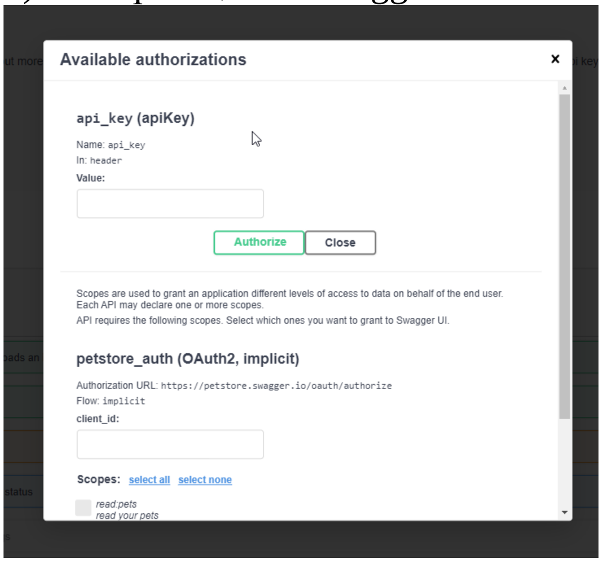
Тема лабораторной работы	lb_5.1
Выполняющий	Ходаев Лев ис221
Помогающий	Стоцкий Владислав ис221
Ход работы	1) Авторизация в Swagger



Available authorizations

api_key (apiKey)

Name: api_key
In: header
Value:

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.
API requires the following scopes. Select which ones you want to grant to Swagger UI.

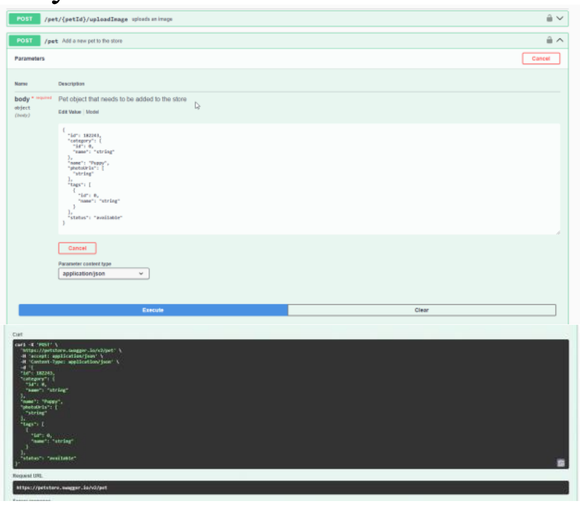
petstore_auth (OAuth2, implicit)

Authorization URL: https://petstore.swagger.io/oauth/authorize
Flow: implicit
client_id:

Scopes: select all select none

read:pets
read your pets

2) работаем с pet, называем Puppy и получаем ответ в Curl



POST /pet/{petId}/uploadImage uploads an image

POST /pet Add a new pet to the store

Parameters

Name	Description
body * required object (body)	Pet object that needs to be added to the store

Edit Value | Model

```
{
  "id": 182243,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "Puppy",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type: application/json

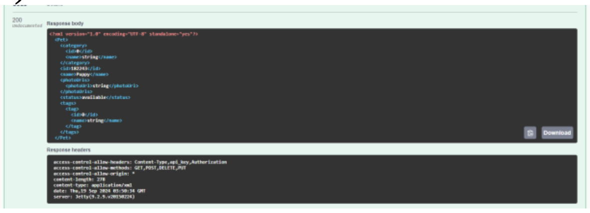
Curl

```
curl -X 'POST' \
  'https://petstore.swagger.io/v2/pet' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 182243,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "Puppy",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}'
```

Request URL

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

3) ответ в xml



200
Undocumented

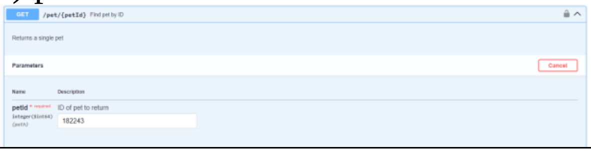
Response body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Pet>
  <category>
    <id>0</id>
    <name>string</name>
  </category>
  <id>182243</id>
  <name>Puppy</name>
  <photoUrls>
    <photoUrl>string</photoUrl>
  </photoUrls>
  <status>available</status>
  <tags>
    <tag>
      <id>0</id>
      <name>string</name>
    </tag>
  </tags>
</Pet>
```

Response headers

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-length: 278
content-type: application/xml
date: Thu,19 Sep 2024 03:50:34 GMT
server: Jetty(9.2.9.v20150224)
```

4) petid

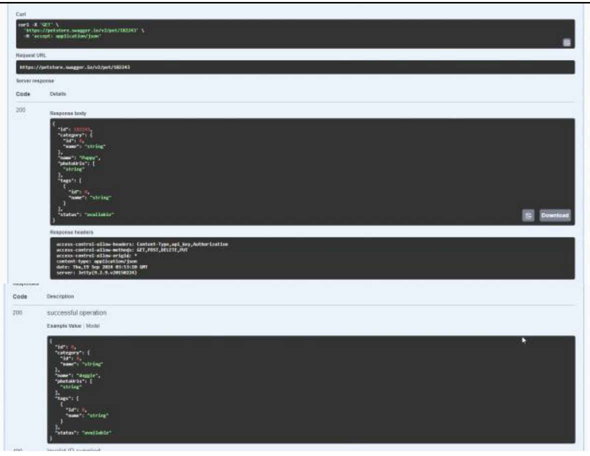


GET /pet/{petId} Find pet by ID

Returns a single pet

Parameters

Name	Description
petId * required integer($int64) (path)	ID of pet to return 182243


Curl

```
curl -X 'GET' \
  'https://petstore.swagger.io/v2/pet/182243' \
  -H 'accept: application/json'
```

Request URL

```
https://petstore.swagger.io/v2/pet/182243
```

Server response

Code | Details

200

Response body

```
{
  "id": 182243,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "Puppy",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Response headers

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Thu,19 Sep 2024 05:53:10 GMT
server: Jetty(9.2.9.v20150224)
```

Code | Description

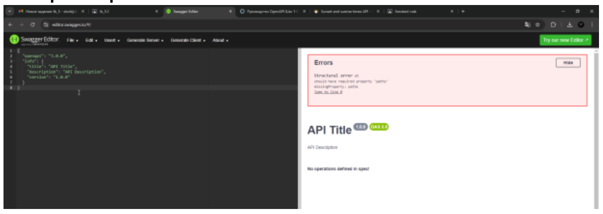
200 successful operation

Example Value | Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

5) редактируем файл спецификации OPenAAPI

1. openapi



```
{
  "openapi": "3.0.0",
  "info": {
    "title": "API Title",
    "description": "API Description",
    "version": "1.0.0"
  }
}
```

Errors

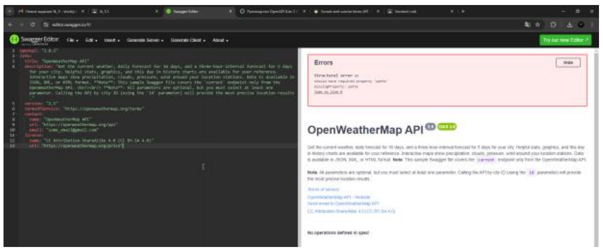
```
Structural error at
should have required property 'paths'
missingProperty: paths
Jump to line 0
```

API Title 1.0.0 OAS 3.0

API Description

No operations defined in spec!

2. info



```
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. <br/><br/> **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: "some_email@gmail.com"
  license:
    name: "CC Attribution-ShareAlike 4.0 (CC BY-SA 4.0)"
    url: "https://openweathermap.org/price"
```

Errors

```
Structural error at
should have required property 'paths'
missingProperty: paths
Jump to line 0
```

OpenWeatherMap API 2.5 OAS 3.0

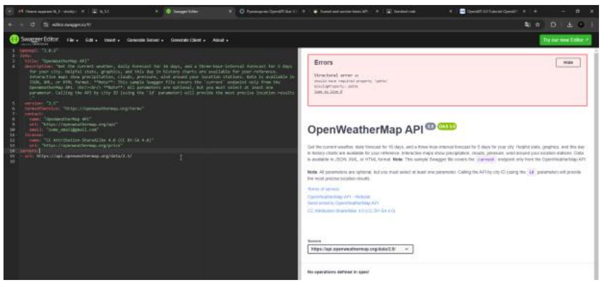
Get the current weather, daily forecast for 16 days, and a three-hour-interval forecast for 5 days for your city. Helpful stats, graphics, and this day in history charts are available for your reference. Interactive maps show precipitation, clouds, pressure, wind around your location stations. Data is 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.

Terms of service
OpenWeatherMap API - Website
Send email to OpenWeatherMap API
CC Attribution-ShareAlike 4.0 (CC BY-SA 4.0)

No operations defined in spec!

3. servers



```
servers:
  - url: https://api.openweathermap.org/data/2.5/
```

Errors

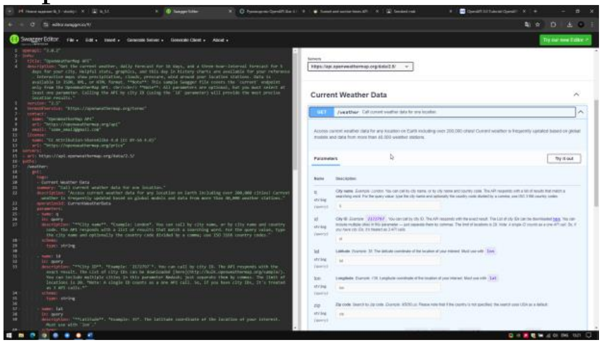
```
Structural error at
should have required property 'paths'
missingProperty: paths
Jump to line 0
```

OpenWeatherMap API 2.5 OAS 3.0

https://api.openweathermap.org/data/2.5/

No operations defined in spec!

4. paths



```
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:
        - 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
        - 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 n API calls.*"
          schema:
            type: string
        - name: lat
          in: query
          description: "**Latitude**. *Example: 35*. The latitude coordinate of the location of your interest. Must use with `lon`."
```

https://api.openweathermap.org/data/2.5/

Current Weather Data

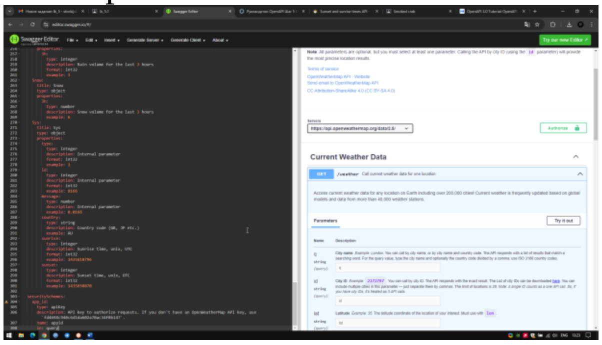
GET /weather Call current weather data for one location

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.

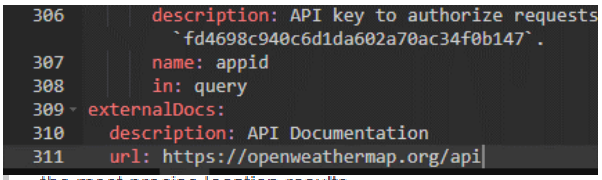
Parameters

Name	Description
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 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 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 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 n API calls.
lat string (query)	Latitude. Example: 35. The latitude coordinate of the location of your interest. Must use with `lon`.
lon string (query)	Longitude. Example: 139. Longitude coordinate of the location of your interest. Must use with `lat`.
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 default.

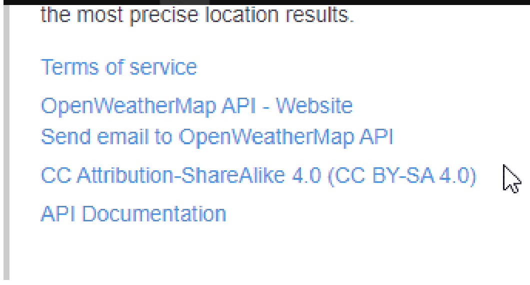
5. components



6. externalDocs



```
        description: API key to authorize requests
          `fd4698c940c6d1da602a70ac34f0b147`.
        name: appid
        in: query
externalDocs:
  description: API Documentation
  url: https://openweathermap.org/api
```



the most precise location results.

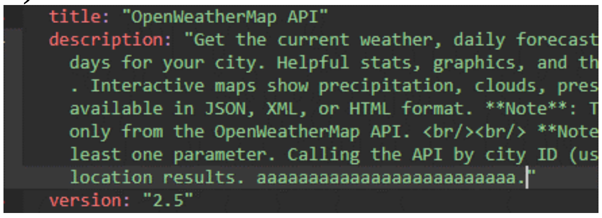
Terms of service

OpenWeatherMap API - Website
Send email to OpenWeatherMap API

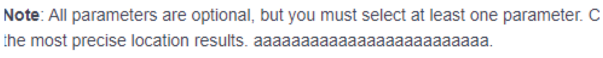
CC Attribution-ShareAlike 4.0 (CC BY-SA 4.0)

API Documentation

7) изменения в info

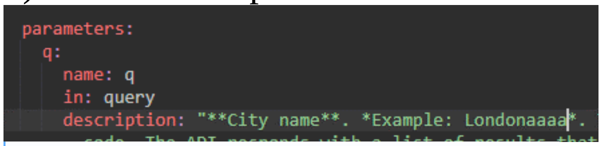


```
  title: "OpenWeatherMap API"
  description: "Get the current weather, daily forecast
    days for your city. Helpful stats, graphics, and th
    . Interactive maps show precipitation, clouds, pres
    available in JSON, XML, or HTML format. **Note**: T
    only from the OpenWeatherMap API. <br/><br/> **Note
    least one parameter. Calling the API by city ID (us
    location results. aaaaaaaaaaaaaaaaaaaaaaaaa."
  version: "2.5"
```

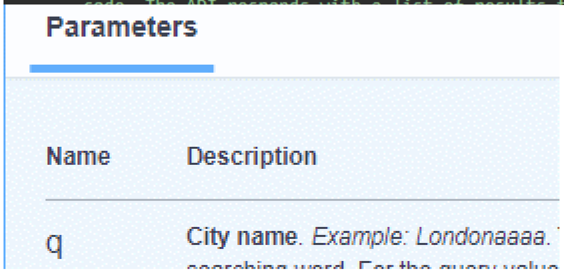


Note: All parameters are optional, but you must select at least one parameter. C
the most precise location results. aaaaaaaaaaaaaaaaaaaaaaaaa.

8) изменения в parameters



```
parameters:
  q:
    name: q
    in: query
    description: "**City name**. *Example: Londonaaaa*.
```

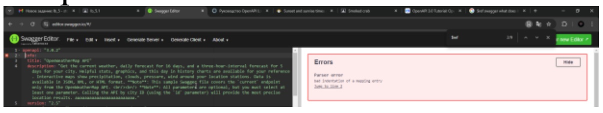


Parameters

Name	Description
q	City name. *Example: Londonaaaa.*

9) указатель $ref служит для ссылки на другие вещи, такие как сайты и тд.

10) добавление пробела и его исправление

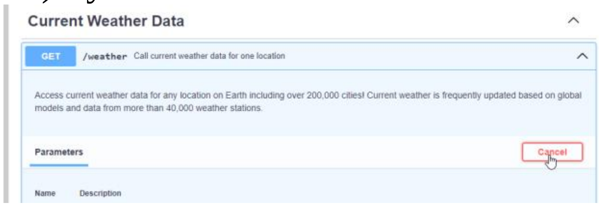


```
openapi: "3.0.2"
info:
  title: "OpenWeatherMap API"
  description: "Get the current weather, daily forecast for 16 days, and a three-hour-interval forecast for 5 days for your city. Helpful stats, graphics, and this day in history charts are available for your reference. Interactive maps show precipitation, 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. <br/><br/> **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. aaaaaaaaaaaaaaaaaaaaaaaaa."
  version: "2.5"
```

Errors

Parser error
bad indentation of a mapping entry
Jump to line 2

11) try it out

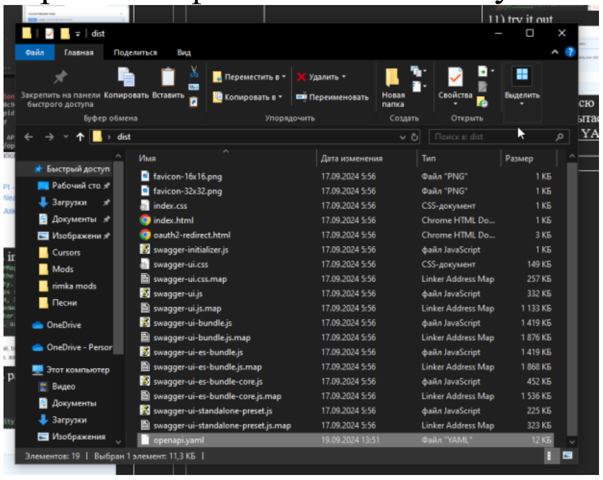


Current Weather Data

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.

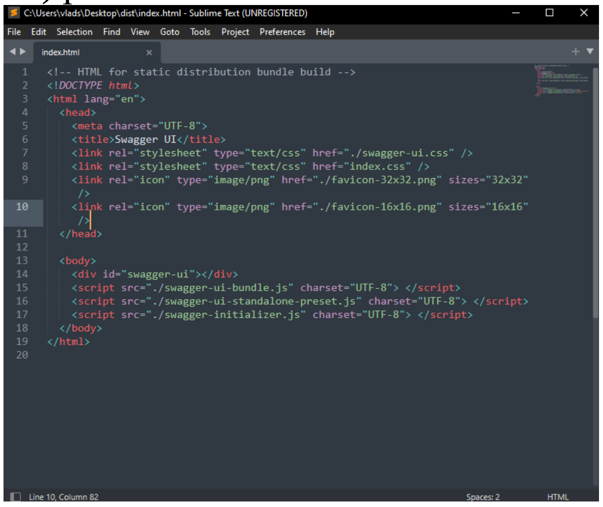
12) сохраняем всю работу в YAML, скачиваем zip, вытаскиваем dist и переносим файл YAML в папку dist



Имя	Дата изменения	Тип	Размер
favicon-16x16.png	17.09.2024 5:56	Файл "PNG"	1 КБ
favicon-32x32.png	17.09.2024 5:56	Файл "PNG"	1 КБ
index.css	17.09.2024 5:56	CSS-документ	1 КБ
index.html	17.09.2024 5:56	Chrome HTML Do...	1 КБ
oauth2-redirect.html	17.09.2024 5:56	Chrome HTML Do...	3 КБ
swagger-initializer.js	17.09.2024 5:56	файл JavaScript	1 КБ
swagger-ui.css	17.09.2024 5:56	CSS-документ	149 КБ
swagger-ui.css.map	17.09.2024 5:56	Linker Address Map	257 КБ
swagger-ui.js	17.09.2024 5:56	файл JavaScript	332 КБ
swagger-ui.js.map	17.09.2024 5:56	Linker Address Map	1 133 КБ
swagger-ui-bundle.js	17.09.2024 5:56	файл JavaScript	1 419 КБ
swagger-ui-bundle.js.map	17.09.2024 5:56	Linker Address Map	1 876 КБ
swagger-ui-es-bundle.js	17.09.2024 5:56	файл JavaScript	1 419 КБ
swagger-ui-es-bundle.js.map	17.09.2024 5:56	Linker Address Map	1 868 КБ
swagger-ui-es-bundle-core.js	17.09.2024 5:56	файл JavaScript	452 КБ
swagger-ui-es-bundle-core.js.map	17.09.2024 5:56	Linker Address Map	1 536 КБ
swagger-ui-standalone-preset.js	17.09.2024 5:56	файл JavaScript	225 КБ
swagger-ui-standalone-preset.js.map	17.09.2024 5:56	Linker Address Map	323 КБ
openapi.yaml	19.09.2024 13:51	Файл "YAML"	12 КБ

Элементов: 19 | Выбран 1 элемент: 11,3 КБ |

13) работа с Sublime Textt

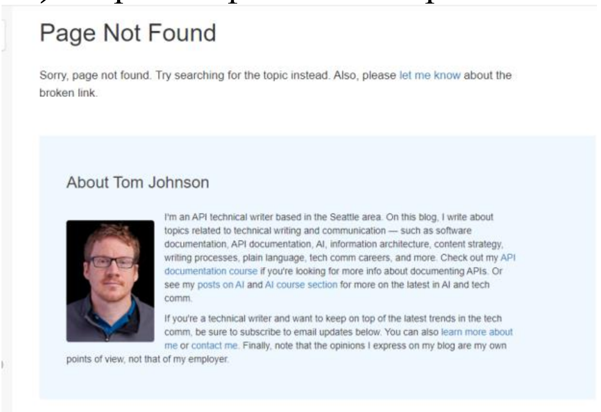


C:\Users\vlads\Desktop\dist\index.html - Sublime Text (UNREGISTERED)

```html
<!-- HTML for static distribution bundle build -->
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Swagger UI</title>
    <link rel="stylesheet" type="text/css" href="./swagger-ui.css" />
    <link rel="stylesheet" type="text/css" href="index.css" />
    <link rel="icon" type="image/png" href="./favicon-32x32.png" sizes="32x32"
    />
    <link rel="icon" type="image/png" href="./favicon-16x16.png" sizes="16x16"
    />
  </head>

  <body>
    <div id="swagger-ui"></div>
    <script src="./swagger-ui-bundle.js" charset="UTF-8"> </script>
    <script src="./swagger-ui-standalone-preset.js" charset="UTF-8"> </script>
    <script src="./swagger-initializer.js" charset="UTF-8"> </script>
  </body>
</html>
```

Line 10, Column 82 Spaces: 2 HTML

	14) открытие файла в интернете  Page Not Found Sorry, page not found. Try searching for the topic instead. Also, please let me know about the broken link. About Tom Johnson I'm an API technical writer based in the Seattle area. On this blog, I write about topics related to technical writing and communication — such as software documentation, API documentation, AI, information architecture, content strategy, writing processes, plain language, tech comm careers, and more. Check out my API documentation course if you're looking for more info about documenting APIs. Or see my posts on AI and AI course section for more on the latest in AI and tech comm. If you're a technical writer and want to keep on top of the latest trends in the tech comm, be sure to subscribe to email updates below. You can also learn more about me or contact me. Finally, note that the opinions I express on my blog are my own points of view, not that of my employer.
Результат	В ходе данной лабораторной работы мы научились работать с редактором Swagger, однако не вышло выполнить последние 2 шага т.к. в файле index.html нет нужной строчки кода, а также не открываются сайты для просмотра файла.

листинг	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. aaaaaaaaaaaaaaaaaaaaaaaaa."
	  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

components:

  parameters:
    q:
      name: q
      in: query
      description: "**City name**.
*Example: Londonaaaa*. 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 &mdash; 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"

  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:
```

```yaml
        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
externalDocs:
  description: API Documentation
  url: https://openweathermap.org/api
```