

15.2 Робота з колекціями та документацією

Сайт: [ІТ курси](#)

Курс: QA manual v.5.0/2024

Книга: 15.2 Робота з колекціями та документацією

Надруковано: Мусятовська Олена

Дата: субота 5 квітня 2025 02:46

Зміст

[Що таке колекція](#)

[Колекція можливості](#)

[Щоб створити нову колекцію](#)

[Види колекцій](#)

[Документація та як з нею працювати](#)

[Документація](#)

[Документація](#)

[Спробуємо надіслати запит](#)

[Очікуваний результат](#)

[Вітаю вас!](#)

Що таке колекція

Що таке колекція



Колекція - це не що інше, як контейнер або папка для зберігання запитів Postman. Простіше кажучи, це сукупність запитів Postman. Колекції відіграють важливу роль в організації запитів, що належать одній програмі тощо.

Наприклад, якщо ви тестуєте або перевіряєте спокійний API, який має 10 кінцевих точок. Тоді має сенс організувати їх у колекцію, яка полегшить такі речі, як застосування змінних колекції, імпорт / експорт, і їх можна буде запускати як частину однієї колекції.



Колекція можливості

Колекція можливості

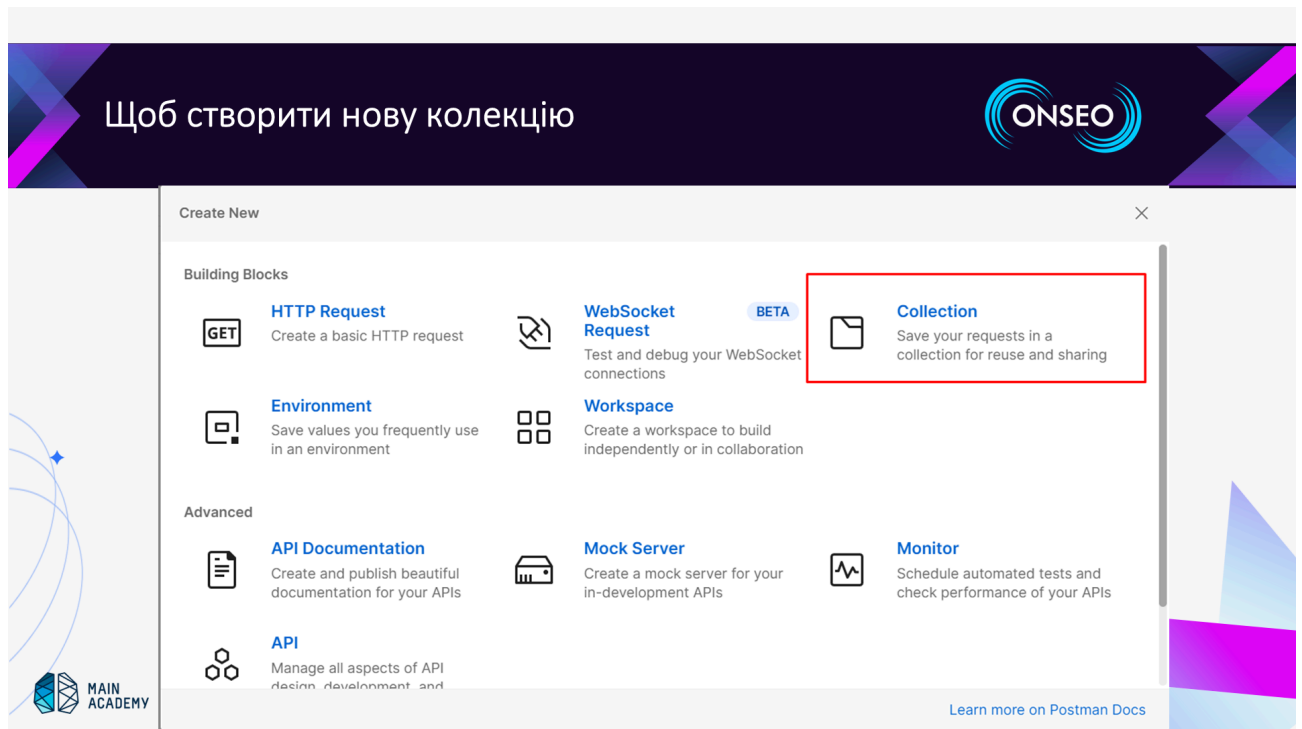


Колекція дозволяє користувачеві:

1. Запустити усі запити одночасно.
2. Розмістити змінні рівня колекції, які можуть застосовуватися до всіх запитів у цій колекції.
Наприклад, замість того, щоб окремо додавати заголовки до кожного запиту, ви можете просто застосувати заголовки до всіх запитів у цій колекції Postman, використовуючи сценарії попереднього запиту або заголовки авторизації.
3. Колекціями можна ділитися з іншими користувачами у форматі JSON або через URL-адреси як розміщені колекції на сервері
4. Виконання загальних тестів для всіх запитів, що належать до колекції.
Наприклад, якщо вам потрібно перевірити код стану для кожного запиту в колекції як HTTP 200, то замість того, щоб додавати цей тест до всіх окремих запитів, ви можете просто додати все це на рівні колекції, і це застосовуватиметься до всіх запитів, коли колекція виконується.



Щоб створити нову колекцію



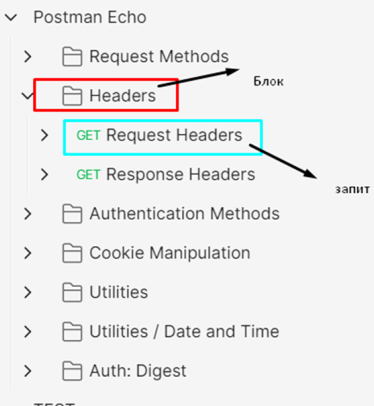

Види колекцій


Види колекцій

У лівій частині функціональні блоки API, розділені за категоріями. В цілому ми самостійно вирішуємо на які блоки буде поділяться ПЗ. Кожен блок містить свої запити, які відносяться до категорії

Folder
Папка — використовується для об'єднання запитів до групи всередині колекції. Наприклад, ви можете створити папку для першої версії свого API - "v1", а всередині згрупувати запити за змістом виконуваних дій - "Order & Checkout", "User profile" тощо. Все обмежується лише вашою фантазією та потребами. Папка, як і колекція може мати свої скрипти, але не змінні.

Request
Запит — основна складова колекції, то заради чого все й починалося. Запит створюється у конструкторі. Конструктор запитів – це головний простір, з яким вам доведеться працювати. Postman вміє виконувати запити за допомогою всіх стандартних методів HTTP, всі параметри запиту під вашим контролем.



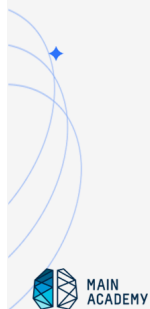


Документація та як з нею працювати

Документація та як з нею працювати



Ми будемо працювати з документацією **postman.echo**
<https://docs.postman-echo.com/?version=latest>



Postman Echo

Postman Echo is service you can use to test your REST clients and make sample API calls. It provides endpoints for `GET`, `POST`, `PUT`, various auth mechanisms and other utility endpoints.

The documentation for the endpoints as well as example responses can be found at <https://postman-echo.com>

Request Methods

HTTP has multiple request "verbs", such as `GET`, `PUT`, `POST`, `DELETE`, `PATCH`, `HEAD`, etc.


An HTTP Method (verb) defines how a request should be interpreted by a server. The endpoints in this section demonstrate various HTTP Verbs. Postman supports all the HTTP Verbs, including some rarely used ones, such as `PROPFIND`, `UNLINK`, etc.

For details about HTTP Verbs, refer to [RFC 2616](#)

Документація

ONSEO

В даному блоці описано інформацію про те, який метод потрібно використовувати, URL для надсилання запиту, параметри, які потрібно передавати в тілі запиту.



GET GET Request

Open Request →

https://postman-echo.com/get?foo1=bar1&foo2=bar2

The HTTP `GET` request method is meant to retrieve data from a server. The data is identified by a unique URI (Uniform Resource Identifier).

A `GET` request can pass parameters to the server using "Query String Parameters". For example, in the following request,

```
http://example.com/hi/there?hand=wave
```

The parameter "hand" has the value "wave".

This endpoint echoes the HTTP headers, request parameters and the complete URI requested.

Query Params

foo1	bar1
foo2	bar2

В даному блоці описано інформацію про очікуваний результат.



GET Request Woops

json

```
{
  "foo1": "bar1",
  "foo2": "bar2"
},
{
  "headers": {
    "x-forwarded-proto": "https",
    "host": "postman-echo.com",
    "accept": "*/*",
    "accept-encoding": "gzip, deflate",
    "cache-control": "no-cache",
    "postman-token": "5c27cd7d-6b16-4e5a-a0ef-191c9a3a275f",
    "user-agent": "PostmanRuntime/7.6.1",
    "x-forwarded-port": "443"
  },
  "url": "https://postman-echo.com/get?foo1=bar1&foo2=bar2"
}
```

Спробуємо надіслати запит

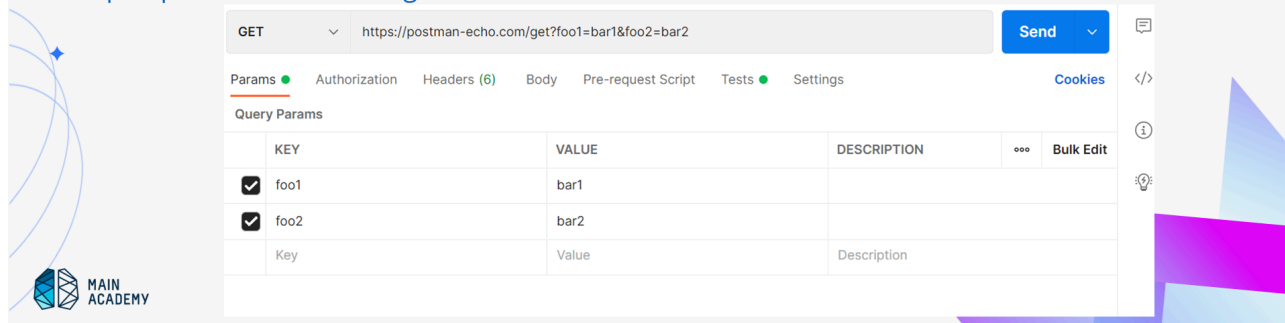
Спробуємо надіслати запит



По-перше, потрібно вибрати метод, який буде використано. У прикладі використовується метод "GET", його потрібно вибрати в Postman.


По-друге, у поле введення URL потрібно ввести також URL, який знаходиться в центральній частині документації:

<https://postman-echo.com/get?foo1=bar1&foo2=bar2>





Очікуваний результат

Очікуваний результат




Params ● Authorization Headers (7) Body Pre-request Script Tests ● Settings Cookies

Body Cookies (1) Headers (7) Test Results (2/2) 200 OK 984 ms 730 B Save Response

Pretty Raw Preview Visualize JSON  

```
1  {
2    "args": {
3      "foo1": "bar1",
4      "foo2": "bar2"
5    },
6    "headers": {
7      "x-forwarded-proto": "https",
8      "x-forwarded-port": "443",
9      "host": "postman-echo.com",
10     "x-amzn-trace-id": "Root=1-62c03838-1856e0f4118c146b017011d7",
11     "user-agent": "PostmanRuntime/7.28.4",
12     "accept": "*/*",
13     "postman-token": "8480c990-dadb-4ef8-aef4-afc2c1a136a5",
14     "accept-encoding": "gzip, deflate, br"
15   },
16   "url": "https://postman-echo.com/get?foo1=bar1&foo2=bar2"
```



Вітаю вас!

