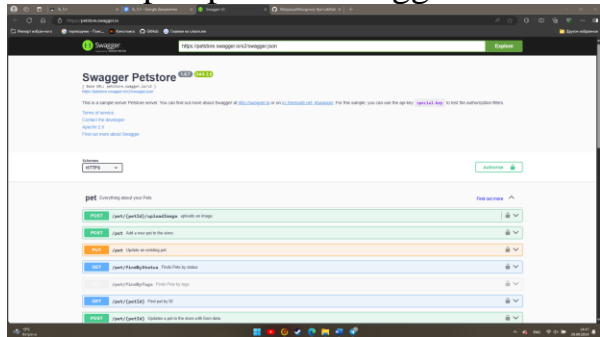
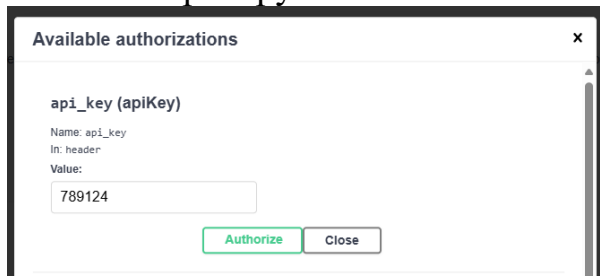
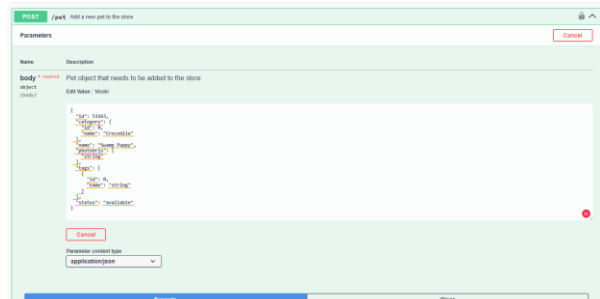
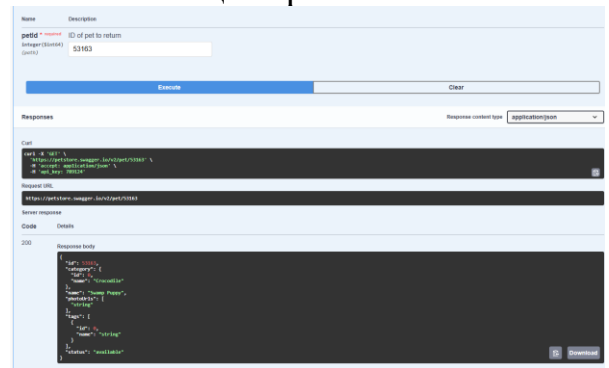
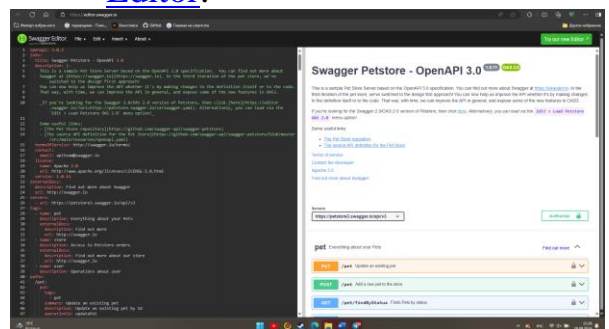


Тема лабораторной работы	lb_5_1
Выполняющий	Мозговой Илья
Помогающий	
Ход работы	<p>1. Заходим на https://petstore.swagger.io/</p>  <p>789124</p> <p>2. Авторизируемся на сайте</p>  <p>3. В post /pet создаем нового питомца</p>  <pre> Curl curl -X 'POST' \ 'https://petstore.swagger.io/v2/pet' \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -d '{ "id": 53163, "category": { "id": 0, "name": "Crocodile" }, "name": "Swamp Puppy", "photoUrls": ["string"], "tags": [{ "id": 0, "name": "string" }], "status": "available" }'</pre>

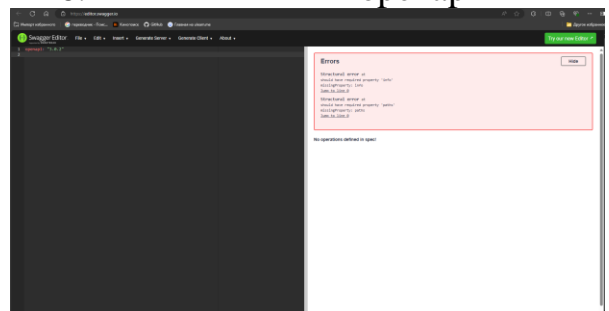
4. Пробуем получить нашего питомца через ID



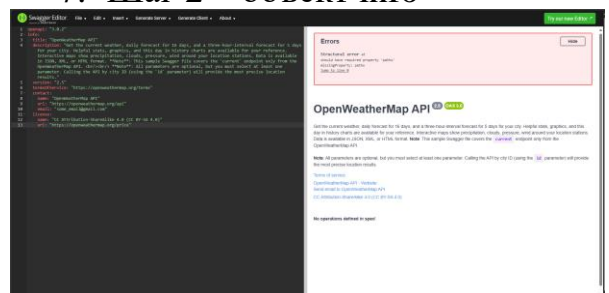
5. Создаем файл спецификации OpenAPI. Открываем [Swagger Editor](#).



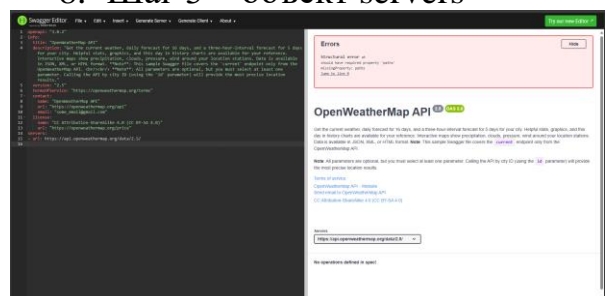
6. Шаг 1 – объект openapi



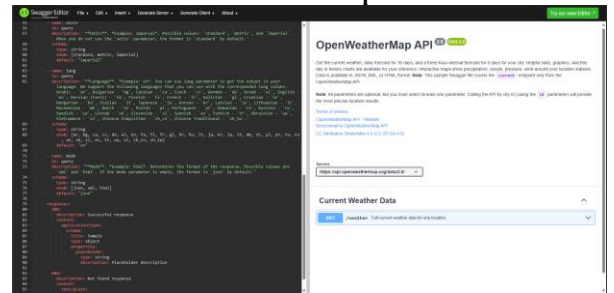
7. Шаг 2 – объект info



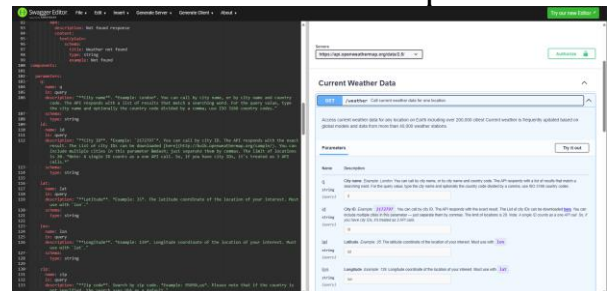
8. Шаг 3 – объект servers



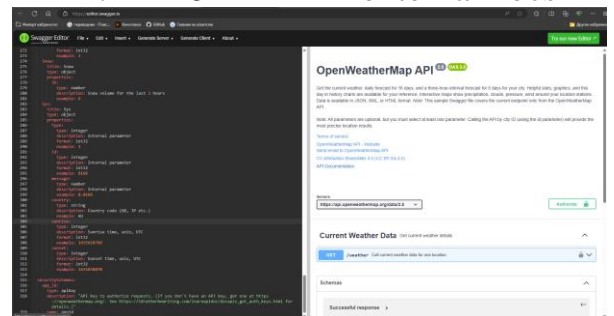
9. Шаг 4 – объект paths



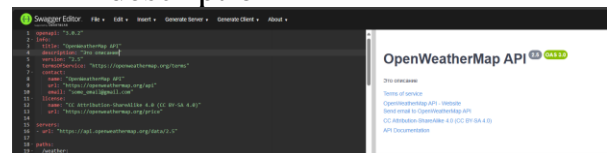
10. Шаг 5 – объект components



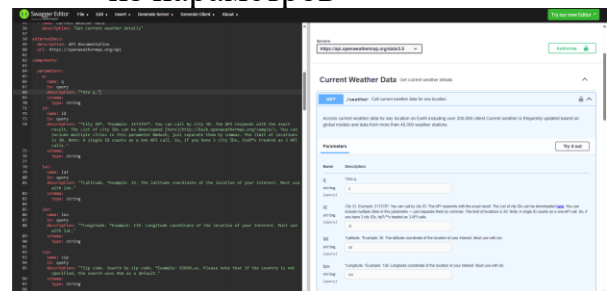
11. Шаг 8 – объект externalDocs



12. Вносим изменение в свойство description



13. Изменяем description одного из параметров

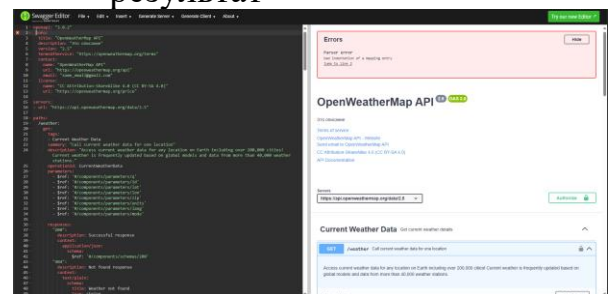


14. Указатель \$ref в объекте response ссылается на схему успешного завершения программы

```
26- operation: Get current weather data
27- parameters:
28-   - $ref: '#/components/parameters/q'
29-   - $ref: '#/components/parameters/id'
30-   - $ref: '#/components/parameters/lat'
31-   - $ref: '#/components/parameters/lon'
32-   - $ref: '#/components/parameters/zip'
33-   - $ref: '#/components/parameters/units'
34-   - $ref: '#/components/parameters/lang'
35-
36- responses:
37-   "200":
38-     description: Successful response
39-     content:
40-       application/json:
41-         schema:
42-           $ref: '#/components/schemas/200'
43-   "404":
44-     description: Not found response
45-     content:
46-       text/plain:
47-         schema:
48-           title: Weather not found
49-           type: string
50-           example: Not found
51-
52- security:
53-   - app_id: []
54-
55- tags:
56-   - name: Current Weather Data
57-     description: "Get current weather details"
58-
59- externalDocs:
```

```
126- Schemas:
127-   "200":
128-     title: Successful response
129-     type: object
130-     properties:
131-       coord:
132-         $ref: '#/components/schemas/Coord'
133-       weather:
134-         type: array
135-         items:
136-           $ref: '#/components/schemas/Weather'
137-       description: (more info weather condition codes)
138-     base:
139-       type: string
140-       description: Internal parameter
141-       example: cmc stations
142-     main:
143-       $ref: '#/components/schemas/Main'
144-     visibility:
145-       type: integer
146-       description: Visibility, meter
147-       example: 16003
148-     wind:
149-       $ref: '#/components/schemas/Wind'
150-     clouds:
151-       $ref: '#/components/schemas/Clouds'
152-     rain:
153-       $ref: '#/components/schemas/Rain'
154-     snow:
155-       $ref: '#/components/schemas/Snow'
156-     dt:
157-       type: integer
158-       description: Time of data calculation, unix, UTC
159-       format: int32
160-       example: 1415658272
161-     sys:
162-       $ref: '#/components/schemas/Sys'
163-     id:
164-       type: integer
165-       description: City ID
166-       format: int32
167-       example: 2172797
168-     name:
169-       type: string
170-       example: Cairns
171-     cod:
172-       type: integer
173-       description: Internal parameter
174-       format: int32
175-       example: 200
176-   Coord:
```

15. Поставим лишний пробел перед info и посмотрим на результат



16. Пробуем выполнить

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 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 . 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 3 city IDs, it's 3 API calls.
lat	Latitude. Example: 35. The latitude coordinate of the location of your interest. Must use with lon.
lon	Longitude. Example: 139. The 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.
units	Units. Example: imperial. Possible values: standard, metric, and imperial. When you do not use the units parameter, the format is standard by default.

Responses

Curl

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

Request URL

```
https://api.openweathermap.org/data/2.5/weather?units=imperial&lang=en&mode=json
```

Server response

Code **Details**

Failed to fetch.
Possible Reasons:

- CORS
- Network Failure
- URL scheme must be "http" or "https" for CORS request.

Responses

Code	Description	Links
200	Successful response	No links

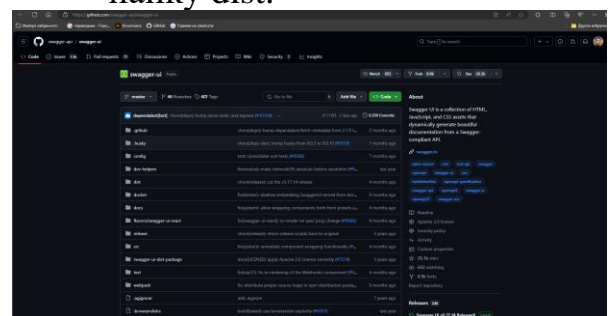
Code **Description** **Links**

200	Successful response	No links
-----	---------------------	----------

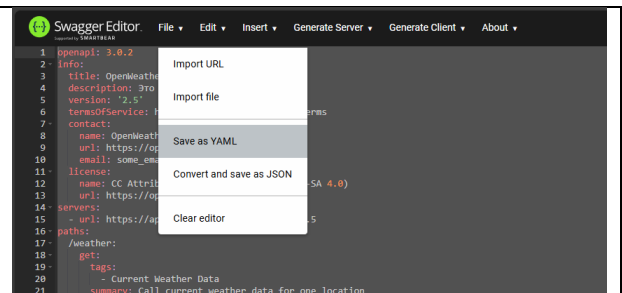
Example Value **Schema**

```
{
  "coord": {
    "lon": 140.77,
    "lat": 35.68
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 293.15,
    "pressure": 1013,
    "humidity": 65,
    "temp_min": 289.15,
    "temp_max": 297.15,
    "sea_level": 0,
    "grnd_level": 0
  },
  "wind": {
    "speed": 3.6,
    "deg": 140
  },
  "clouds": {
    "all": 0
  },
  "dt": 1516731123,
  "sys": {
    "type": 1,
    "id": 5076,
    "country": "JP",
    "sunrise": 1516728000,
    "sunset": 1516744800
  },
  "timezone": 32400,
  "id": 1850876,
  "name": "Tokyo",
  "state": "Tokyo"
}
```

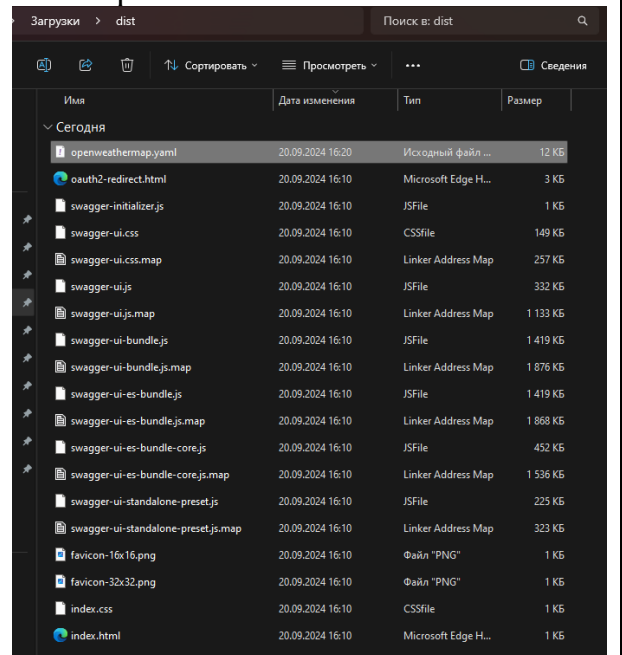
17. Открываем GitHub OpenAPI project, скачиваем ZIP и извлекаем, из него достаем папку dist.



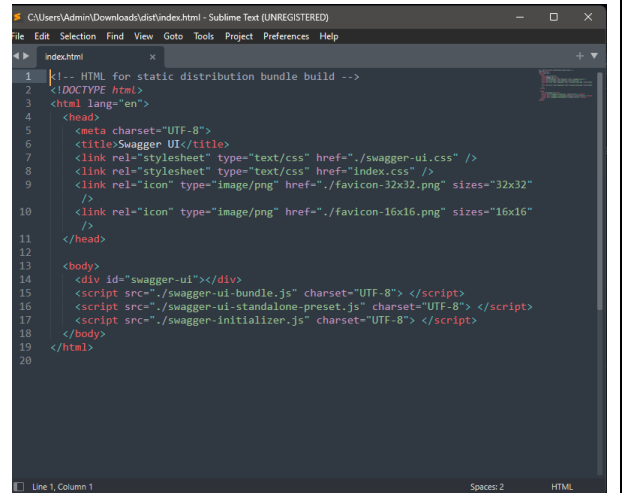
18. Сохраняем как YAML



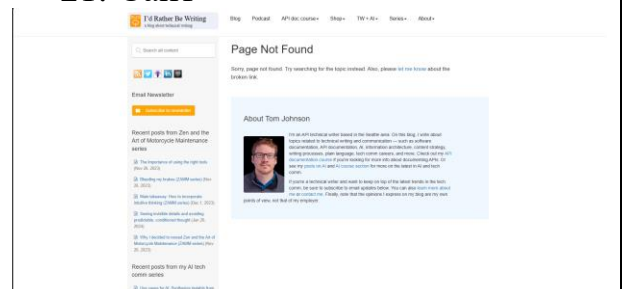
19. Закидываем сохраненный файл в dist



20. Sublime text



21. Сайт



Результат	В ходе данной лабораторной работы мы научились работать с editor swagger, работать со спецификацией openApi.
-----------	--

Критерий оценивания

Отлично	Четко, лаконично, по факту, гитхаб на ура
Хорошо	Небольшие корректировки
Удовлетворительно	С вопросами, и комментариями
Неудовлетворительно	Ничего непонятно

Листинг 1

openapi: "3.0.2"

info:

title: "OpenWeatherMap API"

description: "Это описание"

version: "2.5"

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

contact:

name: "OpenWeatherMap API"

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

email: "some_email@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"

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'**

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

security:

- app_id: []

tags:

- name: Current Weather Data

description: "Get current weather details"

externalDocs:

description: API Documentation

url: <https://openweathermap.org/api>

components:

parameters:

q:

name: q

in: query

description: "*ᐅto q."

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 3 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 json, xml, and html. If the mode parameter is empty, the format is json by default."*

schema:

type: string

enum: [json, xml, html]

default: "json"

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 API key, get one at <https://openweathermap.org/>. See https://idratherbewing.com/learnapidoc/docapis_get_auth_keys.html for details.)"

name: appid

in: query