

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

**Комп'ютерні інформаційні мережі
ЛАБОРАТОРНА РОБОТА №9**

Тема: «Протоколи прикладного рівня. Postman».

Виконала:
Ст. Пелєщак Вероніка
ПМІ-35с

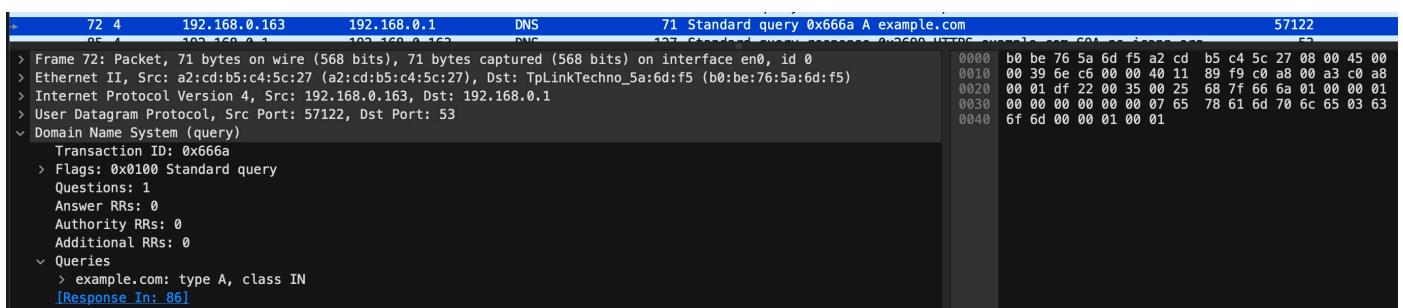
Тема: Протоколи прикладного рівня. Postman.

Мета: Здобути практичні навички з інтерпретації повідомлень протоколів прикладного рівня та опанувати на базовому рівні роботу на платформі Postman.

Хід роботи:

1. Використовуючи *Wireshark*, захопила пакети, здійснюючи активність в браузері, заходила на сайти, які працюють за протоколом **http**.
2. Відшукала серед захоплених пакетів пару *DNS-запит/відповідь*.

- **DNS-запит:**



Transaction ID: 0x666a — унікальний ідентифікатор запиту, який дозволяє серверу повернути відповідь саме на це звернення.

Flags: 0x0100 (Standard query) — повідомлення є стандартним DNS-запитом; клієнт просить виконати рекурсивний пошук (Recursion Desired = 1).

Questions: 1 — у запиті міститься одне питання.

Answer RRs: 0, Authority RRs: 0, Additional RRs: 0 — у запиті немає відповідей або додаткових записів (вони будуть у відповіді).

Queries → example.com, type A, class IN — клієнт запитує IPv4-адресу (тип A) домену *example.com* у класі IN (Internet).

Response In: 86 — Wireshark показує, що відповідь на даний запит міститься у пакеті №86.

- **DNS-відповідь:**

```
Frame 86: Packet, 167 bytes on wire (1336 bits), 167 bytes captured (1336 bits) on interface en0, id 0
Ethernet II, Src: TpLinkTechno_5a:6d:f5 (b0:be:76:5a:6d:f5), Dst: a2:cd:b5:c4:5c:27 (a2:cd:b5:c4:5c:27)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.163
User Datagram Protocol, Src Port: 53, Dst Port: 57122
Domain Name System (response)
  Transaction ID: 0x666a
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 6
  Authority RRs: 0
  Additional RRs: 0
  Queries
    > example.com: type A, class IN
  Answers
    > example.com: type A, class IN, addr 23.192.228.80
    > example.com: type A, class IN, addr 23.220.75.245
    > example.com: type A, class IN, addr 23.220.75.232
    > example.com: type A, class IN, addr 23.215.0.136
    > example.com: type A, class IN, addr 23.192.228.84
    > example.com: type A, class IN, addr 23.215.0.138
[Request In: 72]
[Time: 9.510000 milliseconds]
```

Transaction ID: 0x666a — збігається із запитом, що підтверджує відповідність пакету.

Flags: 0x8180 (Standard query response, No error) — це успішна відповідь; сервер підтримує рекурсію (Recursion Available = 1) та не є авторитетним для домену.

Questions: 1 — повторює питання із запиту.

Answer RRs: 6 — сервер повернув шість IPv4-адрес, пов'язаних із доменом *example.com*.

Authority RRs: 0, Additional RRs: 0 — додаткові записи відсутні.

Answers — наведені шість записів типу A, кожен з яких містить одну з IP-адрес домену:

- 23.192.228.80
- 23.220.75.245

- 23.220.75.232

- 23.215.0.136

- 23.192.228.84

- 23.215.0.138

Request In: 72 — вказує, що ця відповідь є реакцією саме на DNS-запит із пакету №72.

3. За допомогою фільтру http було відображенено відповідні пакети.

No.	Time	Source	Destination	Protocol	Length	Info	Source Port	Host
138	5	192.168.0.163	23.192.228.80	HTTP	446	GET / HTTP/1.1	56493	example.com
140	5	23.192.228.80	192.168.0.163	HTTP	760	HTTP/1.1 200 OK (text/html)	80	
145	5	192.168.0.163	23.192.228.80	HTTP	346	GET /apple-touch-icon-precomposed.png HTTP/1.1	56494	example.com
147	6	23.192.228.80	192.168.0.163	HTTP	970	HTTP/1.1 404 Not Found (text/html)	80	
149	6	192.168.0.163	23.192.228.80	HTTP	334	GET /apple-touch-icon.png HTTP/1.1	56494	example.com
151	6	23.192.228.80	192.168.0.163	HTTP	970	HTTP/1.1 404 Not Found (text/html)	80	
1106	18	192.168.0.163	34.223.124.45	HTTP	447	GET / HTTP/1.1	56503	neverssl.com
1111	18	34.223.124.45	192.168.0.163	HTTP	891	HTTP/1.1 200 OK (text/html)	80	
2571	37	192.168.0.163	146.190.62.39	HTTP	426	GET /css/style.min.css HTTP/1.1	56507	httpforever.com
2602	38	146.190.62.39	192.168.0.163	HTTP	1388	HTTP/1.1 200 OK (text/css)	80	
2645	38	192.168.0.163	146.190.62.39	HTTP	550	GET /css/images/header-major-on-dark.svg HTTP/1.1	56508	httpforever.com
2651	38	146.190.62.39	192.168.0.163	HTTP/XML	1349	HTTP/1.1 200 OK	80	
3876	69	192.168.0.163	188.184.67.127	HTTP	528	GET /hypertext/README.html HTTP/1.1	56511	info.cern.ch
3884	69	188.184.67.127	192.168.0.163	HTTP	97	HTTP/1.1 200 OK (text/html)	80	
4030	78	192.168.0.163	23.215.0.132	HTTP	446	GET / HTTP/1.1	56514	example.org
4041	78	23.215.0.132	192.168.0.163	HTTP	760	HTTP/1.1 200 OK (text/html)	80	
4043	78	192.168.0.163	23.215.0.132	HTTP	397	GET /favicon.ico HTTP/1.1	56514	example.org
4049	78	23.215.0.132	192.168.0.163	HTTP	970	HTTP/1.1 404 Not Found (text/html)	80	
4312	102	192.168.0.163	34.223.124.45	HTTP	501	GET /online HTTP/1.1	56518	slowlushquietzen.
4315	102	34.223.124.45	192.168.0.163	HTTP	593	HTTP/1.1 301 Moved Permanently (text/html)	80	
4317	102	192.168.0.163	34.223.124.45	HTTP	502	GET /online/ HTTP/1.1	56518	slowlushquietzen.
4326	102	34.223.124.45	192.168.0.163	HTTP	137	HTTP/1.1 200 OK (text/html)	80	
4328	102	192.168.0.163	34.223.124.45	HTTP	440	GET /favicon.ico HTTP/1.1	56518	slowlushquietzen.
4329	102	34.223.124.45	192.168.0.163	HTTP	482	HTTP/1.1 200 OK (PNG)	80	

4. Аналіз HTTP-запиту та HTTP-відповіді.

• HTTP-запит:

No.	Time	Source	Destination	Protocol	Length	Info	Source Port	Host
138	5	192.168.0.163	23.192.228.80	HTTP	446	GET / HTTP/1.1	56493	example.com
>	Frame 138: Packet: 446 bytes on wire (3568 bits), 446 bytes captured (3568 bits) on interface en0, id 0						0000	b0 76 5a 6d f5 a2 c
>	Ethernet II, Src: a2:cd:b5:c4:5c:27 (a2:cd:b5:c4:5c:27), Dst: TpLinkTechno_5a:6d:f5 (b0:be:76:5a:6d:f5)						0010	01 b0 00 00 40 00
>	Internet Protocol Version 4, Src: 192.168.0.163, Dst: 23.192.228.80						0020	e4 50 dc ad 00 50 88 8
>	Transmission Control Protocol, Src Port: 56493, Dst Port: 80, Seq: 1, Ack: 1, Len: 380						0030	08 0b 9b 6b 00 00 01 0
>	Hypertext Transfer Protocol						0040	96 ee 47 45 54 20 2f 2
>	GET / HTTP/1.1\r\n						0050	0d 0a 48 6f 73 74 3a 2
>	Request Method: GET						0060	63 6f 6d 0d 0a 55 73 6
>	Request URI: /						0070	20 4d 6f 7a 69 6c 6c 6
>	Request Version: HTTP/1.1						0080	63 69 6e 74 6f 73 68 3
>	Host: example.com\r\n						0090	61 63 20 4f 53 20 58 2
>	User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/26.0.1 Safari/605.1.15\r\n						00a0	20 41 70 70 6c 65 57 6
>	Upgrade-Insecure-Requests: 1\r\n						00b0	2e 31 2e 31 35 20 28 4
>	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n						00c0	6b 65 20 47 65 63 6b 6
>	Accept-Language: en-GB,en;q=0.9\r\n						00d0	6e 2f 32 36 2e 30 2e 3
>	Priority: u=0, i\r\n						00e0	36 30 35 2e 31 2e 31 3
>	Accept-Encoding: gzip, deflate\r\n						00f0	65 2d 49 6e 73 65 63 7
>	Connection: keep-alive\r\n						0100	73 74 73 3a 20 31 0d 0
>	\r\n						0110	74 65 78 74 2f 68 74 6
>	[Response in frame: 140]						0120	61 74 69 6f 6e 2f 78 6
>	[Full request URI: http://example.com/]						0130	61 70 6c 69 63 61 7
>							0140	71 3d 30 2e 39 2c 2a 2
>							0150	0a 41 63 63 65 70 74 2
>							0160	3a 20 65 6e 2d 47 42 2
>							0170	0d 0a 50 72 69 6f 72 6
>							0180	20 69 0d 0a 41 63 63 6

1. GET / HTTP/1.1

- **GET** — метод запиту, означає отримання ресурсу без надсилання тіла.
- **/** — шлях до ресурсу; у цьому випадку запитується головна сторінка сайту.
- **HTTP/1.1** — версія протоколу, яку використовує клієнт.

2. Host: example.com

- Вказує домен, до якого звертається браузер.
- Необхідний елемент HTTP/1.1, оскільки один сервер може обслуговувати багато сайтів.

3. User-Agent: Mozilla/5.0 (Macintosh...)

- Ідентифікує браузер, ОС, рендеринг-двигун.
- Використовується сервером для адаптації контенту (мова, формат, кодування).

4. Upgrade-Insecure-Requests: 1

- Браузер сигналізує, що **віддає перевагу HTTPS**, якщо сервер може перенаправити.
- Це механізм підвищення безпеки доступу.

5. Accept: text/html,application/xhtml+xml,...

- Список форматів, які клієнт може приймати.

- Перший елемент — **text/html**, тобто клієнт очікує HTML-сторінку.

6. Accept-Language: en-GB,en;q=0.9

- Визначає бажану мову відповідей.
- Параметр q задає пріоритет (чим вище, тим важливіше).

7. Priority: u=0, i

- Інформація про пріоритетність запиту.
- Використовується сучасними браузерами для оптимізації завантаження ресурсів.

8. Accept-Encoding: gzip, deflate

- Клієнт дозволяє надсилати відповідь у стисненому вигляді.
- Це пришвидшує передачу та економить трафік.

9. Connection: keep-alive

- Просить не закривати TCP-з'єднання після відповіді.
- Дозволяє виконувати кілька запитів напряму — швидше та ефективніше.

10. Response in frame: 140

- Wireshark показує, у якому пакеті знаходиться відповідь на це питання.

- **HTTP-відповідь:**

140 5 23.192.228.80 192.168.0.163 HTTP 760 HTTP/1.1 200 OK (text/html) 80

> Frame 140: Packet, 760 bytes on wire (6080 bits), 760 bytes captured (6080 bits) on interface en0, id 0

> Ethernet II, Src: TP-Link Techno_5a:6d:f5 (00:0c:29:5a:6d:f5), Dst: a2:cd:b5:c4:5c:27 (a2:cd:b5:c4:5c:27)

> Internet Protocol Version 4, Src: 23.192.228.80, Dst: 192.168.0.163

> Transmission Control Protocol, Src Port: 80, Dst Port: 56493, Seq: 1, Ack: 381, Len: 694

HTTP Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Response Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK

Accept-Ranges: bytes\r\nContent-Type: text/html\r\nETag: "bc2473a18e003bdb249eba5ce893033f:1760028122.592274"\r\nLast-Modified: Thu, 09 Oct 2025 16:42:02 GMT\r\nVary: Accept-Encoding\r\nContent-Encoding: gzip\r\nCache-Control: max-age=86000\r\nDate: Thu, 13 Nov 2025 18:29:10 GMT\r\nContent-Length: 363\r\nConnection: keep-alive\r\n\r\n[Request in frame: 138]
[Time since request: 209.846000 milliseconds]
[Request URI: /]
[Full request URL: http://example.com/]
Content-encoded entity body (gzip): 363 bytes -> 513 bytes
File Data: 513 bytes

Line-based text data: text/html (1 lines)

0000 a2 cd b5 c4 5c 27 b0
0010 02 ea 9c b2 40 00 30
0020 00 a3 00 50 dc ad d2
0030 01 fb 7e 36 00 00 01
0040 34 e4 48 54 54 50 2f
0050 4b 0d 0a 41 63 63 65
0060 3a 20 62 79 74 65 73
0070 2d 54 79 70 65 3a 20
0080 0d 0a 45 54 61 67 3a
0090 31 38 65 30 30 33 62
00a0 63 65 38 33 30 33 33
00b0 38 31 32 32 2e 35 39
00c0 73 74 2d 4d 6f 64 69
00d0 2c 20 30 39 20 4f 63
00e0 3a 34 32 3a 30 32 20
00f0 3a 20 41 63 63 65 70
0100 67 0d 0a 43 6f 6e 74
0110 69 6e 67 3a 20 67 7a
0120 2d 43 6f 6e 74 72 6f
0130 65 3d 38 36 30 30 30
0140 68 75 2c 20 31 33 20
0150 31 38 3a 32 39 3a 31
0160 6e 74 65 6e 74 2d 4c
0170 33 0d 0a 43 6f 6e 6e
0180 65 65 70 2d 61 6c 69
0190 00 00 00 00 00 00 00
01a0 14 a1 5c b7 0d 15 3f
01b0 02 a6 71 5b 8b fc 54
01c0 b8 44 f1 c4 e3 f1 4c
01d0 6d f5 f7 29 2c f8 a1
01e0 8b ed f3 07 b8 c9 a2
01f0 06 e1 c1 61 53 2c 84
0200 a6 38 91 e1 b1 31 b8
0210 5b a6 0e 2c 36 75 96
0220 e8 de 87 18 66 6f d4

1. HTTP/1.1 200 OK

- **HTTP/1.1** — версія протоколу, яку використовує сервер.
- **200** — код успішної відповіді.
- **OK** — текстове пояснення статусу.

2. Accept-Ranges: bytes

- Сервер дозволяє клієнту завантажувати частини файлу (наприклад, докачування).
- Корисно при нестабільних з'єднаннях або великих файлах.

3. Content-Type: text/html

- Вказує MIME-тип ресурсу.
- У цьому випадку сервер повертає HTML-сторінку.

4. ETag: "bc2473a1e..."

- Унікальний ідентифікатор конкретної версії файлу.
- Дозволяє браузеру перевіряти, чи змінився ресурс (для кешування).

5. Last-Modified: Thu, 09 Oct 2025 16:42:02 GMT

- Дата останньої зміни файлу на сервері.
- Браузер може використовувати це для перевірки кешу.

6. Vary: Accept-Encoding

- Сервер може повернати різні версії ресурсу залежно від заголовків у запиті.
- Наприклад: gzip, deflate або нестиснена версія.

7. Content-Encoding: gzip

- Відповідь стиснена GZIP.
- Зменшує об'єм даних, що передаються.

8. Cache-Control: max-age=86000

- Період, протягом якого клієнт може зберігати відповідь у кеші, не запитуючи її повторно.
- 86 000 секунд ≈ майже 1 доба.

9. Date: Thu, 13 Nov 2025 18:29:10 GMT

- Час формування цієї відповіді сервером.
- Важливий заголовок для синхронізації кешу.

10. Content-Length: 363

- Розмір поверненого тіла відповіді (у стислому вигляді).
- Браузер розуміє, скільки даних очікувати.

11. Connection: keep-alive

- Сервер погоджується залишити TCP-з'єднання відкритим.
- Покращує продуктивність завантаження сторінки.

12. Request in frame: 138

- Сервер указує, на який запит ця відповідь.
- Важливо для аналізу трафіку — підтверджує кореляцію.

13. Тіло відповіді (HTML-код)

- Wireshark відображає HTML-дані у стислому gzip-форматі.
- Саме ті дані браузер відображає як веб-сторінку.

5. У процесі перехоплення HTTP-трафіку були зафіковані лише запити типу **GET**. Інших типів (таких як POST, HEAD, PUT, DELETE тощо) не спостерігалося. Це пояснюється тим, що при відкритті звичайних вебсторінок браузер використовує переважно саме метод **GET**, оскільки він призначений для отримання ресурсів із сервера — HTML-сторінок, стилів, зображень, іконок тощо. Методи на кшталт **POST** чи **PUT** зазвичай використовуються при надсиланні даних на сервер (заповнення форм, авторизація, робота з API), а таких дій під час захоплення трафіку не виконувалося. Саме тому в трафіку вони відсутні. Однак, у процесі аналізу HTTP-трафіку були перехоплені декілька типів HTTP-відповідей:

- **200 OK** — успішна відповідь сервера, ресурс повернуто коректно (HTML-сторінка, зображення, CSS-файл тощо).
- **301 Moved Permanently** — сервер повідомляє про постійне перенаправлення ресурсу на іншу адресу. Браузер автоматично виконує повторний GET-запит.
- **404 Not Found** — сервер не знайшов запитуваний файл (наприклад, деякі іконки, які браузер запитує автоматично).
- Також були видимі 200 OK з різними MIME-типами: **text/html**, **text/css**, **image/png**, **image/svg+xml**, що означає успішне повернення відповідного типу ресурсу.

Це відповідає типовій поведінці веб-браузера, який при відкритті сторінки завантажує не лише HTML, але й пов'язані ресурси (зображення, стилі, іконки).

6. Відкрила вебверсію *Postman* та створила новий персональний робочий простір. Створила колекцію.

A screenshot of the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'History', and 'Flows'. The main area shows a collection named 'weather' under 'ComputerNetworks'. A message says 'This collection is empty. Add a request to start working.' Below the collection name, tabs for 'Overview', 'Authorization', 'Scripts', 'Variables', and 'Runs' are visible. A note on the right says 'Make things easier for your teammates with a complete collection description.'

7. Натиснула на кнопку **New** і обрала тип запиту **HTTP**. Ввела адресу кінцевої точки <https://api.openweathermap.org/data/2.5/weather> у поле Enter.

A screenshot of the Postman request configuration screen. It shows a GET request to the URL <https://api.openweathermap.org/data/2.5/weather>. Below the URL, tabs for 'Docs', 'Params' (selected), 'Authorization', 'Headers (7)', 'Body', 'Scripts', and 'Settings' are visible. A 'Send' button is on the right.

8. Визначилась з типом API-запиту про поточну погоду: за назвою міста.

Parameters

<code>q</code>	required	City name, state code and country code divided by comma, Please refer to ISO 3166 for the state codes or country codes. You can specify the parameter not only in English. In this case, the API response should be returned in the same language as the language of requested location name if the location is in our predefined list of more than 200,000 locations.
<code>appid</code>	required	Your unique API key (you can always find it on your account page under the " API key " tab)
<code>mode</code>	optional	Response format. Possible values are <code>xml</code> and <code>html</code> . If you don't use the <code>mode</code> parameter format is JSON by default. Learn more
<code>units</code>	optional	Units of measurement. <code>standard</code> , <code>metric</code> and <code>imperial</code> units are available. If you do not use the <code>units</code> parameter, <code>standard</code> units will be applied by default. Learn more
<code>lang</code>	optional	You can use this parameter to get the output in your language. Learn more

9. Перейшла на вкладку **Params** і задайла імена та значення параметрів.

GET https://api.openweathermap.org/data/2.5/weather ?q=Novoivorivsk&units=metric&lang=ua

Params

Key	Value	Description
q	Novoivorivsk	
units	metric	
lang	ua	
Key	Value	Description

10. На вкладці **Authorization** задала параметри (тип авторизації, назву параметра у запиті, значення авторизаційного ключа і місце авторизаційної інформації у API-запиті).

Auth Type

API Key

The authorization header will be automatically generated when you send the request. Learn more about [API Key](#).

Key

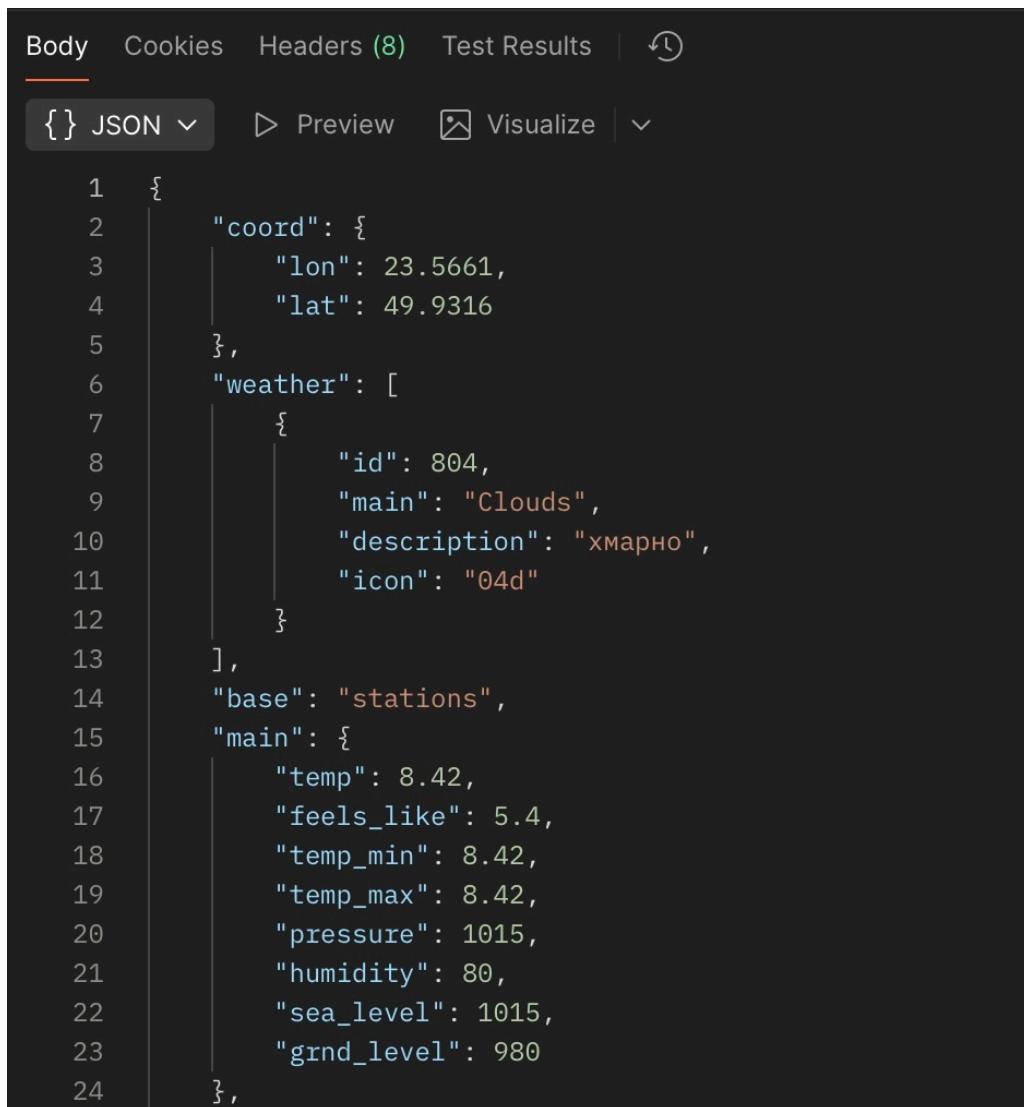
Value

Add to

11. Виконала запит.

- **coord** — географічні координати місцевості (широта й довгота), за якими визначається точне розташування міста.
- **weather** — загальний стан погоди: тип явища (наприклад, хмарність), короткий опис, код іконки для відображення.
- **base** — службове поле, яке вказує джерело або тип станції, що передає дані.
- **main** — основні метеорологічні параметри: температура, «відчувається як», мінімальна та максимальна температура, атмосферний тиск, вологість.

- **visibility** — максимальна дистанція видимості у метрах.
- **wind** — дані про вітер: швидкість, напрямок та пориви.
- **clouds** — рівень хмарності у відсотках.
- **dt** — час формування даних у форматі *Unix time*.
- **sys** — додаткова інформація: країна, час сходу та заходу сонця.
- **timezone** — часовий пояс для регіону (у секундах від *UTC*).
- **id** — унікальний ідентифікаційний номер населеного пункту.
- **name** — назва міста, для якого отримано дані.
- **cod** — статус відповіді сервера (200 означає, що запит виконано успішно).



The screenshot shows a JSON viewer interface with the following details:

- Headers (8)**: Shows 8 headers in the request.
- Body**: The main content area displays a JSON object with the following structure:

```

1  {
2    "coord": {
3      "lon": 23.5661,
4      "lat": 49.9316
5    },
6    "weather": [
7      {
8        "id": 804,
9        "main": "Clouds",
10       "description": "хмарно",
11       "icon": "04d"
12     }
13   ],
14   "base": "stations",
15   "main": {
16     "temp": 8.42,
17     "feels_like": 5.4,
18     "temp_min": 8.42,
19     "temp_max": 8.42,
20     "pressure": 1015,
21     "humidity": 80,
22     "sea_level": 1015,
23     "grnd_level": 980
24   }

```

```

25     "visibility": 10000,
26     "wind": {
27         "speed": 5.56,
28         "deg": 296,
29         "gust": 9.45
30     },
31     "clouds": {
32         "all": 99
33     },
34     "dt": 1763195481,
35     "sys": {
36         "country": "UA",
37         "sunrise": 1763185101,
38         "sunset": 1763217752
39     },
40     "timezone": 7200,
41     "id": 708313,
42     "name": "Novoiaivorivsk",
43     "cod": 200
44 }
```

12. Зберігши запит у колекцію, створила копію запиту, додала до переліку параметрів параметр **mode** зі значенням **html** і виконала запит.

The screenshot shows a browser developer tools Network tab with the following details:

- Request URL: <http://api.openweathermap.org/data/2.5/weather?q=Novoiaivorivsk&mode=html>
- Status: 200 OK
- Time: 329 ms
- Size: 2.51 kB
- Headers (8): Includes Content-Type: application/json, Content-Length: 1015, Date: Mon, 17 Jul 2017 11:10:00 GMT, Server: Apache/2.4.10 (Ubuntu), Vary: Accept-Encoding, X-Powered-By: PHP/7.0.10-2+deb.sury.org~wheezy+1, and more.
- Body (HTML Preview): Shows the weather data for Novoiaivorivsk, including temperature (8.42°C), weather icon (cloudy), and various atmospheric conditions like clouds (99%), humidity (80%), wind (5.56 m/s), and pressure (1015hPa). A "More.." link is present.

13. Виконала додатковий запит до іншого погодного API: http://api.openweathermap.org/data/2.5/air_pollution.

The screenshot shows the Postman interface with a GET request to http://api.openweathermap.org/data/2.5/air_pollution?lat=49.8397&lon=24.0297. The 'Params' tab is selected, displaying four parameters: appid, lat, and lon (all checked), and a Key field (unchecked). A 'Go to authorization' link is visible.

The screenshot shows the Postman interface after sending the request. The response status is 200 OK, with a duration of 419 ms and a body size of 507 B. The response body is displayed as JSON:

```
1 {  
2   "coord": {  
3     "lon": 24.032,  
4     "lat": 49.842  
5   },  
6   "list": [  
7     {  
8       "main": {  
9         "aqi": 2  
10      },  
11      "components": {  
12        "co": 121.58,  
13        "no": 0.11,  
14        "no2": 0.7,  
15        "o3": 81.98,  
16        "so2": 3.38,  
17        "pm2_5": 10.63,  
18        "pm10": 13.19,  
19        "nh3": 0.01  
20      },  
21      "dt": 1763202396  
22    }  
23  ]  
24 }
```

Для виконання даного пункту було використано додатковий API сервісу **OpenWeather — Air Pollution API**. Даний сервіс дозволяє отримувати інформацію про поточний рівень забруднення повітря за заданими координатами.

Запит виконувався методом **GET** з такими параметрами:

- **lat** — широта міста (49.8397, м. Львів);
- **lon** — довгота (24.0297);
- **appid** — унікальний API-ключ користувача.

У відповідь сервіс повернув JSON-об'єкт, який містить показники якості повітря.

Основні значення якості повітря

- **aqi = 2** — загальний індекс якості повітря.

Значення 2 відповідає категорії «задовільно», тобто повітря не містить значних концентрацій забруднювачів і є відносно безпечним.

Концентрації забруднюючих речовин (components)

У цьому блоці наведено дані про основні домішки в атмосфері (в мкг/м³):

- **co = 121.58** — концентрація чадного газу (CO);
- **no = 0.11** — оксид азоту;
- **no2 = 0.7** — діоксид азоту (NO₂), один із основних транспортних забруднювачів;
- **o3 = 81.98** — озон, характерний для денних годин;
- **so2 = 3.38** — діоксид сірки (SO₂);
- **pm2_5 = 10.63** — дрібнодисперсний пил PM2.5, найнебезпечніший забруднювач;
- **pm10 = 13.19** — пил PM10;
- **nh3 = 0.01** — амоніак.

Усі показники знаходяться на низькому або помірному рівні.

Час вимірювання (dt)

Параметр dt = 1763202396 — це UNIX *timestamp*, який вказує час проведення вимірювання.

Висновок: У ході лабораторної роботи було досліджено роботу протоколів прикладного рівня на прикладі HTTP та DNS, проаналізовано структуру запитів і відповідей у Wireshark та визначено призначення ключових полів. За допомогою Postman було сформовано та виконано API-запити до сервісу OpenWeather для отримання поточної погоди в різних форматах, а також виконано додатковий запит до Air Pollution API для отримання інформації про якість повітря. Виконані завдання дозволили закріпити практичні навички роботи з мережевими протоколами та REST API й краще зрозуміти взаємодію між клієнтом і сервером у сучасних вебсервісах.