:
110     name: lang
111     in: query
112     description: "**Language**. *Example: en*. You can use lang parameter to get the output in your language.
113       We support the following languages that you can use with the corresponded lang values: Arabic - `ar`,
114       Bulgarian - `bg`, Catalan - `ca`, Czech - `cz`, German - `de`, Greek - `el`, English - `en`, Persian
115       (Farsi) - `fa`, Finnish - `fi`, French - `fr`, Galician - `gl`, Croatian - `hr`, Hungarian - `hu`,
```

7. Введем в поле q Лондон, а ниже введем апи ключ и получим погоду в Лондоне

q

string

(query)

City name. Example: London. You can call by city name, or by city name and country code. The API responds with a list of city name and optionally the country code divided by a comma; use ISO 3166 country codes.

London

id

string

(query)

City ID. Example: 2172797. You can call by city ID. The API responds with the exact result. The List of city IDs can be do separate them by commas. The limit of locations is 20. Note: A single ID counts as a one API call. So, if you have city IDs, it's

id

lat

string

(query)

Latitude. Example: 35. The latitude coordinate of the location of your interest. Must use with lon.

lat

lon

string

(query)

Longitude. Example: 139. Longitude coordinate of the location of your interest. Must use with lat.

lon

zip

string

(query)

Zip code. Search by zip code. Example: 95050,us. Please note that if the country is not specified, the search uses USA as a

zip

units

string

(query)

Units. Example: imperial. Possible values: standard, metric, and imperial. When you do not use the unit

imperial

lang

string

(query)

Language. Example: en. You can use lang parameter to get the output in your language. We support the following languages Bulgarian - bg, Catalan - ca, Czech - cz, German - de, Greek - el, English - en, Persian (Farsi) - fa, Fi Hungarian - hu, Italian - it, Japanese - ja, Korean - kr, Latvian - la, Lithuanian - lt, Macedonian - mk Russian - ru, Swedish - se, Slovak - sk, Slovenian - sl, Spanish - es, Turkish - tr, Ukrainian - ua, Viet zh_tw.

en

mode

string

(query)

Mode. Example: html. Determines the format of the response. Possible values are xml and html. If the mode paramet

json

API

string

(query)

Write there your API key from OpenWeatherMap

65cc3d7af05e58d0d6aed73242c949f8

Execute

Curl

```
curl -X 'GET' \
'https://api.openweathermap.org/data/2.5/weather?q=London&units=imperial&lang=en&mode=json&API=65cc3d7af05e58d0d6aed73242c949f8' \
-H 'accept: application/json'
```

Request URL

https://api.openweathermap.org/data/2.5/weather?q=London&units=imperial&lang=en&mode=json&API=65cc3d7af05e58d0d6aed73242c949f8

Server response

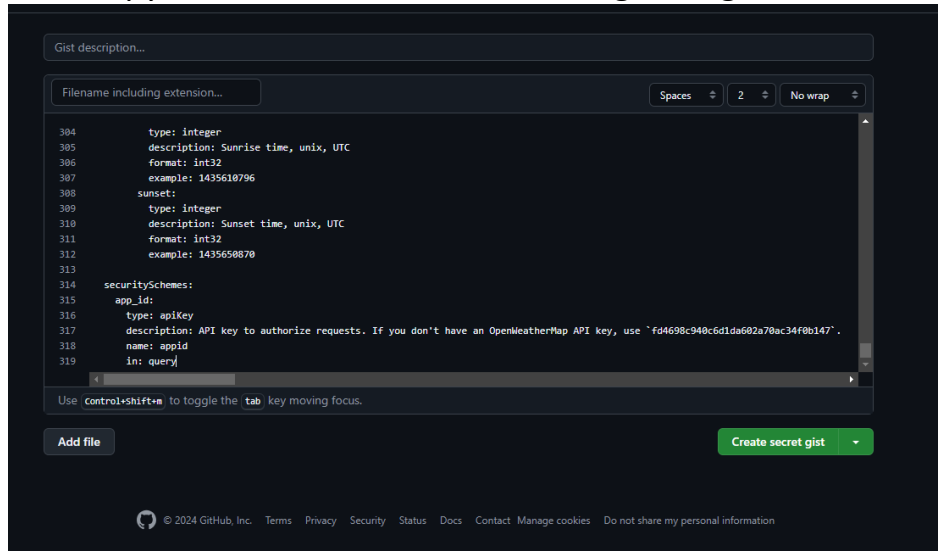
Code	Details
------	---------

Undocumented	<div>Failed to fetch. Possible Reasons:<ul style="list-style-type: none">CORSNetwork FailureURL scheme must be "http" or "https" for CORS request.</div>
--------------	--

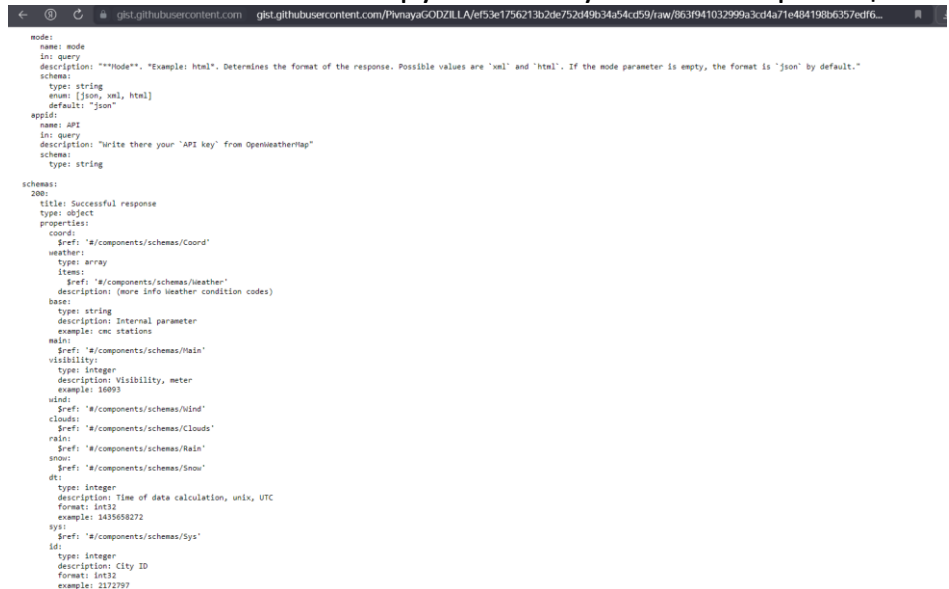
Responses

Code	Description	Links
200	<div>Successful response</div> <div>Media type: application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "coord": { "lon": 145.77, "lat": -16.92 }, "weather": [{ "id": 803, "main": "Clouds", "description": "broken clouds", "icon": "04n" }], "base": "cmc stations", "main": { "temp": 293.25, "pressure": 1019, "humidity": 83, "temp_min": 289.62, "temp_max": 295.37, "sea_level": 984, "grnd_level": 990 } }</pre></div>	No link
404	<div>Not found response</div> <div>Media type: text/plain</div>	No link

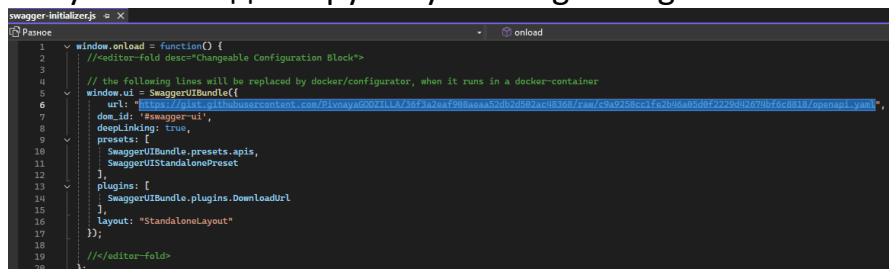
8. Копируем наш код и добавляем на github gist.



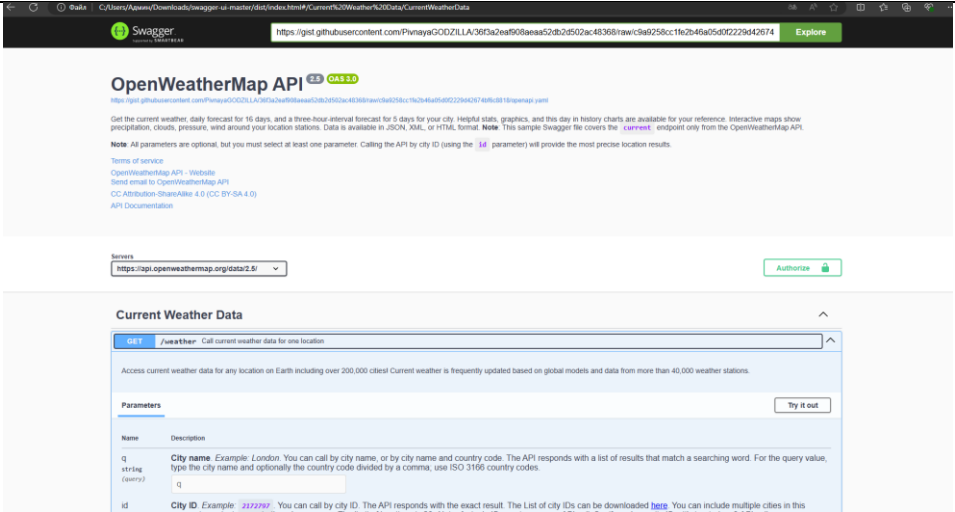
9. Нажимаем raw и копируем ссылку с чистой страницы.



10. В swagger-initializer.js меняем строку url на нашу, что мы получили в ходе загрузки yaml на github gist.



11. Удостоверяемся, что все работает

	
Результат	В ходе очень трудной работы мы научились использовать swagger-editor с оренарі, научились писать код для оренарі, и попрактиковались в поиске иголки в стоге сена (найти одну строку, среди множества файлов и их строк)
Оценка	

Листинг

Номер 7

```
{
  "coord": {
    "lon": 145.77,
    "lat": -16.92
  },
  "weather": [
    {
      "id": 803,
      "main": "Clouds",
      "description": "broken clouds",
      "icon": "04n"
    }
  ],
  "base": "cmc stations",
  "main": {
```

```
"temp": 293.25,  
"pressure": 1019,  
"humidity": 83,  
"temp_min": 289.82,  
"temp_max": 295.37,  
"sea_level": 984,  
"grnd_level": 990  
},  
"visibility": 16093,  
"wind": {  
  "speed": 5.1,  
  "deg": 150  
},  
"clouds": {  
  "all": 75  
},  
"rain": {  
  "3h": 3  
},  
"snow": {  
  "3h": 6  
},  
"dt": 1435658272,  
"sys": {  
  "type": 1,  
  "id": 8166,  
  "message": 0.0166,  
  "country": "AU",  
  "sunrise": 1435610796,  
  "sunset": 1435650870  
},  
"id": 2172797,  
"name": "Cairns",
```


"cod": 200

}

Homep 6

openapi: "3.0.2"

info:

title: "OpenWeatherMap API"

description: "Get the current weather, daily forecast for 16 days, and a three-hour-interval forecast for 5 days for your city. Helpful stats, graphics, and this day in history charts are available for your reference. Interactive maps show precipitation, clouds, pressure, wind around your location stations. Data is available in JSON, XML, or HTML format. **Note**: This sample Swagger file covers the `current` endpoint only from the OpenWeatherMap API. **Note**: All parameters are optional, but you must select at least one parameter. Calling the API by city ID (using the `id` parameter) will provide the most precise location results."

version: "2.5"

termsOfService: "https://openweathermap.org/terms"

contact:

name: "OpenWeatherMap API"

url: "https://openweathermap.org/api"

email: "notsaid@gmail.com"

license:

name: "CC Attribution-ShareAlike 4.0 (CC BY-SA 4.0)"

url: "https://openweathermap.org/price"

servers:

- url: "https://api.openweathermap.org/data/2.5/"

externalDocs:

description: API Documentation

url: https://openweathermap.org/api

paths:

/weather:

get:

tags:

- Current Weather Data

summary: "Call current weather data for one location"

description: "Access current weather data for any location on Earth including over 200,000 cities! Current weather is frequently updated based on global models and data from more than 40,000 weather stations."

operationId: CurrentWeatherData

parameters:

- \$ref: '#/components/parameters/q'
- \$ref: '#/components/parameters/id'
- \$ref: '#/components/parameters/lat'
- \$ref: '#/components/parameters/lon'
- \$ref: '#/components/parameters/zip'
- \$ref: '#/components/parameters/units'
- \$ref: '#/components/parameters/lang'
- \$ref: '#/components/parameters/mode'
- \$ref: '#/components/parameters/api'

responses:

200:

description: Successful response

content:

application/json:

schema:

\$ref: '#/components/schemas/200'

404:

description: Not found response

content:

text/plain:

schema:

title: Weather not found

type: string

example: Not found

components:

parameters:

q:

name: q

in: query

description: "***City name**". *Example: London*. You can call by city name, or by city name and country code. The API responds with a list of results that match a searching word. For the query value, type the city name and optionally the country code divided by a comma; use ISO 3166 country codes."

schema:

type: string

id:

name: id

in: query

description: "***City ID**". *Example: `2172797`*. You can call by city ID. The API responds with the exact result. The List of city IDs can be downloaded [here](http://bulk.openweathermap.org/sample/). You can include multiple cities in this parameter — just separate them by commas. The limit of locations is 20. *Note: A single ID counts as a one API call. So, if you have city IDs, it's treated as 3 API calls.*"

schema:

type: string

lat:

name: lat

in: query

description: "***Latitude**". *Example: 35*. The latitude coordinate of the location of your interest. Must use with `lon`."

schema:

type: string

lon:

name: lon

in: query

description: "***Longitude**". *Example: 139*. Longitude coordinate of the location of your interest. Must use with `lat`."

schema:

type: string

zip:

name: zip

in: query

description: **Zip code**. Search by zip code. *Example: 95050,us*. Please note that if the country is not specified, the search uses USA as a default."

schema:

type: string

units:

name: units

in: query

description: **Units**. *Example: imperial*. Possible values: `standard`, `metric`, and `imperial`. When you do not use the `units` parameter, the format is `standard` by default.'

schema:

type: string

enum: [standard, metric, imperial]

default: "imperial"

lang:

name: lang

in: query

description: **Language**. *Example: en*. You can use lang parameter to get the output in your language. We support the following languages that you can use with the corresponded lang values: Arabic - `ar`, Bulgarian - `bg`, Catalan - `ca`, Czech - `cz`, German - `de`, Greek - `el`, English - `en`, Persian (Farsi) - `fa`, Finnish - `fi`, French - `fr`, Galician - `gl`, Croatian - `hr`, Hungarian - `hu`, Italian - `it`, Japanese - `ja`, Korean - `kr`, Latvian - `la`, Lithuanian - `lt`, Macedonian - `mk`, Dutch - `nl`, Polish - `pl`, Portuguese - `pt`, Romanian - `ro`, Russian - `ru`, Swedish - `se`, Slovak - `sk`, Slovenian - `sl`, Spanish - `es`, Turkish - `tr`, Ukrainian - `ua`, Vietnamese - `vi`, Chinese Simplified - `zh_cn`, Chinese Traditional - `zh_tw`.'

schema:

type: string

enum: [ar, bg, ca, cz, de, el, en, fa, fi, fr, gl, hr, hu, it, ja, kr, la, lt, mk, nl, pl, pt, ro, ru, se, sk, sl, es, tr, ua, vi, zh_cn, zh_tw]

default: "en"

mode:

name: mode

in: query

description: "***Mode**". *Example: html*. Determines the format of the response. Possible values are `xml` and `html`. If the mode parameter is empty, the format is `json` by default."

schema:

type: string

enum: [json, xml, html]

default: "json"

api:

name: API

in: query

description: "Write there your `API key` from OpenWeatherMap"

schema:

type: string

schemas:

200:

title: Successful response

type: object

properties:

coord:

\$ref: '#/components/schemas/Coord'

weather:

type: array

items:

\$ref: '#/components/schemas/Weather'

description: (more info Weather condition codes)

base:

type: string

description: Internal parameter

example: cmc stations

main:

\$ref: '#/components/schemas/Main'

visibility:

type: integer

description: Visibility, meter

example: 16093

wind:

\$ref: '#/components/schemas/Wind'

clouds:

\$ref: '#/components/schemas/Clouds'

rain:

\$ref: '#/components/schemas/Rain'

snow:

\$ref: '#/components/schemas/Snow'

dt:

type: integer

description: Time of data calculation, unix, UTC

format: int32

example: 1435658272

sys:

\$ref: '#/components/schemas/Sys'

id:

type: integer

description: City ID

format: int32

example: 2172797

name:

type: string

example: Cairns

cod:

type: integer

description: Internal parameter

format: int32

example: 200

Coord:

title: Coord

type: object

properties:

lon:

type: number

description: City geo location, longitude

example: 145.77000000000001

lat:

type: number

description: City geo location, latitude

example: -16.920000000000002

Weather:

title: Weather

type: object

properties:

id:

type: integer

description: Weather condition id

format: int32

example: 803

main:

type: string

description: Group of weather parameters (Rain, Snow, Extreme etc.)

example: Clouds

description:

type: string

description: Weather condition within the group

example: broken clouds

icon:

type: string

description: Weather icon id

example: 04n

Main:

title: Main

type: object

properties:

temp:

type: number

description: 'Temperature. Unit Default: Kelvin, Metric: Celsius, Imperial: Fahrenheit.'

example: 293.25

pressure:

type: integer

description: Atmospheric pressure (on the sea level, if there is no sea_level or grnd_level data),
hPa

format: int32

example: 1019

humidity:

type: integer

description: Humidity, %

format: int32

example: 83

temp_min:

type: number

description: 'Minimum temperature at the moment. This is deviation from current temp that is possible for large cities and megalopolises geographically expanded (use these parameter optionally). Unit Default: Kelvin, Metric: Celsius, Imperial: Fahrenheit.'

example: 289.81999999999999

temp_max:

type: number

description: 'Maximum temperature at the moment. This is deviation from current temp that is possible for large cities and megalopolises geographically expanded (use these parameter optionally). Unit Default: Kelvin, Metric: Celsius, Imperial: Fahrenheit.'

example: 295.37

sea_level:

type: number

description: Atmospheric pressure on the sea level, hPa

example: 984

grnd_level:

type: number

description: Atmospheric pressure on the ground level, hPa

example: 990

Wind:

title: Wind

type: object

properties:

speed:

type: number

description: 'Wind speed. Unit Default: meter/sec, Metric: meter/sec, Imperial: miles/hour.'

example: 5.0999999999999996

deg:

type: integer

description: Wind direction, degrees (meteorological)

format: int32

example: 150

Clouds:

title: Clouds

type: object

properties:

all:

type: integer

description: Cloudiness, %

format: int32

example: 75

Rain:

title: Rain

type: object

properties:

3h:

type: integer

description: Rain volume for the last 3 hours

format: int32

example: 3

Snow:

title: Snow

type: object

properties:

3h:

type: number

description: Snow volume for the last 3 hours

example: 6

Sys:

title: Sys

type: object

properties:

type:

type: integer

description: Internal parameter

format: int32

example: 1

id:

type: integer

description: Internal parameter

format: int32

example: 8166

message:

type: number

description: Internal parameter

example: 0.0166

country:

type: string

description: Country code (GB, JP etc.)

example: AU

sunrise:

type: integer

description: Sunrise time, unix, UTC

format: int32

example: 1435610796

sunset:

type: integer

description: Sunset time, unix, UTC

format: int32

example: 1435650870

securitySchemes:

app_id:

type: apiKey

description: API key to authorize requests. If you don't have an OpenWeatherMap API key, use
`fd4698c940c6d1da602a70ac34f0b147`.

name: appid

in: query