

# Лабораторна робота 3

**Тема.** Клієнт-серверна архітектура ПЗ. Створення RESTful API

**Мета :** ознайомитися з принципами клієнт-серверної архітектури та зрозуміти її роль у сучасних програмних системах. Навчитися створювати RESTful API для взаємодії між клієнтом і сервером. Закріпити практичні навички роботи з HTTP-запитами та відповідями. Розвинути вміння проєктувати та реалізовувати ендпоїнти для типових CRUD-операцій.

## 1 Завдання

Створюєм публічний репозиторій та додаєм у нього

- Файл json та записуєм у нього данні

```
pastu@Marusya MINGW64 /c/LW3_MiA_Pastukh (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   db.json

pastu@Marusya MINGW64 /c/LW3_MiA_Pastukh (main)
$ git commit -m "Json file adding"
[main 7fe637c] Json file adding
 1 file changed, 10 insertions(+)
 create mode 100644 db.json
```

На скриншоті ми бачим, що ми створили та закомітили файл джесон на нашо віддалений репозиторій

| Змн.      | Арк. | № докум.    | Підпис | Дата | ЛР.ОК24.ПІ231.24.03   |      |         |
|-----------|------|-------------|--------|------|-----------------------|------|---------|
| Розроб.   |      | Пастух М.М. |        |      | Kлієнт-серверна       | Літ. | Акр.    |
| Перевір.  |      | Когут В.С.  |        |      | архітектура ПЗ.       |      | Акрушин |
| Н. Контр. |      |             |        |      | Створення RESTful API | 1    | 39      |
| Затверд.  |      |             |        |      |                       |      | XПК     |

LW3\_MiA\_Pastukh

**Code**

1 Branch 0 Tags

Go to file Add file Code About

Mosska1 Update db.json 0134b94 · 37 minutes ago 3 Commits

README.md first commit 45 minutes ago

db.json Update db.json 37 minutes ago

README

Activity 0 stars 0 watching 0 forks

No releases published Create a new release

Packages No packages published Publish your first package

### Віддалений репозиторій

Далі запускаєм Fake Rest

Need more? Private repo, random values, REST+GraphQL... check our sponsor Mockend

My JSON Server beta Plans Free for OSS

LW3\_MiA\_Pastukh by mosska1

Sponsor Share Fork

Available resources

- users 2
- posts 2
- db db

You can use GET, POST, PUT, PATCH and DELETE. Changes aren't persisted between calls.

Source

[https://github.com/mosska1/LW3\\_MiA\\_Pastukh/blob/\[main\]master/db.json](https://github.com/mosska1/LW3_MiA_Pastukh/blob/[main]master/db.json)

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

|      |   |
|------|---|
| Арк. | 2 |
|------|---|

## Далі створюєм колекцію у Postman

The screenshot shows the Postman interface with a collection named 'LW3\_Fake\_REST'. The collection contains five requests:

- GET Get all users
- GET Get user by id
- GET Get posts
- GET Get users with filtration
- GET Get something

On the right side, there are statistics for the collection: 5 requests, 1 view, 0 forks, 0 watchers, and it was created by 'You' on September 28, 2025, at 1:20 PM by Maria Pastuh.

І ось усі виконані запити:

The screenshot shows the 'Get all users' request details in Postman. The URL is https://my-json-server.typicode.com/moshka1/LW3\_MIA\_Pastuh/users. The response body is a JSON array of two user objects:

```
[{"id": 1, "name": "Alice", "email": "alice@example.com", "role": "admin"}, {"id": 2, "name": "Bob", "email": "bob@example.com", "role": "user"}]
```

*Get all users*

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 3    |

LW3\_Fake\_REST / Get user by id

GET https://my-json-server.typicode.com/mosshka1/LW3\_MIA\_Pastukh/users/1

| Key   | Value             | Description |
|-------|-------------------|-------------|
| id    | 1                 |             |
| name  | Alice             |             |
| email | alice@example.com |             |
| role  | admin             |             |

Body Cookies Headers (21) Test Results

200 OK | 148 ms | 1.09 KB | Save Response

```
{
  "id": 1,
  "name": "Alice",
  "email": "alice@example.com",
  "role": "admin"
}
```

### Get user by id

LW3\_Fake\_REST / Get posts

GET https://my-json-server.typicode.com/mosshka1/LW3\_MIA\_Pastukh/posts?\_sort=title&\_order=asc&\_limit=1

| Key    | Value | Description |
|--------|-------|-------------|
| _sort  | title |             |
| _order | asc   |             |
| _limit | 1     |             |

Body Cookies Headers (23) Test Results

200 OK | 231 ms | 1.17 KB | Save Response

```
[
  {
    "id": 1,
    "title": "Hello REST",
    "userId": 1,
    "tags": [
      "intro",
      "rest"
    ]
  }
]
```

### Get posts

|      |      |          |        |      |                     |      |
|------|------|----------|--------|------|---------------------|------|
| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|      |      |          |        |      |                     | 4    |

The screenshot shows the Postman interface with a collection named "LW3\_Fake\_REST". A GET request is made to [https://my-json-server.typicode.com/mosshka1/LW3\\_MIA\\_Pastukh/posts?userId=2](https://my-json-server.typicode.com/mosshka1/LW3_MIA_Pastukh/posts?userId=2). The query parameters "userId" and "key" are set to "2". The response is a 200 OK status with a JSON body containing one item:

```

1 [ 
2   { 
3     "id": 2,
4     "title": "JSON Tips",
5     "userId": 2,
6     "tags": [
7       "json"
8     ]
9   }
10 ]

```

### Get users with filter

The screenshot shows the Postman interface with the same collection "LW3\_Fake\_REST". A GET request is made to [https://my-json-server.typicode.com/mosshka1/LW3\\_MIA\\_Pastukh/posts?q=REST](https://my-json-server.typicode.com/mosshka1/LW3_MIA_Pastukh/posts?q=REST). The query parameters "q" and "key" are set to "REST". The response is a 200 OK status with a JSON body containing one item:

```

1 [ 
2   { 
3     "id": 1,
4     "title": "Hello REST",
5     "userId": 1,
6     "tags": [
7       "intro",
8       "rest"
9     ]
10 }
11 ]

```

### Get find

|      |      |          |        |      |                     |      |
|------|------|----------|--------|------|---------------------|------|
| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|      |      |          |        |      |                     | 5    |

## Далі ми робим запити через консоль(в мому прикладі це Windows Powershell; порядок і вид запитів такий самий як і в Постмені)

```
PS C:\Users\pastu> curl.exe -i "https://my-json-server.typicode.com/mosshka1/LW3_Mia_Pastukh/users"
HTTP/1.1 200 OK
Date: Sun, 28 Sep 2025 10:31:47 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 193
Connection: keep-alive
access-control-allow-credentials: true
Cache-Control: no-cache
etag: W/"bf-sxvQHUiQzV1XKu/Gjd+V6aWL4"
expires: -1
nel: {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
pragma: no-cache
report-to: heroku-nel, endpoints:[{"url": "https://nel.herokuapp.com/reports?s=AcfJ33Nw9Bhj5GL60kk0HAp4ue5CoKh09rZh%2BuWANA%3D\u0026sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6\u0026ts=1759055507"}, {"url": "https://nel.herokuapp.com/reports?s=AcfJ33Nw9Bhj5GL60kk0HAp4ue5CoKh09rZh%2BuWANA%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055507"}]
reporting-endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=AcfJ33Nw9Bhj5GL60kk0HAp4ue5CoKh09rZh%2BuWANA%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055507"
Server: cloudflare
vary: Origin, Accept-Encoding
via: 2.0 heroku-router
x-content-type-options: nosniff
x-powered-by: Express
cf-cache-status: DYNAMIC
CF-RAY: 986292b4fb27ecea-WAW
alt-svc: h3=":443"; ma=86400
[{"id": 1, "name": "Alice", "email": "alice@example.com", "role": "admin"}, {"id": 2, "name": "Bob", "email": "bob@example.com", "role": "user"}]
```

### Усі користувачі

```
PS C:\Users\pastu> curl.exe -i "https://my-json-server.typicode.com/mosshka1/LW3_Mia_Pastukh/users/1"
HTTP/1.1 200 OK
Date: Sun, 28 Sep 2025 10:32:02 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 83
Connection: keep-alive
access-control-allow-credentials: true
Cache-Control: no-cache
etag: W/"53-95ftQ0WcNxof4yZGFZ48BVYCXHo"
expires: -1
nel: {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
pragma: no-cache
report-to: heroku-nel, endpoints:[{"url": "https://nel.herokuapp.com/reports?s=rDpqO297F5PB6ZWMk1JgjPt%2BmuaeYSDVpydMSUEsSUPw%3D\u0026sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6\u0026ts=1759055522"}, {"url": "https://nel.herokuapp.com/reports?s=rDpqO297F5PB6ZWMk1JgjPt%2BmuaeYSDVpydMSUEsSUPw%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055522"}]
reporting-endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=rDpqO297F5PB6ZWMk1JgjPt%2BmuaeYSDVpydMSUEsSUPw%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055522"
Server: cloudflare
vary: Origin, Accept-Encoding
via: 2.0 heroku-router
x-content-type-options: nosniff
x-powered-by: Express
cf-cache-status: DYNAMIC
CF-RAY: 986293172e578282-WAW
alt-svc: h3=":443"; ma=86400
{
  "id": 1,
  "name": "Alice",
  "email": "alice@example.com",
  "role": "admin"
}
```

### Один користувач

```
PS C:\Users\pastu> curl.exe -i "https://my-json-server.typicode.com/mosshka1/LW3_Mia_Pastukh/posts?userId=2"
HTTP/1.1 200 OK
Date: Sun, 28 Sep 2025 10:32:11 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 100
Connection: keep-alive
access-control-allow-credentials: true
Cache-Control: no-cache
etag: W/"6d-TSdiSlyo5YeEKK1XKAJ+e2yFg"
expires: -1
nel: {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
pragma: no-cache
report-to: heroku-nel, endpoints:[{"url": "https://nel.herokuapp.com/reports?s=lmlVhIBwB2ktAWIV232nXploCuCchdJhrhwC0keU1PM%3D\u0026sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6\u0026ts=1759055531"}, {"url": "https://nel.herokuapp.com/reports?s=lmlVhIBwB2ktAWIV232nXploCuCchdJhrhwC0keU1PM%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055531"}]
reporting-endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=lmlVhIBwB2ktAWIV232nXploCuCchdJhrhwC0keU1PM%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055531"
Server: cloudflare
vary: Origin, Accept-Encoding
via: 2.0 heroku-router
x-content-type-options: nosniff
x-powered-by: Express
cf-cache-status: DYNAMIC
CF-RAY: 9862934e6c9ce4d-WAW
alt-svc: h3=":443"; ma=86400
[{"id": 2, "title": "JSON Tips", "userId": 2, "tags": ["json"]}]
```

### Фільтр

| Змн. | Арк. | № докум. | Підпис | Дата | Арк. |
|------|------|----------|--------|------|------|
|      |      |          |        |      | 6    |

ЛР.ОК24.ПІ231.24.03

```

PS C:\Users\pastu> curl.exe -i "https://my-json-server.typicode.com/mosshkai/LW3_Mia_Pastukh/posts?_sort=title&_order=asc&_limit=1"
HTTP/1.1 200 OK
Date: Sun, 28 Sep 2025 10:32:18 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 116
Connection: keep-alive
access-control-allow-credentials: true
access-control-expose-headers: X-Total-Count
Cache-Control: no-cache
etag: W/"14-uaD0H7KXSFdrVAldLkbkX0x6Elg"
expires: -1
nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "", "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}], "group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=igwyZYI7RCQVZDeK656NNmqV30jfNFEL2SXYxidf3kk%3D\u0026sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6\u0026ts=1759055538"}], "max_age": 3600}
reporting-endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=igwyZYI7RCQVZDeK656NNmqV30jfNFEL2SXYxidf3kk%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=1759055538"
Server: cloudflare
vary: Origin, Accept-Encoding
via: 2.0 heroku-router
x-content-type-options: nosniff
x-powered-by: Express
x-total-count: 1
cf-cache-status: DYNAMIC
CF-RAY: 9862937d1be33578-WAW
alt-svc: h3=":443"; ma=86400
[{"id": 1, "title": "Hello REST", "userId": 1, "tags": ["intro", "rest"]}]

```

## Сортування + ліміт

```

PS C:\Users\pastu> curl.exe -i "https://my-json-server.typicode.com/mosshkai/LW3_Mia_Pastukh/posts?q=json"
HTTP/1.1 200 OK
Date: Sun, 28 Sep 2025 10:32:26 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 100
Connection: keep-alive
access-control-allow-credentials: true
Cache-Control: no-cache
etag: W/"64-TSbIiSLy0SYEkK1XKAJ+e2yFg"
expires: -1
nel: {"report_to": "heroku-nel", "response_headers": [{"Via": "", "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}], "group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=kX8gPGFvQdduzNKJ38h80uUvNimCZ742Gowfs4du72c%3D\u0026sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6\u0026ts=175905546"}], "max_age": 3600}
reporting-endpoints: heroku-nel="https://nel.herokuapp.com/reports?s=kX8gPGFvQdduzNKJ38h80uUvNimCZ742Gowfs4du72c%3D&sid=929419e7-33ea-4e2f-85f0-7d8b7cd5cbd6&ts=175905546"
Server: cloudflare
vary: Origin, Accept-Encoding
via: 2.0 heroku-router
x-content-type-options: nosniff
x-powered-by: Express
cf-cache-status: DYNAMIC
CF-RAY: 986293adca034f7-WAW
alt-svc: h3=":443"; ma=86400
[{"id": 2, "title": "JSON Tips", "userId": 2, "tags": ["json"]}]

```

## Пошук

## 2 Завдання

### Заходим на сайт і лутаєм кота



Hey ! Do you like Cataas? Do you want to support the project ? [Buy me a beer](#)

Hey cat lovers, new major version of cataas :

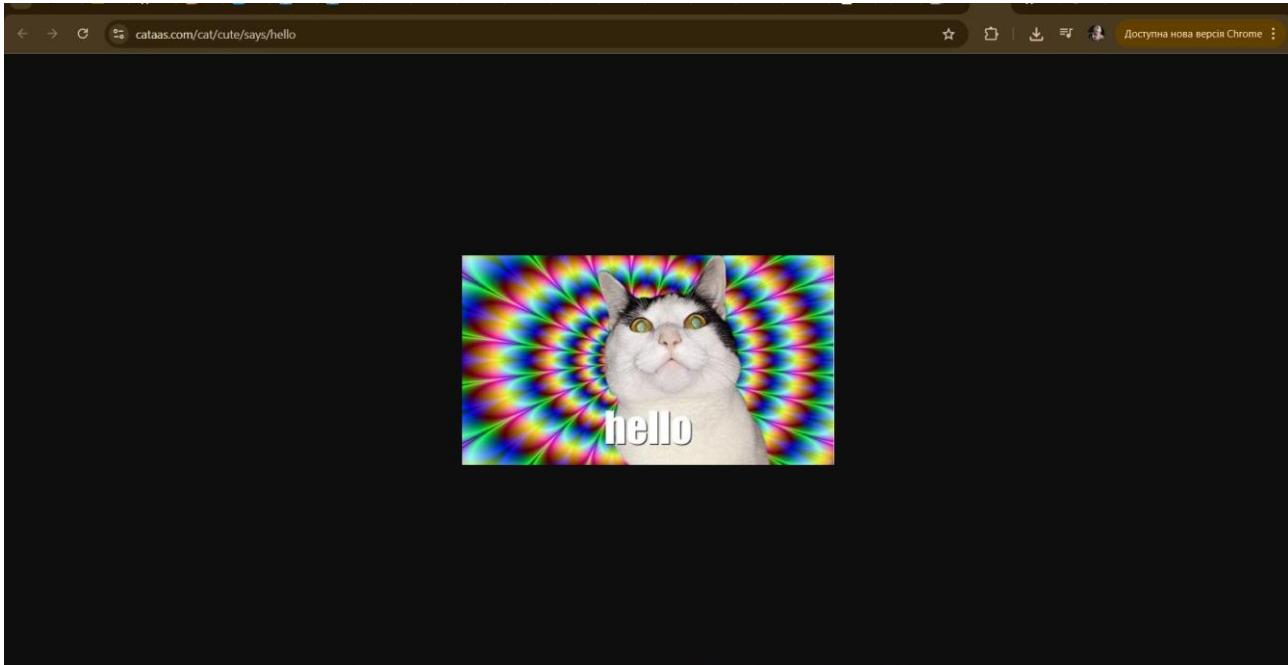
- Now JSON will returns if request contain `application/json` header, API doc updated [here](#)
- Fix tags search and fix tags combined with `,` separator (see documentation)

#### Basic

| Url                 | Description  | Example                                      |
|---------------------|--|--|
| /cat                | Will return a random cat   | <a href="#">Random cat</a>                   |
| /cat:tag            | Will return a random cat with a <code>:tag</code> . You can combine multiple tags with <code>:tag</code> separator | <a href="#">Random orange cute cat</a>       |
| /cat/gif            | Will return a random gif cat \o/   | <a href="#">Random gif cat</a>               |
| /cat/says/:text     | Will return a random cat saying <code>:text</code>   | <a href="#">Random cat saying hello</a>      |
| /cat:tag/says/:text | Will return a random cat with a <code>:tag</code> and saying <code>:text</code>                                    | <a href="#">Random cute cat saying hello</a> |

|      |      |          |        |      |      |
|------|------|----------|--------|------|------|
| Змн. | Арк. | № докум. | Підпис | Дата | Арк. |
|      |      |          |        |      | 7    |

ЛР.ОК24.ПІ231.24.03



Далі переходим в Postman та робим запити там

A screenshot of the Postman application. The left sidebar shows 'Maria Pastuh's Workspace' with collections like 'LW3\_CATAAS\_API' and 'LW3\_Fake\_REST'. The main panel shows a 'GET /cat' request to 'https://cataas.com/cat'. The response status is '200 OK' with a response time of '275 ms' and a size of '40.45 KB'. The response body is a photograph of a brown and white cat standing on a mat with the text 'OH NO NOT YOU AGAIN' printed on it.

Отримати кома

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 8    |

Maria Pastuh's Workspace

LW3\_CATAAS\_API / Get cat with text

GET https://cataas.com/cat/says/i don't like man

200 OK

**i don't like man**

Запит кота з моїм настроєм

```

[{"id": "04eEqh0fAL8l5nt3", "tags": ["two", "double", "black"], "mimetype": "image/jpeg", "createdat": "2022-07-18T11:28:29.596Z"}, {"id": "05Xd4JtN14983pns", "tags": ["Cute"], "mimetype": "image/jpeg", "createdat": "2024-05-27T17:55:08.552Z"}, {"id": "09wExpacQzvI9jEM", ...}
]

```

Отримали айдішиніки коміков

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 9    |

Postman screenshot showing a successful API call to `https://cataas.com/cat/gif`. The response body displays a color GIF of a cat walking on a wooden floor.

*Отримали гіфку котіка*

Postman screenshot showing a successful API call to `https://cataas.com/cat/says/API?filter=mono&width=400&height=200`. The response body displays a black and white filtered image of a cat.

*Котик з параметрами фільтру*

|      |      |          |        |      |                     |      |
|------|------|----------|--------|------|---------------------|------|
| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|      |      |          |        |      |                     | 10   |

The screenshot shows the Postman interface with the following details:

- Collection:** LW3\_CATAAS\_API
- Request:** GET https://cataas.com/api/cats?limit=5
- Query Params:**
  - Key: limit Value: 5
- Response Status:** 200 OK
- Response Body (JSON):**

```

1 [
2   {
3     "id": "8aeE0hDfAL8l5nt3",
4     "tags": [
5       "two",
6       "double",
7       "black"
8     ],
9     "mimetype": "image/jpeg",
10    "createdAt": "2022-07-18T11:28:29.596Z"
11  },
12  {
13    "id": "85Xd4JtN14983pns",
14    "tags": [
15      "Cute"
16    ],
17    "mimetype": "image/jpeg",
18    "createdAt": "2024-05-27T17:55:00.552Z"
19  }
]

```

### Аудіоінікі коміков

The screenshot shows the Postman interface with the following details:

- Collection:** LW3\_CATAAS\_API
- Request:** GET https://cataas.com/cat/gif?json=true
- Query Params:**
  - Key: json Value: true
- Response Status:** 200 OK
- Response Body (JSON):**

```

1 {
2   "id": "RxxKAUiXRG91ft68",
3   "tags": [
4     "cute",
5     "gif"
6   ],
7   "created_at": "2016-09-22T19:50:40.795Z",
8   "url": "https://cataas.com/cat/RxxKAUiXRG91ft68?position=center",
9   "mimetype": "image/gif"
10 }

```

### Гіфка з данними джесон

Далі виконуєм запити через сайт Swagger ui

|      |      |          |        |      | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |                     | 11   |
|      |      |          |        |      |                     |      |

The screenshot shows the SMARTBEAR API Hub Design interface. On the left, the sidebar includes sections for Info, Tags, Servers, default (with three GET methods listed), and Schemas. The main area displays an OpenAPI 3.0.0 specification. The specification defines two paths: '/cat' and '/cat/says/{text}'. The '/cat' path has a 'get' method with a summary of 'Get a random cat image', a response of '200' with a description of 'JPEG image', and a schema of 'image/jpeg'. The '/cat/says/{text}' path also has a 'get' method with a summary of 'Get a cat image with overlaid text', parameters ('text' with type 'string'), and a response of '200' with a description of 'Hello'. The right side of the interface shows a preview of the 'default' endpoint for the '/cat' method, displaying the 'Execute' button and a 'Cancel' button.

```

openapi: 3.0.0
info:
  title: CATAAS demo via Swagger
  version: 1.0.0
  servers:
    - url: https://cataas.com
  paths:
    /cat:
      get:
        summary: Get a random cat image
        responses:
          '200':
            description: JPEG image
            content:
              image/jpeg:
                schema:
                  type: string
                  format: binary
    /cat/says/{text}:
      get:
        summary: Get a cat image with overlaid text
        parameters:
          - name: text
            in: path
            required: true
            schema:
              type: string
              example: Hello
        responses:
          '200':
            description: Get a random cat image
            content:
              application/json:
                schema:
                  type: array
                  items:
                    type: object
                    properties:
                      id:
                        type: string
                      created_at:
                        type: string
                      tags:
                        type: array
                        items:
                          type: string

```

## Ось що ми йому написали

```

openapi: 3.0.0                                     format: binary
info:                                            /api/cats:
  title: CATAAS demo via Swagger               get:
  version: 1.0.0                                 summary: List cat images metadata
  servers:                                         responses:
    - url: https://cataas.com                     '200':
  paths:                                           description: JSON list of cats
    /cat:                                         content:
      get:                                         application/json:
        summary: Get a random cat image           schema:
        responses:                                type: array
          '200':                                 items:
            description: JPEG image             type: object
            content:                           properties:
              image/jpeg:                       id:
                schema:                         type: string
                  type: string
                  format: binary
              schema:                         created_at:
                type: string
              type: string
              example: Hello
              tags:
                type: array
                items:
                  type: string
            responses:                         type: string
              '200':
                description: JPEG image
                content:
                  image/jpeg:
                    schema:
                      type: string

```

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 12   |

Далі виконуєм запити

Snippets ▾

cURL (bash) cURL (PowerShell) cURL (CMD)

```
curl -X 'GET' \
'https://cataas.com/cat' \
-H 'accept: image/jpeg'
```

Request URL

<https://cataas.com/cat>

Server response

**Code** **Details**

---

200 Response body



|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Змн. | Арк. | № докум. | Підпис | Дата |

```
Type,Accept,Origin,Authorization  
access-control-allow-methods: GET,POST,PUT,DELETE,OPTIONS  
access-control-allow-origin: *  
cache-control: no-cache  
content-length: 85548  
content-type: image/jpeg  
date: Sun, 28 Sep 2025 11:31:30 GMT  
etag: W/"14e2c-ogx4k9SBTyQgLOzM9vBFX03Gges"  
expires: -1  
server: nginx  
x-powered-by: Express
```

Request duration

3082 ms

Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | JPEG image  | No links |

Media type

image/jpeg

▼

Controls `Accept` header.

Example Value | Schema

Отримати кома

|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

14

## Parameters

Cancel

| Name | Description |
|------|-------------|
|------|-------------|

|                        |  |
|------------------------|--|
| <b>text</b> * required |  |
|------------------------|--|

|               |  |
|---------------|--|
| <b>string</b> |  |
|---------------|--|

|               |  |
|---------------|--|
| <i>(path)</i> |  |
|---------------|--|

|       |
|-------|
| Hello |
|-------|

Execute

Clear

## Responses

### Snippets ▾

cURL (bash)

cURL (PowerShell)

cURL (CMD)

```
curl -X 'GET' \
'https://cataas.com/cat/says/Hello' \
-H 'accept: image/jpeg'
```



|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

15

Snippets

cURL (bash) cURL (PowerShell) cURL (CMD)

```
curl -X 'GET' \
'https://cataas.com/cat/says/Hello' \
-H 'accept: image/jpeg'
```

Request URL

<https://cataas.com/cat/says/Hello>

Server response

| Code | Details       |
|------|---------------|
| 200  | Response body |



|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

16

```
access-control-allow-methods: GET,POST,PUT,DELETE,OPTIONS
access-control-allow-origin: *
cache-control: no-cache
content-length: 32731
content-type: image/jpeg
date: Sun, 28 Sep 2025 11:31:46 GMT
etag: W/"7fdb-aWdcXdWG+UF6IQtdcpkVanNdkk"
expires: -1
server: nginx
x-powered-by: Express
```

Request duration

2051 ms

#### Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | JPEG image  | No links |

Media type

image/jpeg

Controls [Accept](#) [header](#).

[Example Value](#) | [Schema](#)

string

Komik з надписом

|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

17

Parameters

No parameters

Cancel

Execute Clear

Responses

Snippets ▾

cURL (bash) cURL (PowerShell) cURL (CMD)

```
curl -X 'GET' \
  'https://cataas.com/api/cats' \
  -H 'accept: application/json'
```

Request URL

```
https://cataas.com/api/cats
```

Server response

Code Details

This screenshot shows a user interface for testing a REST API. At the top, there's a 'Parameters' section with a 'Cancel' button. Below it are 'Execute' and 'Clear' buttons. The main area is titled 'Responses' and contains a 'Snippets' section with tabs for 'cURL (bash)', 'cURL (PowerShell)', and 'cURL (CMD)'. A code snippet for a curl command is displayed, which sends a GET request to https://cataas.com/api/cats with an accept header of application/json. Below this is a 'Request URL' field containing the same URL. Further down is a 'Server response' field, which is currently empty. At the bottom, there are 'Code' and 'Details' buttons.

|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|

ЛР.ОК24.ПІ231.24.03

Арк.

18

Request URL

<https://cataas.com/api/cats>

Server response

Code Details

200

Response body

```
[  
  {  
    "id": "04eEQhDfAL815nt3",  
    "tags": [  
      "two",  
      "double",  
      "black"  
    ],  
    "mimetype": "image/jpeg",  
    "createdAt": "2022-07-18T11:28:29.596Z"  
  },  
  {  
    "id": "05Xd4JtN14983pns",  
    "tags": [  
      "Cute"  
    ],  
    "mimetype": "image/jpeg",  
    "createdAt": "2024-05-27T17:55:08.552Z"  
  },  
  {  
    "id": "09wFxpacQzvf9jfM",  
    "tags": [  
      "black",  
      "white",  
      "fluffy",  
      "adorable"  
    ],  
    "mimetype": "image/jpeg",  
    "createdAt": "2024-05-27T17:55:08.552Z"  
  }]
```

|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

19

```

{
  "id": "0F0IKAPoWiE755P",
  "tags": [
    "meet",
    "cute"
  ],
  "mimetype": "image/jpeg",
  "createdAt": "2024-06-18T09:46:45.702Z"
},

```

Download

Response headers

```

access-control-allow-headers: X-Requested-With,Content-Type,Accept-Origin,Authorization
access-control-allow-methods: GET,POST,PUT,DELETE,OPTIONS
access-control-allow-origin: *
cache-control: no-cache
content-encoding: gzip
content-size: 1101
content-type: application/json; charset=utf-8
date: Sun, 28 Sep 2025 11:31:57 GMT
etag: W/"44d-3y5xZa1TMByl1M9tEc715PUBSP0"
expires: -1
server: nginx
x-powered-by: Express

```

Request duration

861 ms

Responses

*Отримуєм аддішиніки коміків*

### 3 Завдання

Сайт: <https://open.er-api.com>

#### Крок 1 - Створюємо DTO

```

public class Latest
{
    public string result { get; set; } = string.Empty;
    public string base_code { get; set; } = string.Empty;
    public string time_last_update_utc { get; set; } = string.Empty;
    public Dictionary<string, decimal> rates { get; set; } = new();
}

```

#### Крок 2 — Налаштuvати HttpClient

```

private static readonly HttpClient _http = new HttpClient
{
    BaseAddress = new Uri("https://open.er-api.com/v6/"), // ← слеш в кінці
    Timeout = TimeSpan.FromSeconds(15)
}

```

| Змн. | Арк. | № докум. | Підпис | Дата | Арк. |
|------|------|----------|--------|------|------|
|      |      |          |        |      | 20   |

```
};
```

### Крок 3 — Виконати GET-запит

```
var response = await _http.GetAsync("latest/USD");
response.EnsureSuccessStatusCode();
```

### Крок 4 — Отримати JSON як рядок

```
var json = await response.Content.ReadAsStringAsync();
Console.WriteLine("Raw JSON:");
Console.WriteLine(json);
```

### Крок 5 — Розпарсити JSON у об'єкт Latest

```
var latest = JsonSerializer.Deserialize<Latest>(json, new JsonSerializerOptions
{
    PropertyNameCaseInsensitive = true // імена полів не чутливі до регістру
});
```

### Крок 6 — Використати дані з об'єкта

```
if (latest != null)
{
    Console.WriteLine("\n==== Результат парсингу ====");
    Console.WriteLine($"Result: {latest.result}");
    Console.WriteLine($"Base: {latest.base_code}");
    Console.WriteLine($"Updated: {latest.time_last_update_utc}");

    if (latest.rates.TryGetValue("UAH", out var uah))
        Console.WriteLine($"USD → UAH: {uah}");
    if (latest.rates.TryGetValue("EUR", out var eur))
        Console.WriteLine($"USD → EUR: {eur}");

    Console.WriteLine($"Всього валют у відповіді: {latest.rates.Count}");
}
```

### Код:

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Text.Json;
using System.Threading.Tasks;

public class Latest
{
    public string result { get; set; } = string.Empty;
    public string base_code { get; set; } = string.Empty;
    public string time_last_update_utc { get; set; } =
string.Empty;
    public Dictionary<string, decimal> rates { get; set; }
= new();
}

public class Pair
{
```

```
    public string result { get; set; } = string.Empty;
    public string base_code { get; set; } = string.Empty;
    public string target_code { get; set; } = string.Empty;
    public decimal conversion_rate { get; set; }
}

class Program
{
    private static readonly HttpClient _http = new()
    {
        BaseAddress = new Uri("https://open.er-
api.com/v6/"),
        Timeout = TimeSpan.FromSeconds(15)
    };

    static async Task Main()
    {
        try
```

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

```

    {
        Console.WriteLine("Запит до ExchangeRate
API...");
    }

    var resp = await _http.GetAsync("latest/USD");
    Console.WriteLine($"Request URL:
{resp.RequestMessage?.RequestUri}");
    Console.WriteLine($"Status:
{(int)resp.StatusCode} {resp.StatusCode}");
    resp.EnsureSuccessStatusCode();

    var json = await
resp.Content.ReadAsStringAsync();

    var pretty = JsonSerializer.Serialize(
        JsonSerializer.Deserialize<object>(json)!, new JsonSerializerOptions { WriteIndented = true });
    Console.WriteLine("Raw JSON (pretty):");
    Console.WriteLine(pretty);

    var latest =
JsonSerializer.Deserialize<Latest>(json, new JsonSerializerOptions
{
    PropertyNameCaseInsensitive = true
});

    if (latest is not null)
    {
        Console.WriteLine("\n==== Результат
парсингу ====");
        Console.WriteLine($"Result:
{latest.result}");
        Console.WriteLine($"Base:
{latest.base_code}");
    }
}

```

```

    Console.WriteLine($"Updated:
{latest.time_last_update_utc}");

    foreach (var code in new[] { "UAH", "EUR",
"PLN" })
    {
        if (latest.rates.TryGetValue(code, out var
rate))
            Console.WriteLine($"USD → {code}:
{rate}");
    }
    Console.WriteLine($"Всього валют:
{latest.rates.Count}");
}

var pairResp = await
_http.GetAsync("pair/USD/UAH");
pairResp.EnsureSuccessStatusCode();
var pairJson = await
pairResp.Content.ReadAsStringAsync();
var pair =
JsonSerializer.Deserialize<Pair>(pairJson, new
JsonSerializerOptions { PropertyNameCaseInsensitive =
true });
if (pair?.result == "success")
    Console.WriteLine($"\\nUSD → UAH (pair):
{pair.conversion_rate}");
}
catch (Exception ex)
{
    Console.WriteLine("Помилка: " +
ex.Message);
}
}
}

```

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Змн | Арк. | № докум. | Підпис | Дата |

## Результати

```
Запит до ExchangeRate API...
Request URL: https://open.er-api.com/v6/latest/USD
Status: 200 OK
Raw JSON (pretty):
{
  "result": "success",
  "provider": "https://www.exchangerate-api.com",
  "documentation": "https://www.exchangerate-api.com/docs/free",
  "terms_of_use": "https://www.exchangerate-api.com/terms",
  "time_last_update_unix": 1759017751,
  "time_last_update_utc": "Sun, 28 Sep 2025 00:02:31 \u0002B0000",
  "time_next_update_unix": 1759105261,
  "time_next_update_utc": "Mon, 29 Sep 2025 00:21:01 \u0002B0000",
  "time_eol_unix": 0,
  "base_code": "USD",
  "rates": {
    "USD": 1,
    "AED": 3.6725,
    "AFN": 67.864751,
    "ALL": 82.371472,
    "AMD": 382.956195,
    "ANG": 1.79,
    "AOA": 919.644732,
    "ARS": 1332,
    "AUD": 1.52877,
    "AWG": 1.79,
    "AZN": 1.760278,
    "BAM": 1.672817,
    "BBD": 2,
    "BDT": 121.768741,
    "BGN": 1.672908,
    "BHD": 0.376,
    "BIF": 2968.196383,
    "BMD": 1,
    "BND": 1.292363,
    "BOB": 6.922571,
    "BRL": 5.358323,
    "BSD": 1,
    "BTN": 88.738384,
    "BWP": 13.857097,
    "BYN": 3.161041,
    "CHF": 0.798727,
    "CLP": 957.27094,
    "CNY": 7.138756,
    "COP": 3889.069628,
    "CRC": 504.548728,
    "CUP": 24,
    "CVE": 94.309426,
    "CZK": 28.794685,
    "DJF": 177.721,
    "DKK": 6.381172,
    "DOP": 62.033057,
    "DZD": 129.438976,
    "EGP": 48.130798,
    "ERN": 15,
    "ETB": 143.392415,
    "EUR": 0.8553,
    "FJD": 2.248785,
    "FKP": 0.747102,
    "FOK": 6.381172,
    "GBP": 0.747105,
    "GEL": 2.707331,
    "GGP": 0.747102,
    "GHS": 12.679412,
    "GIP": 0.747102,
    "GMD": 73.396812,
    "GNF": 8684.808238,
    "GTQ": 7.661452,
    "GYD": 289.061914,
    "HKD": 7.779962,
    "HNL": 26.242996,
    "HRK": 6.444242,
    "HTG": 131.014828,
    "HUF": 334.780351,
    "IDR": 16734.78514,
    "ILS": 3.356577,
    "IMP": 0.747102,
    "INR": 88.738419,
    "IQD": 1308.617896,
    "IRR": 42230.978321,
    "ISK": 121.647063,
    "JEP": 0.747102,
```

|      |      |          |        |      |                     |      |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | Арк. |
|      |      |          |        |      |                     |      |
| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | 23   |

```

"KES": 129.188749,
"KGS": 87.346796,
"KHR": 4008.793783,
"KID": 1.528768,
"KMF": 428.778997,
"KRW": 1489.877219,
"KWD": 0.385688,
"KYD": 0.833333,
"KZT": 543.116911,
"LAK": 21716.364773,
"LBW": 89500,
"LRK": 302.441342,
"LRD": 180.302011,
"LSL": 17.383856,
"LYD": 5.39509,
"MAD": 9.074626,
"MDL": 16.694522,
"MGA": 4448.59357,
"MKD": 52.792171,
"MMK": 2100.885926,
"MTN": 3669.781985,
"MOB": 8.013361,
"MRU": 39.99882,
"MRV": 45.647912,
"MWK": 15.433141,
"MXN": 1743.689635,
"MXN": 18.495764,
"MYR": 4.219515,
"MZM": 63.562483,
"NAD": 17.383856,
"NGN": 1484.894837,
"NGN": 36.778979,
"NOK": 9.987465,
"NPR": 141.981414,
"NZD": 1.73328,
"QAR": 0.384497,
"PAB": 1,
"PEN": 3.498359,
"PGK": 4.214973,
"PHP": 58.165293,
"PKR": 283.583522,
"PLN": 3.658154,
"PYG": 7102.736763,
"QAR": 3.64,
"RON": 4.348374,
"RSD": 100.073583,
"RUB": 83.589463,
"RWF": 1451.866453,
"SAR": 3.75,
"SBD": 8.235087,
"SCR": 14.94767,
"SDG": 588.373839,
"SEK": 9.42133,
"SGD": 1.292364,
"SHP": 0.747192,
"SLE": 23.268029,
"SLL": 23268.928586,
"SOS": 571.569809,
"SRD": 38.153805,
"SSP": 4692.063305,
"STN": 20.984799,
"SVP": 13111.313567,
"SZL": 17.383856,
"THB": 32.230914,
"TJS": 9.410113,
"TTT": 3.499318,
"TND": 2.913729,
"TOP": 2.362466,
"TRY": 41.587014,
"TTD": 6.759222,
"TVB": 1.528768,
"TWD": 30.465228,
"TZS": 2440.057849,
"UAH": 41.491911,
"UGX": 3553.395116,
"UYU": 39.891196,
"UZS": 12157.000825,
"VES": 175.6471,
"VND": 26207.764434,
"VUV": 119.979394,
"WST": 2.698894,
}

"TZS": 2440.057849,
"UAH": 41.491911,
"UGX": 3553.395116,
"UYU": 39.891196,
"UZS": 12157.000825,
"VES": 175.6471,
"VND": 26207.764434,
"VUV": 119.979394,
"WST": 2.698894,
"XAF": 561.038662,
"XCD": 2.7,
"XCG": 1.79,
"XDR": 0.732052,
"XOF": 561.038662,
"XPF": 102.064412,
"YER": 239.310519,
"ZAR": 17.383981,
"ZMW": 23.798514,
"ZWL": 26.5708
}
}

== Результат парсингу ===
Result: success
Base: USD
Updated: Sun, 28 Sep 2025 00:02:31 +0000
USD ₴ UAH: 41,491911
USD ₪ EUR: 0,8553
USD ₩ PLN: 3,658154
Всього валют: 163
Помилка: Response status code does not indicate success: 404 (Not Found).

C:\Users\pastu\source\repos\LW3_Mod\LW3_Mod\bin\Debug\net8.0\LW3_Mod.exe (process 3372) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

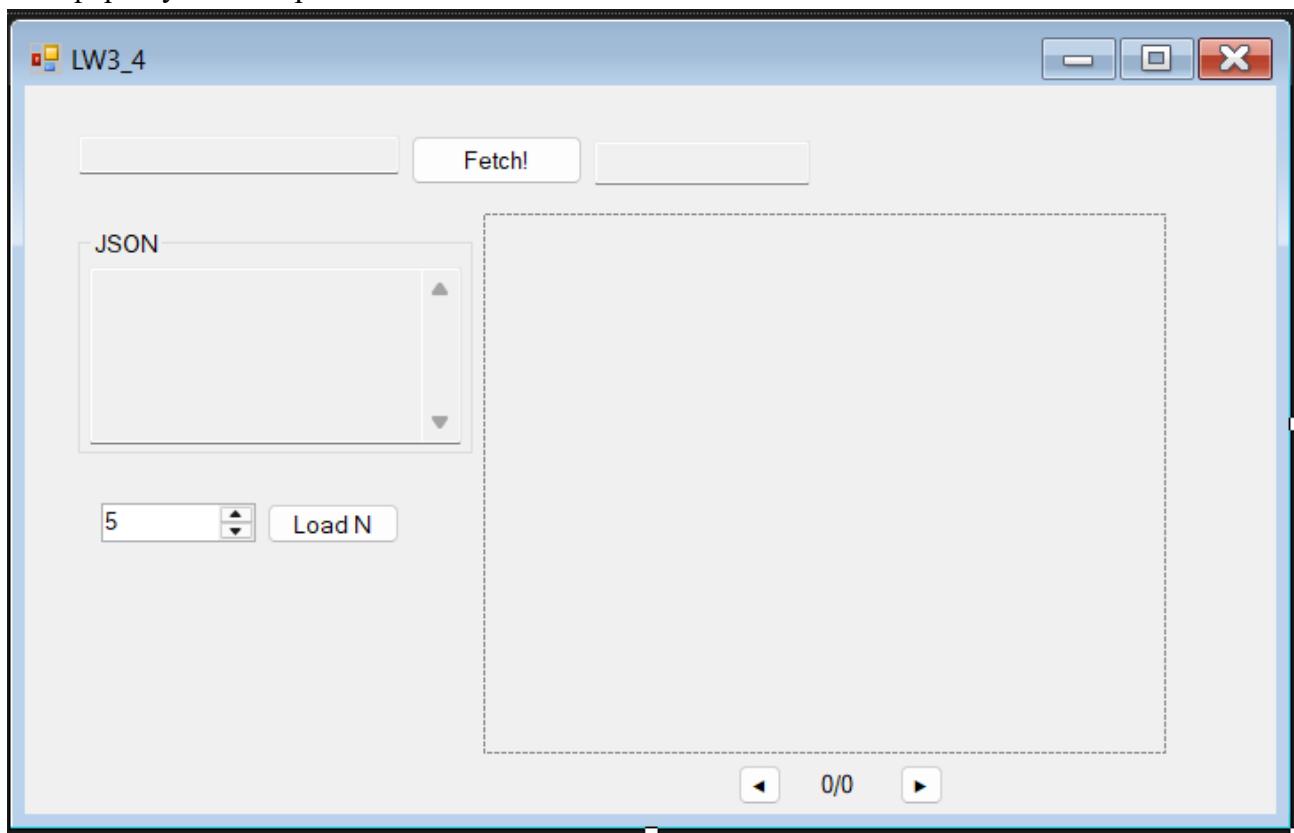
```

| Змн. | Арк. | № докум. | Підпис | Дата | Арк. |
|------|------|----------|--------|------|------|
|      |      |          |        |      | 24   |

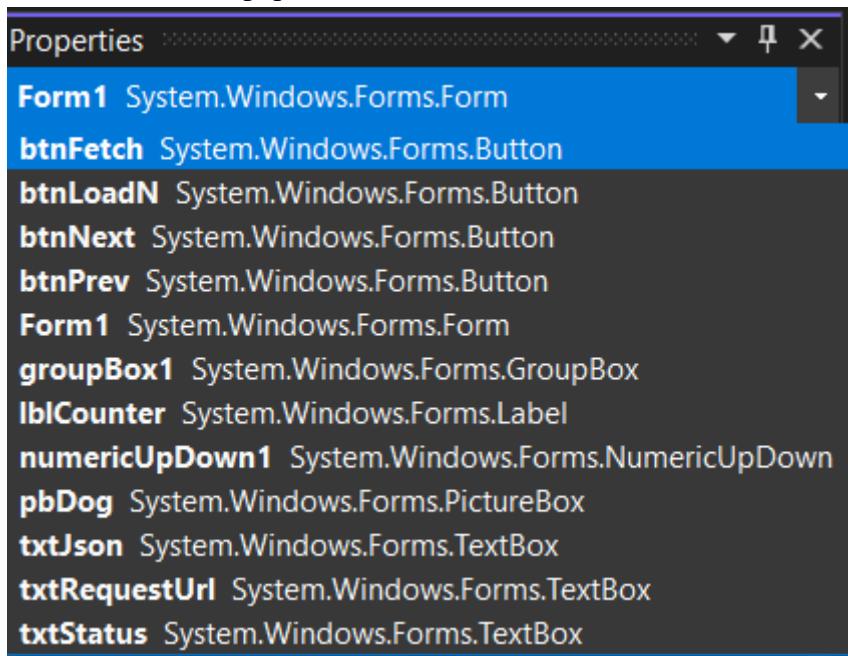
ЛР.ОК24.ПІ231.24.03

## 4 Завдання

Ось форма у дизайнері



Наші об'єкти на формі



| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

## Class1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LW3_4
{
    public class DogSingleResponse
    {
        public string message { get; set; } = string.Empty;
        public string status { get; set; } = string.Empty;
    }

    public class DogMultiResponse
    {
        public List<string> message { get; set; } = new List<string>();
        public string status { get; set; } = string.Empty;
    }
}
```

## Form1.cs

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Text.Json;
using System.Windows.Forms;

namespace LW3_4
{
    public partial class Form1 : Form
    {
        private static readonly HttpClient http = new HttpClient
        {
            BaseAddress = new Uri("https://dog.ceo/"),
            Timeout = TimeSpan.FromSeconds(15)
        };

        private readonly JsonSerializerOptions jsonOpts =
new JsonSerializerOptions
{
    PropertyNameCaseInsensitive = true,
    WriteIndented = true
};

        private List<string> _urls = new List<string>();
        private int _i = -1;

        public Form1()
        {
            InitializeComponent();
            pbDog.SizeMode =
PictureSizeMode.Zoom;
        }

        private void btnFetch_Click(object sender, EventArgs e)
        {
            btnFetch.Enabled = false;
            try
            {
                var resp = await
http.GetAsync("api/breeds/image/random");
                SetMeta(resp);
                resp.EnsureSuccessStatusCode();

                string json = await
resp.Content.ReadAsStringAsync();
                SetPrettyJson(json);
                string url = "";
                using (var doc = JsonDocument.Parse(json))
                {
                    if
(doc.RootElement.TryGetProperty("message", out var
msgProp) && msgProp.ValueKind ==
JsonValueKind.String)
                        url = msgProp.GetString();
                }

                if (!string.IsNullOrEmpty(url))
                {
                    _urls = new List<string> { url };
                    _i = 0;
                    pbDog.LoadAsync(url);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void SetMeta(HttpResponseMessage response)
        {
            UpdateNav();
        }

        private void SetPrettyJson(string json)
        {
            richTextBox1.Text = json;
        }
    }
}
```

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

```

        }
    else
    {
        _urls.Clear(); _i = -1; pbDog.Image = null;
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message,
    "Помилка");
}
finally
{
    UpdateNav();
    btnFetch.Enabled = true;
}
}

private async void btnLoadN_Click(object sender,
EventArgs e)
{
    btnLoadN.Enabled = btnPrev.Enabled =
    btnNext.Enabled = false;
    try
    {
        int n = (int)numericUpDown1.Value;
        var resp = await
http.GetAsync("api/breeds/image/random/" + n);
        SetMeta(resp);
        resp.EnsureSuccessStatusCode();

        string json = await
resp.Content.ReadAsStringAsync();
        SetPrettyJson(json);

        var list = new List<string>();
        using (var doc = JsonDocument.Parse(json))
        {
            var root = doc.RootElement;
            if (root.TryGetProperty("message", out var
arr) && arr.ValueKind == JsonValueKind.Array)
            {
                foreach (var it in arr.EnumerateArray())
                {
                    if (it.ValueKind ==
JsonValueKind.String)
                    {
                        var u = it.GetString();
                        if (!string.IsNullOrEmpty(u))
list.Add(u);
                    }
                }
            }
        }
    }

    if (list.Count > 0)
    {
        _urls = list;
        _i = 0;
        pbDog.LoadAsync(_urls[_i]);
    }
    else
    {
        _urls.Clear(); _i = -1; pbDog.Image = null;
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message,
    "Помилка");
}
finally
{
    UpdateNav();
    btnLoadN.Enabled = true;
}
}

private void btnPrev_Click(object sender,
EventArgs e)
{
    if (_i > 0)
    {
        _i--;
        pbDog.LoadAsync(_urls[_i]);
        UpdateNav();
    }
}

private void btnNext_Click(object sender,
EventArgs e)
{
    if (_i >= 0 && _i < _urls.Count - 1)
    {
        _i++;
        pbDog.LoadAsync(_urls[_i]);
        UpdateNav();
    }
}

private void UpdateNav()
{
    btnPrev.Enabled = _i > 0;
    btnNext.Enabled = _i >= 0 && _i < _urls.Count
    - 1;
    lblCounter.Text = _urls.Count == 0 ? "0/0" : (_i
+ 1) + "/" + _urls.Count;
}

private void SetMeta(HttpResponseMessage resp)
{
    txtRequestUrl.Text = resp.RequestMessage !=
null ? resp.RequestMessage.RequestUri.ToString() : "";
}

```

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

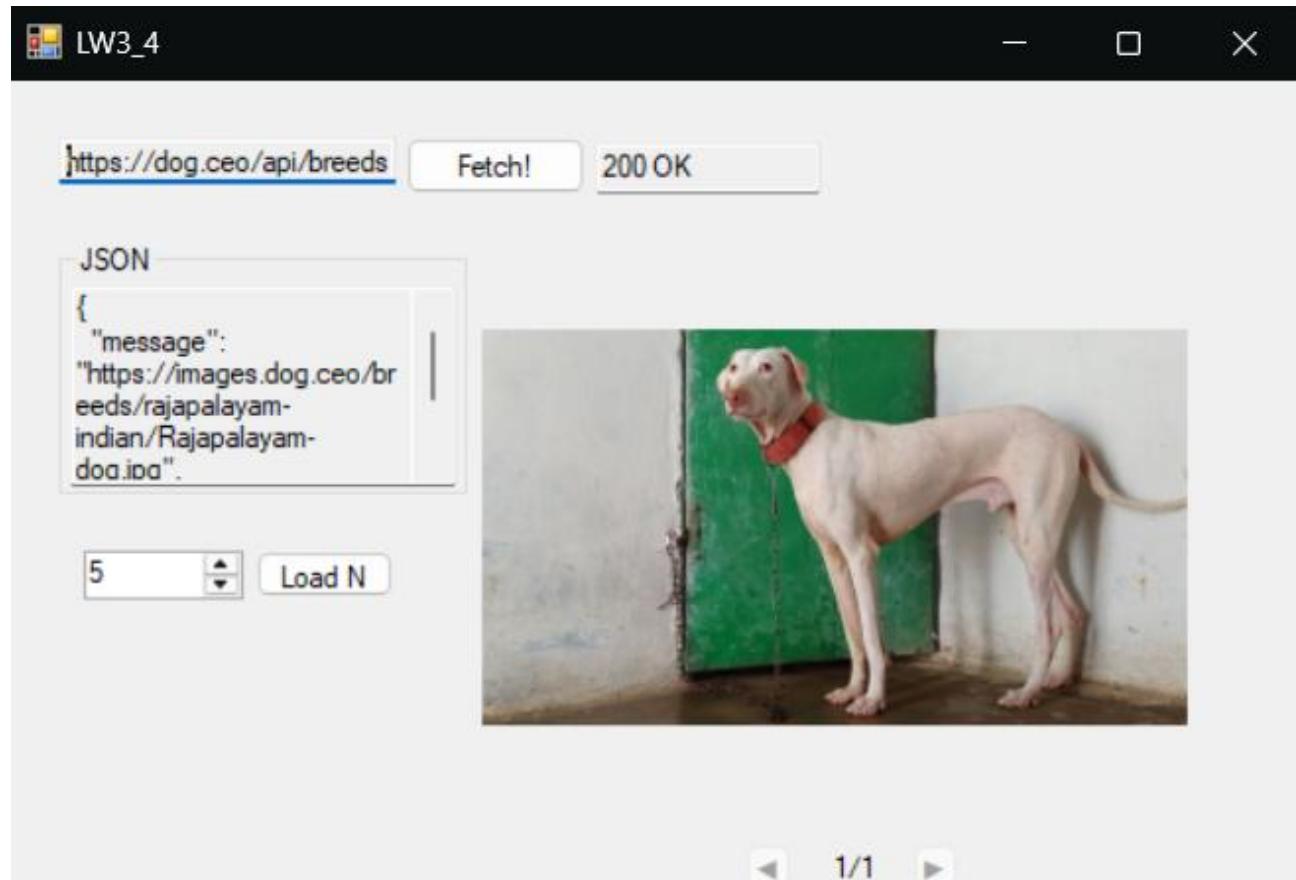
```

txtStatus.Text = ((int)resp.StatusCode) + " " +
resp.ReasonPhrase;
}

private void SetPrettyJson(string json)
{
    try
    {
        using (var doc = JsonDocument.Parse(json))
        {
            txtJson.Text =
JsonSerializer.Serialize(doc.RootElement, jsonOpts);
        }
    }
    catch
    {
        txtJson.Text = json;
    }
}

```

## Реалізація форми



*Натискаємо кнопку **Фетч!** та нам в пікчербокс виводиться песик*

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| ЗМН. | Арк. | № докум. | Підпис | Дата |

JP,OK24,PI231,24,03

---

Арк.

28

LW3\_4

<https://dog.ceo/api/breeds> Fetch! 200 OK

JSON

```
{ "message": "https://images.dog.ceo/breeds/maltese/n02085936_9632.jpg", "status": "success" }
```

5  Load N



1/1

Повторюємо цю дію

LW3\_4

<https://dog.ceo/api/breeds> Fetch! 200 OK

JSON

```
{ "message": [ "https://images.dog.ceo/breeds/corgi-cardigan/n02113186_2635.j" ] }
```

5  Load N



1/5

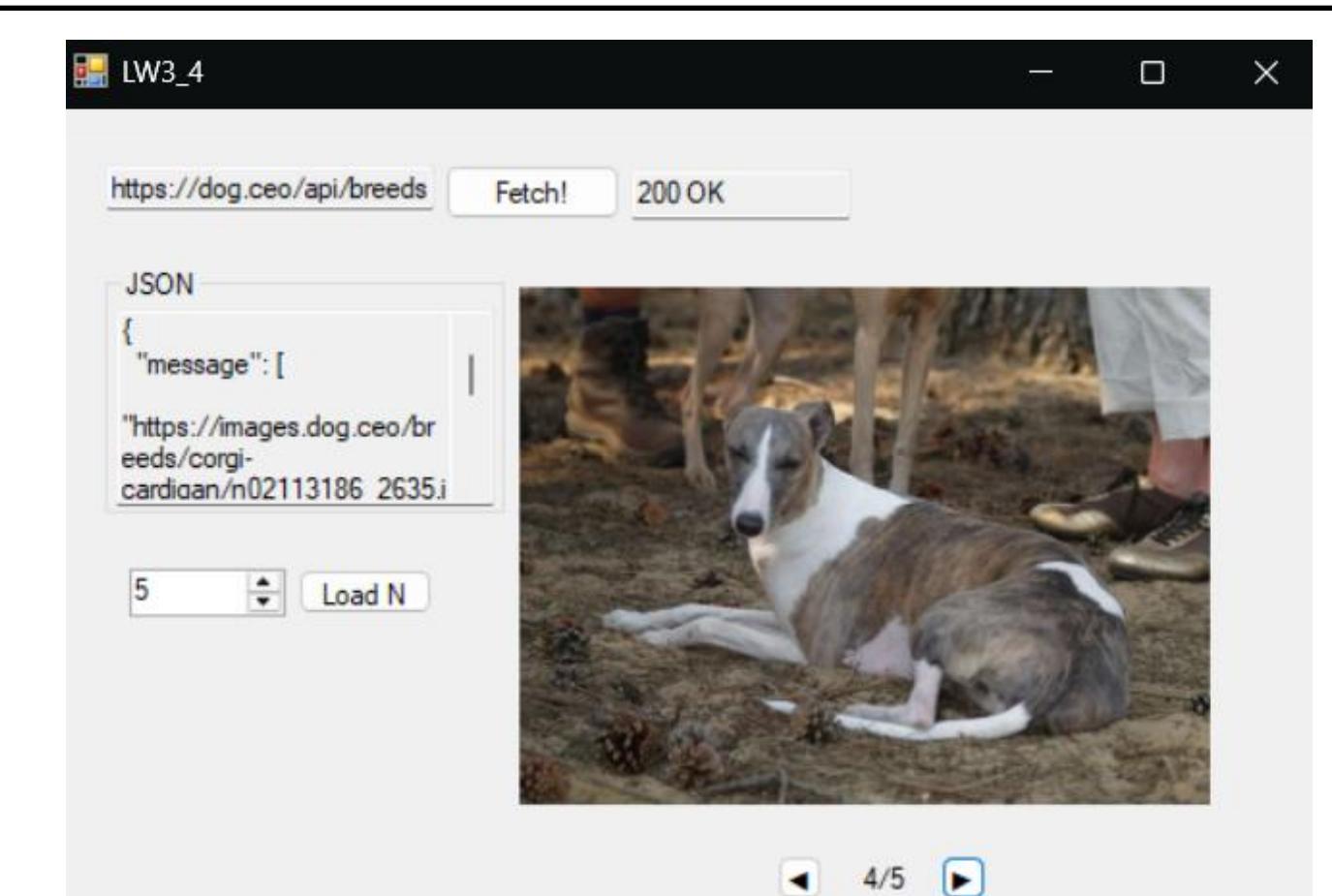
Тут треба звернути увагу на кнопку лоадН. В нас там обрано 5 фото собак і на наступній скріні в нас показано ще 4 песика з «Галереї песиків»

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|

ЛР.ОК24.ПІ231.24.03

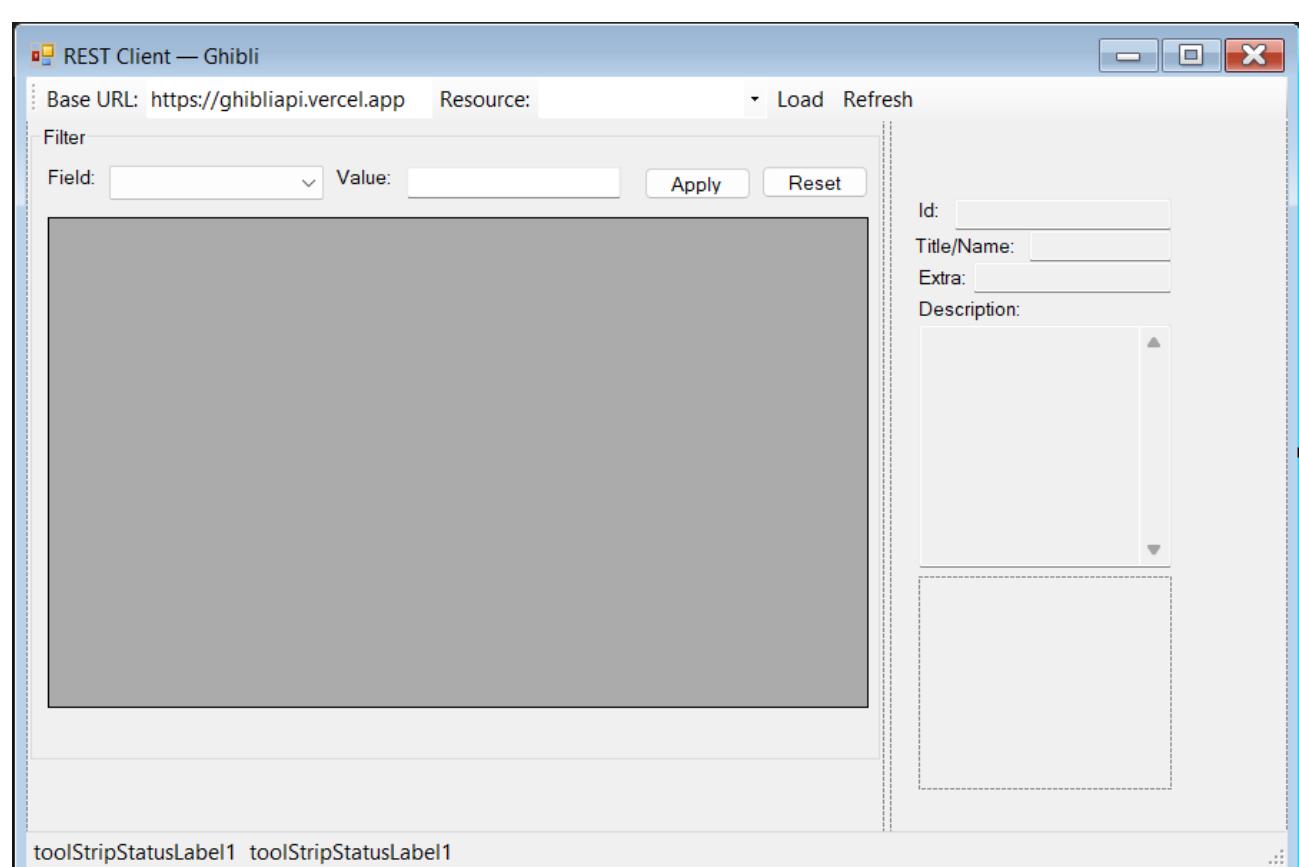
Арк.

29



## 5 Завдання

Форма у дизайнери



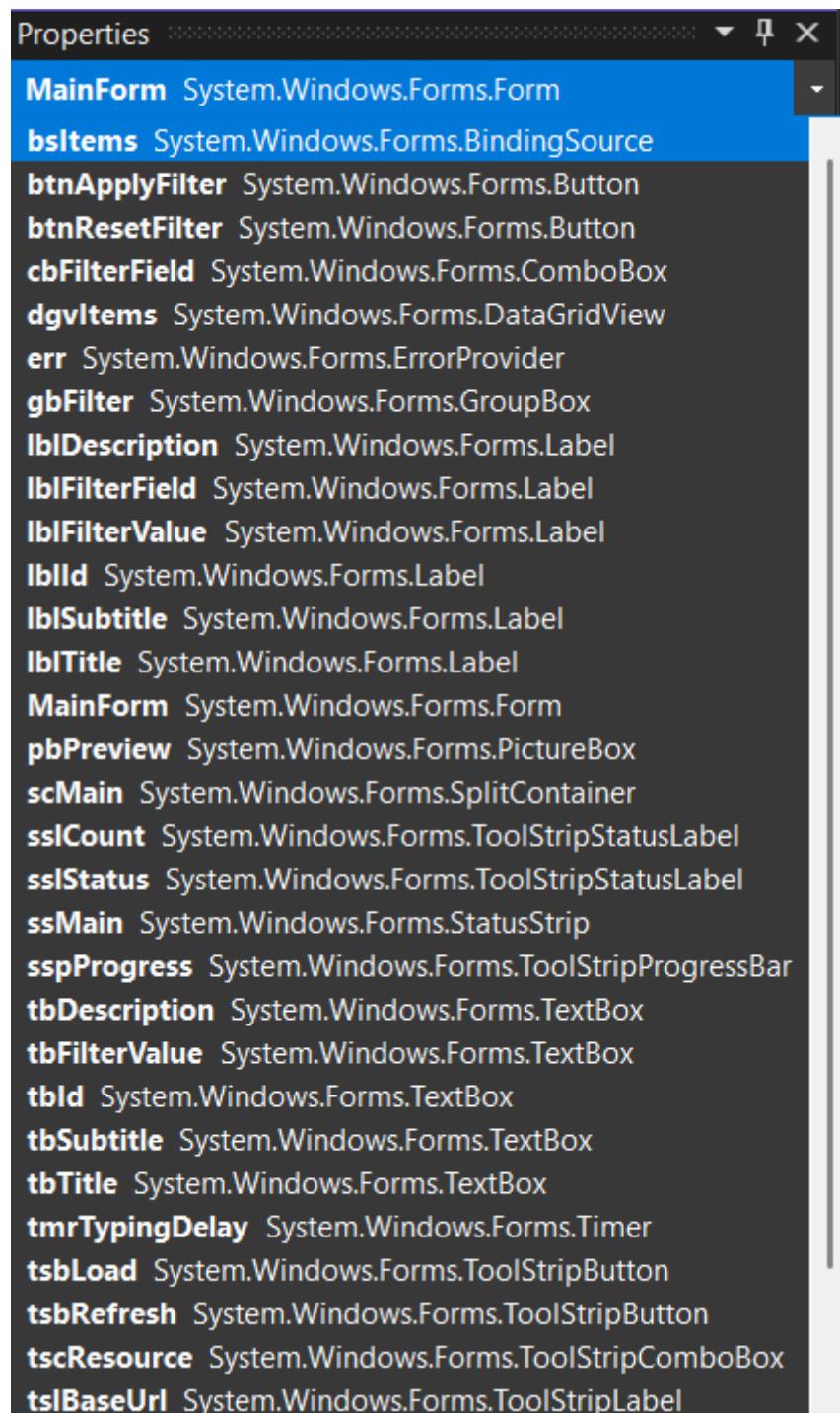
| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

30

## Елементи на формі(не всі, усі не вміщаються на скріншот)



|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Змн. | Арк. | № докум. | Підпис | Дата |

ЛР.ОК24.ПІ231.24.03

Арк.

31

## Class1.cs

```
using Newtonsoft.Json;

namespace LW3_Form5
{
    public class Film
    {
        [JsonProperty("id")] public string Id { get; set; } = "";
        [JsonProperty("title")] public string Title { get; set; } = "";
        [JsonProperty("original_title")] public string OriginalTitle { get; set; } = "";
        [JsonProperty("director")] public string Director { get; set; } = "";
        [JsonProperty("release_date")] public string ReleaseDate { get; set; } = "";
        [JsonProperty("description")] public string Description { get; set; } = "";
        [JsonProperty("url")] public string Url { get; set; } = "";
    }
}
```

## ApiClient.cs

```
using System;
using System.Collections.Generic;
using System.Net.Http;
using System.Threading.Tasks;
using Newtonsoft.Json;

namespace LW3_Form5
{
    public class ApiClient : IDisposable
    {
        private readonly HttpClient _http;

        public ApiClient(string baseUrl)
        {
            _http = new HttpClient { BaseAddress = new Uri(baseUrl) };
            _http.Timeout = TimeSpan.FromSeconds(15);
        }

        public async Task<List<Film>> GetFilmsAsync()
        {
            using (var resp = await
            _http.GetAsync("/films"))
            {
                resp.EnsureSuccessStatusCode();
                var json = await
                resp.Content.ReadAsStringAsync();
                return
                JsonConvert.DeserializeObject<List<Film>>(json);
            }
        }

        public void Dispose()
        {
            _http.Dispose();
        }
    }
}
```

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

## Form1.cs

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LW3_Form5
{
    public partial class MainForm : Form
    {
        private ApiClient _client;
        private List<Film> _allFilms = new List<Film>();

        public MainForm()
        {
            InitializeComponent();

            dgvItems.DataSource = bsItems;
            dgvItems.AutoGenerateColumns = false;

            tstBaseUrl.Text = "https://ghibliapi.vercel.app";
            tscResource.Items.Clear();
            tscResource.Items.Add("/films");
            tscResource.SelectedIndex = 0;
            tscResource.Enabled = false;

            tsbLoad.Click += async (s, e) => await LoadFilmsAsync();
            tsbRefresh.Click += async (s, e) => await LoadFilmsAsync();
            dgvItems.SelectionChanged += async (s, e) => await LoadFilmDetailsAsync();
            btnApplyFilter.Click += (s, e) => ApplyFilter();
            btnResetFilter.Click += (s, e) => ResetFilter();
            tbFilterValue.KeyDown += (s, e) => { if (e.KeyCode == Keys.Enter) { ApplyFilter(); e.SuppressKeyPress = true; } };

            EnsureFilmColumns();
            PopulateFilterFields();
        }

        private void ToggleUi(bool enabled)
        {
            tsbLoad.Enabled = enabled;
            tsbRefresh.Enabled = enabled;
            sspProgress.Visible = !enabled;
        }

        private async Task LoadFilmsAsync()
        {
            var baseUrl = (tstBaseUrl.Text ?? "").Trim();
            if (string.IsNullOrEmpty(baseUrl)) return;

            try
            {
                ToggleUi(false);
                sslStatus.Text = "Loading films...";

                if (_client != null) _client.Dispose();
                _client = new ApiClient(baseUrl);

                var items = await _client.GetFilmsAsync();

                _allFilms = items;
                bsItems.DataSource = _allFilms;
                dgvItems.Refresh();

                sslCount.Text = items.Count + " items";
                sslStatus.Text = "Ready";
            }
            catch (Exception ex)
            {
                sslStatus.Text = "Error: " + ex.Message;
                MessageBox.Show(ex.Message, "Load error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            finally
            {
                ToggleUi(true);
            }
        }

        private async Task LoadFilmDetailsAsync()
        {
            try
            {
                if (_client == null) return;
                if (dgvItems.SelectedRows.Count == 0) return;

                var idCell =
                    dgvItems.SelectedRows[0].Cells["Id"].Value;
                var id = idCell == null ? "" :
                    idCell.ToString();
                if (string.IsNullOrEmpty(id)) return;

                sslStatus.Text = "Loading details...";
                var film = await _client.GetFilmByIdAsync(id);
                if (film != null)
                {
                    tbId.Text = film.Id;
                    tbTitle.Text = film.Title;
                    tbSubtitle.Text = film.Director;
                    tbDescription.Text = film.Description;
                }
                sslStatus.Text = "Ready";
            }
            catch (Exception ex)
            {
                sslStatus.Text = "Error: " + ex.Message;
                MessageBox.Show(ex.Message, "Load error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}
```

| Змн. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

```

        }
        catch (Exception ex)
        {
            sslStatus.Text = "Details error: " +
            ex.Message;
        }
    }

    private void EnsureFilmColumns()
    {
        dgvItems.ScrollBars = ScrollBars.Both;
        dgvItems.AutoGenerateColumns = false;
        dgvItems.DefaultCellStyle.WrapMode =
        DataGridViewTriState.False;
        dgvItems.RowHeadersVisible = false;
        dgvItems.AutoSizeColumnsMode =
        DataGridViewAutoSizeColumnsMode.AllCells;
        if (dgvItems.Columns.Count > 0 &&
        dgvItems.Columns["Id"] != null)
        {
            dgvItems.Columns["Id"].Visible = true;
            dgvItems.Columns["Id"].Width = 260;
            return;
        }

        dgvItems.Columns.Clear();

        var colId = new DataGridViewTextBoxColumn
        {
            DataPropertyName = "Id",
            Name = "Id",
            HeaderText = "Id",
            Visible = true,
            Width = 260
        };
        dgvItems.Columns.Add(colId);

        var colTitle = new
        DataGridViewTextBoxColumn
        {
            DataPropertyName = "Title",
            Name = "Title",
            HeaderText = "Title"
        };
        dgvItems.Columns.Add(colTitle);
        var colDirector = new
        DataGridViewTextBoxColumn
        {
            DataPropertyName = "Director",
            Name = "Director",
            HeaderText = "Director"
        };
        dgvItems.Columns.Add(colDirector);
        var colYear = new
        DataGridViewTextBoxColumn
        {
            DataPropertyName = "ReleaseDate",
            Name = "ReleaseDate",
            HeaderText = "Year"
        };
        dgvItems.Columns.Add(colYear);
    }

    private void PopulateFilterFields()
    {
        cbFilterField.Items.Clear();
        cbFilterField.Items.Add("Title");
        cbFilterField.Items.Add("Director");
        cbFilterField.Items.Add("Year");
        cbFilterField.SelectedIndex = 0;
    }

    private void ApplyFilter()
    {
        if (_allFilms == null || _allFilms.Count == 0)
        return;

        var field = cbFilterField.SelectedItem == null ?
        "Title" : cbFilterField.SelectedItem.ToString();
        var value = (tbFilterValue.Text ??
        "").Trim().ToLower();

        if (string.IsNullOrEmpty(value))
        {
            bsItems.DataSource = _allFilms;
            sslCount.Text = _allFilms.Count + " items";
            sslStatus.Text = "Filter cleared";
            return;
        }

        var filtered = new
        System.Collections.Generic.List<Film>();
        foreach (var f in _allFilms)
        {
            string candidate;
            switch (field)
            {
                case "Title": candidate = f.Title; break;
                case "Director": candidate = f.Director;
                break;
                case "Year": candidate = f.ReleaseDate;
                break;
                default: candidate = f.Title; break;
            }

            if (!string.IsNullOrEmpty(candidate) &&
            candidate.ToLower().IndexOf(value,
            StringComparison.InvariantCultureIgnoreCase) >= 0)
            {
                filtered.Add(f);
            }
        }
    }
}

```

|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

```

        }

        bsItems.DataSource = filtered;
        sslCount.Text = filtered.Count + " items";
        sslStatus.Text = "Filter applied";
    }
}

private void ResetFilter()
{
}

```

## Реалізація форми

The screenshot shows a Windows application window titled "REST Client — Ghibli". At the top, it displays the "Base URL: https://ghibliapi.vercel.app" and "Resource: /films". Below this is a "Filter" section with a dropdown set to "Title" and a text input field containing "Value: Castle in the Sky". There are "Apply" and "Reset" buttons. To the right of the filter is a detailed view of the film "Castle in the Sky". The detailed view includes fields for "Id: 2ba70d1-42bb-4437-b551-e5fed5a87abe", "Title: Castle in the Sky", "Direct: Hayao", and a large text block describing the movie's plot. At the bottom left, there is a message "Ready 22 items".

| Id                                   | Title                    | Direct |
|--------------------------------------|--------------------------|--------|
| 2ba70d1-42bb-4437-b551-e5fed5a87abe  | Castle in the Sky        | Hayao  |
| 12cfb892-aac0-4c5b-94af-521852e46d6a | Grave of the Fireflies   | Isao T |
| 58611129-2dbc-4a81-a72f-77ddfc1b1b49 | My Neighbor Totoro       | Hayao  |
| ea660b10-85c4-4ae3-8a5f-41cea3648e3e | Kiki's Delivery Service  | Hayao  |
| 4e236f34-b981-41c3-8c65f8c9000b94e7  | Only Yesterday           | Isao T |
| ebbb6b7c-945c-41ee-a792-de0e43191bd8 | Porco Rosso              | Hayao  |
| 1b67aa9a-2e4a-45af-ac98-64d6ad15b16c | Pom Poko                 | Isao T |
| ff24da26-a969-4f0e-ba1e-a122ead6c6e3 | Whisper of the Heart     | Yoshif |
| 0440483e-ca0e-4120-8c50-4c8cd9b965d6 | Princess Mononoke        | Hayao  |
| 45204234-adfd-45cb-a505-a8e7a676b114 | My Neighbors the Yamadas | Isao T |

Тут ми натиснули кнопку *Лоад*, і в нас завантажились данні

|      |      |          |        |      |
|------|------|----------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
|      |      |          |        |      |

ЛР.ОК24.ПІ231.24.03

Арк.

35

REST Client — Ghibli

Base URL: <https://ghibliapi.vercel.app> Resource: /films Load Refresh

Filter

| Field: | Title    | Value: | Apply | Reset |
|--------|----------|--------|-------|-------|
|        | Title    |        |       |       |
|        | Id       |        |       |       |
|        | Director |        |       |       |
|        | Year     |        |       |       |

| Id                                   | Title                    | Director |
|--------------------------------------|--------------------------|----------|
| 2ba70d1-42bb-4437-8551-e5ed5a87abe   | Castle in the Sky        | Hayao    |
| 12cfb892-aac0-4c5b-94af-521852e46d6a | Grave of the Fireflies   | Isao     |
| 58611129-2dbc-4a81-a72f-77ddfc1b1b49 | My Neighbor Totoro       | Hayao    |
| ea660b10-85c4-4ae3-8a5f-41cea3648e3e | Kiki's Delivery Service  | Hayao    |
| 4e236f34-b981-41c3-8c65-f8c9000b94e7 | Only Yesterday           | Isao     |
| ebbb6b7c-945c-41ee-a792-de0e43191bd8 | Porco Rosso              | Hayao    |
| 1b67aa9a-2e4a-45af-ac98-64d6ad15b16c | Pom Poko                 | Isao     |
| ff24da26-a969-4f0e-ba1e-a122ead6c6e3 | Whisper of the Heart     | Yoshif   |
| 0440483e-ca0e-4120-8c50-4c8cd9b965d6 | Princess Mononoke        | Hayao    |
| 45204234-adfd-45cb-a505-a8e7a676b114 | My Neighbors the Yamadas | Isao     |

Id: 2ba70d1-42bb-4437-8551-e5ed5a87abe  
 Title/Name: Castle in the Sky  
 Extra: Hayao Miyazaki  
 Description:  
 The orphan Sheeta inherited a mysterious crystal that links her to the mythical sky-kingdom of Laputa. With the help of resourceful Pazu and a rollicking band of sky pirates, she makes her way to the ruins of the

Ready 22 items

Тут ми можемо переглянути поля по яким ми можемо робити пошуки/фільтри

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 36   |

REST Client — Ghibli

Base URL: <https://ghibliapi.vercel.app> Resource: /films Load Refresh

Filter

Field: Title Value: C Apply Reset

| Id                                   | Title                           | Director        |
|--------------------------------------|---------------------------------|-----------------|
| 2ba70d1-42bb-4437-b551-e5fed5a87abe  | Castle in the Sky               | Hayao Miyazaki  |
| ea660b10-85c4-4ae3-8a5f-41cea3648e3e | Kiki's Delivery Service         | Hayao Miyazaki  |
| ebbb6b7c-945c-41ee-a792-de0e43191bd8 | Porco Rosso                     | Hayao Miyazaki  |
| 0440483e-ca0e-4120-8c50-4c8cd9b965d6 | Princess Mononoke               | Hayao Miyazaki  |
| 90b72513-afd4-4570-84de-a56c312fdf81 | The Cat Returns                 | Hiroyuki Morita |
| cd3d059c-09f4-4ff3-8d63-bc765a5184fa | Howl's Moving Castle            | Hayao Miyazaki  |
| 578ae244-7750-4d9f-867bf3cd3d6fecf4  | The Tale of the Princess Kaguya | Isao Takahata   |
| 790e0028-a31c-4626-a694-86b7a8cada40 | Earwig and the Witch            | Gorō Miyazaki   |

Id: 2ba70d1-42bb-4437-b551-e5fed5a87abe

Title/Name: Castle in the Sky

Extra: Hayao Miyazaki

Description:

The orphan Sheeta inherited a mysterious crystal that links her to the mythical sky-kingdom of Laputa. With the help of resourceful Pazu and a rollicking band of sky pirates, she makes her way to the ruins of the

Filter applied 8 items

Тут ми можемо бачити реалізований пошук по назві фільму де є літера «С»

|      |      |          |        |      |                     |      |
|------|------|----------|--------|------|---------------------|------|
| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|      |      |          |        |      |                     | 37   |

REST Client — Ghibli

Base URL: <https://ghibliapi.vercel.app> Resource: /films Load Refresh

Filter

Field: Director Value: Isao Takahata Apply Reset

|                                | Title                           | Director      | Year |
|--------------------------------|---------------------------------|---------------|------|
| 12-aac0-4c5b-94af-521852e46d6a | Grave of the Fireflies          | Isao Takahata | 1988 |
| 4-b981-41c3-8c65f8c9000b94e7   | Only Yesterday                  | Isao Takahata | 1991 |
| 9a-2e4a-45af-ac98-64d6ad15b16c | Pom Poko                        | Isao Takahata | 1992 |
| 34-adfd-45cb-a505-a8e7a676b114 | My Neighbors the Yamadas        | Isao Takahata | 1994 |
| 44-7750-4d9f-867bf3cd3d6fecf4  | The Tale of the Princess Kaguya | Isao Takahata | 2013 |

Ready 5 items

Id: 4e236f34-b981-41c3-8c65f8c9000b94e7  
Title/Name: Only Yesterday  
Extra: Isao Takahata  
Description:  
It's 1982, and Taeko is 27 years old, unmarried, and has lived her whole life in Tokyo. She decides to visit her family in the countryside, and as the train travels through the night, memories flood back of her younger

Тут ми бачимо що пошук відбувається по режисеру

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 38   |

The screenshot shows a REST Client interface with the title "REST Client — Ghibli". At the top, it displays the "Base URL: https://ghibliapi.vercel.app" and the "Resource: /films". Below this, there is a "Filter" section with a dropdown "Field: Year" set to "Year" and a "Value: 1988", with "Apply" and "Reset" buttons. A table below the filter shows two rows of film data:

|                                   | Title                  | Director       | Year |
|-----------------------------------|------------------------|----------------|------|
| fb892-aac0-4c5b-94af-521852e46d6a | Grave of the Fireflies | Isao Takahata  | 1988 |
| 11129-2dbc-4a81-a72f-77ddfc1b1b49 | My Neighbor Totoro     | Hayao Miyazaki | 1988 |

To the right of the table, a detailed view of the first film is shown in a box:

|   |
|---|
| <b>Id:</b> 12cfb892-aac0-4c5b-94af-521852e46d6a   |
| <b>Title/Name:</b> Grave of the Fireflies   |
| <b>Extra:</b> Isao Takahata   |
| <b>Description:</b>   |
| In the latter part of World War II, a boy and his sister, orphaned when their mother is killed in the firebombing of Tokyo, are left to survive on their own in what remains of civilian life in Japan. The plot follows this boy and his |

Filter applied 2 items

*А тут ми бачимо пошук по конкретному році*

**Висновок:** опрацьовано принципи клієнт-серверної архітектури та роль REST у сучасних системах. Реалізовано RESTful API з типовими CRUD-ендпойнтами, відпрацьовано роботу з HTTP-запитами/відповідями і JSON. Закріплено навички проєктування маршрутів, обробки кодів стану та помилок, що підвищує надійність взаємодії клієнта й сервера.

| Змн. | Арк. | № докум. | Підпис | Дата | ЛР.ОК24.ПІ231.24.03 | Арк. |
|------|------|----------|--------|------|---------------------|------|
|      |      |          |        |      |                     | 39   |