Вступ до Postman та тестування API для початківців

Зміст

Розділ І. ВСТУП ДО POSTMAN

1.1 Як встановити postman

Postman є безкоштовним для використання. Є два способи, як ним користуватися:

- Перейти в браузері за посиланням postman.com
- Як автономний додаток, який вам треба завантажити на свій комп'ютер(доступний для Windows, macOS, або Linux)

Не використовуйте застарілий додаток Google Chrome

Перейдіть на postman.com та створіть свій акаунт.(Рис. 1.1)

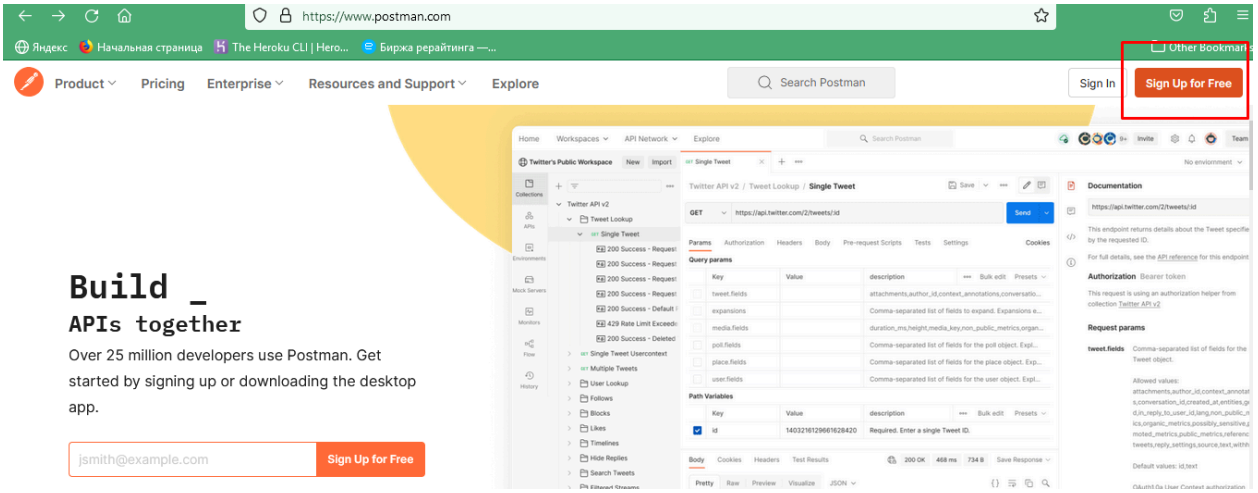

Рис. 1.1 Реєстрація на сайті postman.com

Після реєстрації переходимо до графи workspaces та обираємо my workspace. (Рис. 1.2). В цьому робочому середовищі будемо працювати.

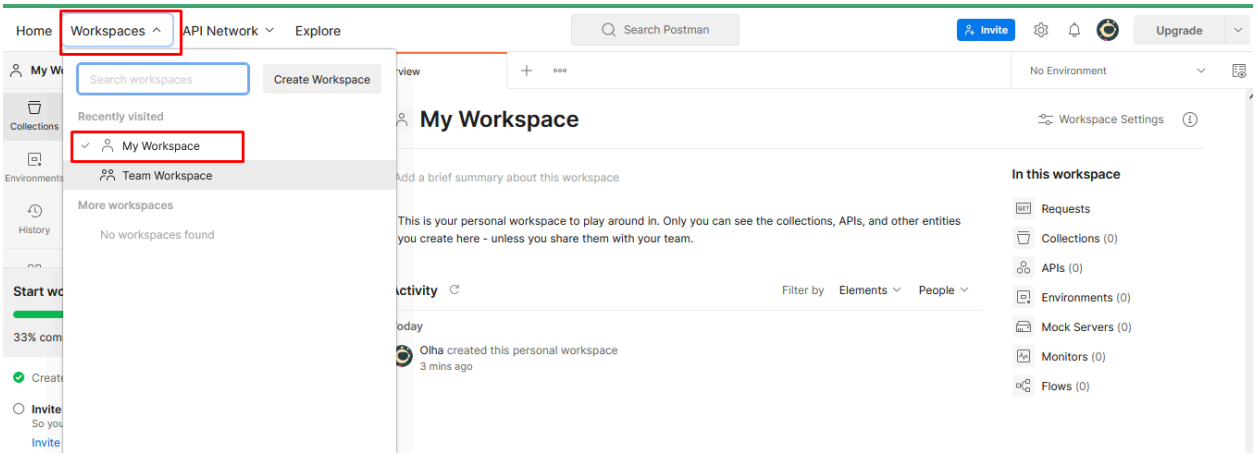

Рис.1.2 Робоче середовище

1.2 Як оновити postman

Оновлення postman надзвичайно важливо, так як нова версія може включати виправлення помилок, нові функції, або важливі оновлення безпеки.

Вам прийде повідомлення, коли вийде нова версія, але ви також можете перевірити наявність нової версії в меню додатку. (Рис.1.3)

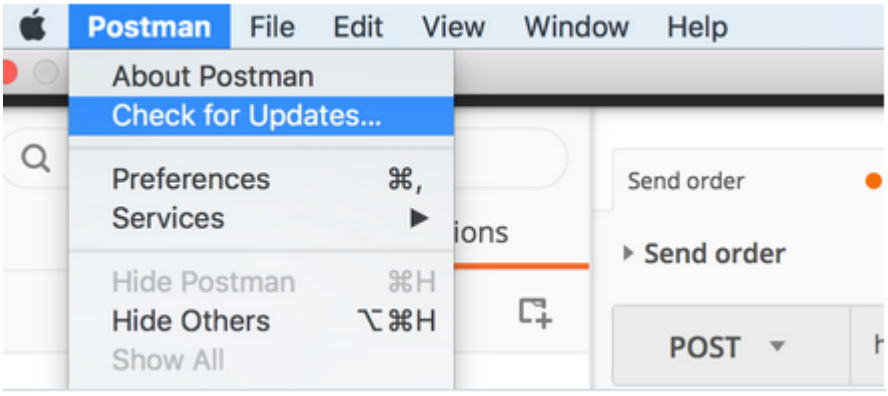

Рис. 1.3 Перевірка оновлених версій

1.3 Вступ до API

Інтерфейс прикладного програмування або API це готові конструкції мови програмування, що дозволяють розробнику будувати складну функціональність із меншими зусиллями. Вони "приховують" складніший код від програміста, забезпечуючи простоту використання.

Додати API можна в розділі My Workspace. (Рис.1.4)

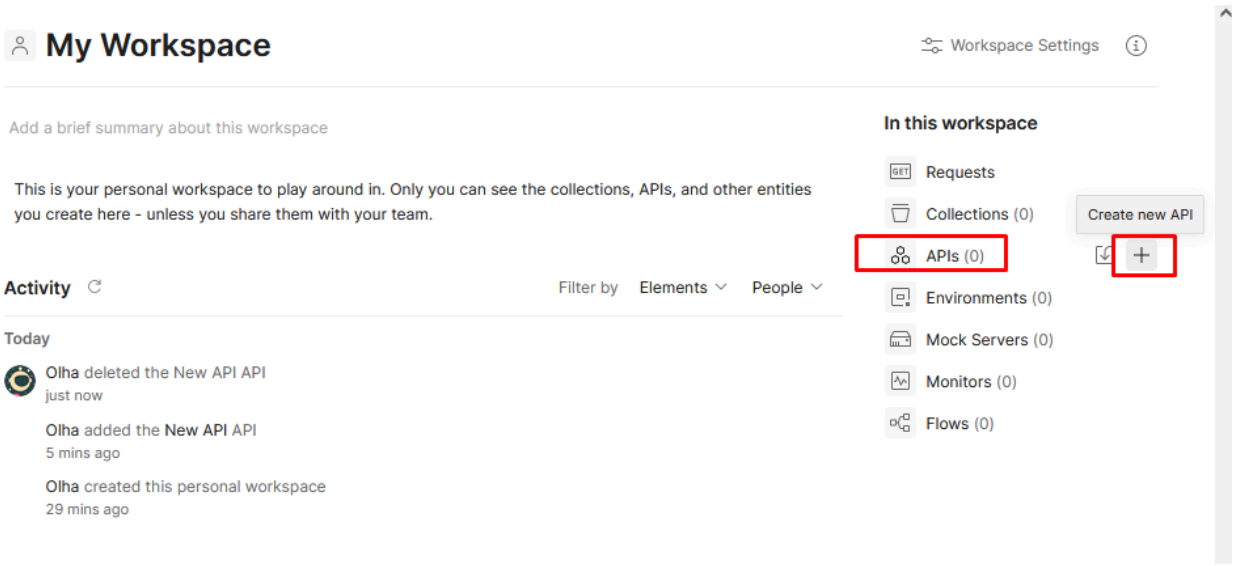

Рис. 1.4 Створити API

У вікні, яке відкриється додати посилання на запит. Для цього натисніть на плюс. (Рис.1.5)

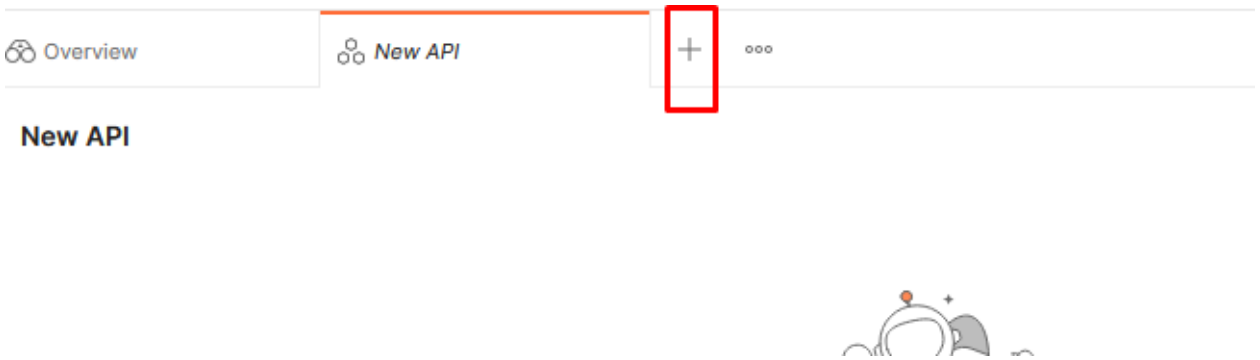

Рис. 1.5

Оберіть метод GET та вставте посилання. Рис. 1.6

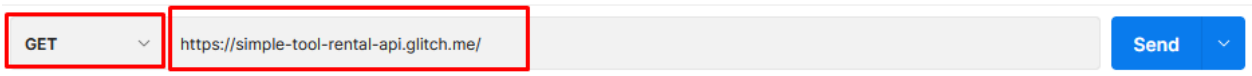

Рис. 1.6

Для того, щоб повернути статус API додайте **/status** до посилання на запит та натисніть «відправити»(Рис. 7)

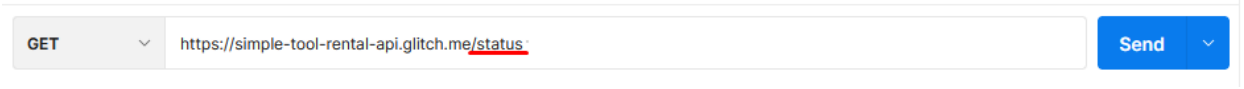

Рис. 1.7

За кілька секунд ви отримаєте відповідь. ‘UP’ означає, що API працює правильно та ми можемо працювати з цим API. (Рис 1.8)

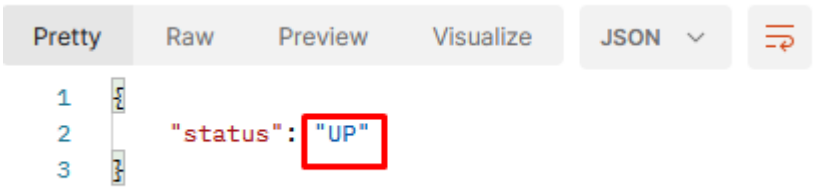

Рис. 1.8

Рис.5 Додання API посилання

1.4 Протокол HTTP

Розглянемо, що відбулося, коли ми отримали статус API. Postman відправив запит на сервер. Сервер отримав запит та відправив дані postman. Для того, аби така комунікація відбулась, ми використали протокол HTTP. HTTP – це протокол, який дає можливість клієнту(в даному випадку Postman) та серверу, який виконує API, комунікувати. HTTPS – це захищена та зашифрована версія HTTP, тому краще використовувати HTTPS. (Рис. 1.9)

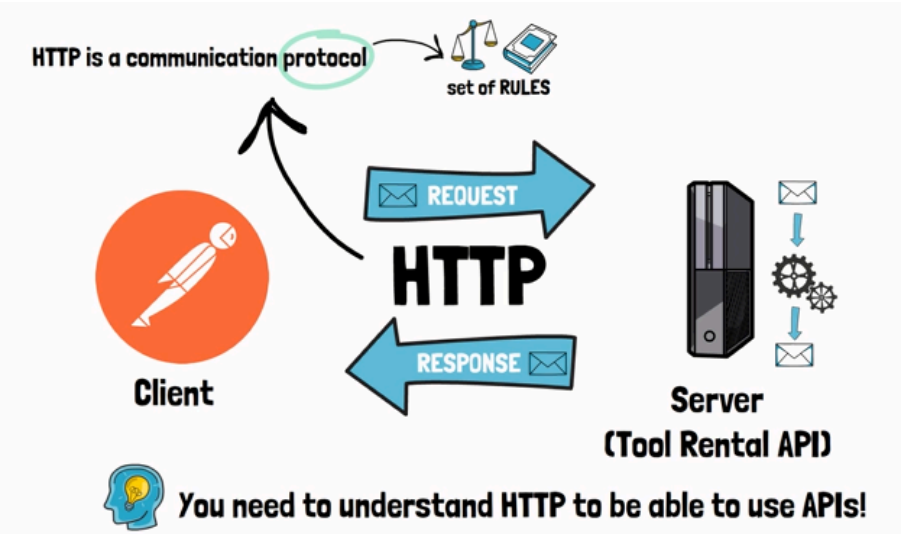

Рис. 1.9

HTTP запит містить:

- Метод запиту
- Адресу / URL
- Хедери
- Баді

HTTP відповідь містить:

- Статус коду
- Хедери
- Баді

Розділ 2. ФОРМУВАННЯ ЗАПИТІВ В POSTMAN

2.1 Колекції postman

Колекція – це як папка, яка зберігає всі запити, які зазвичай відносяться до API або якось пов’язані один з одним. Для того, щоб створити колекцію, треба зберегти API, а у вікні, яке відкриється вибрати «нова колекція» або «створити колекцію». (Рис. 2.1)

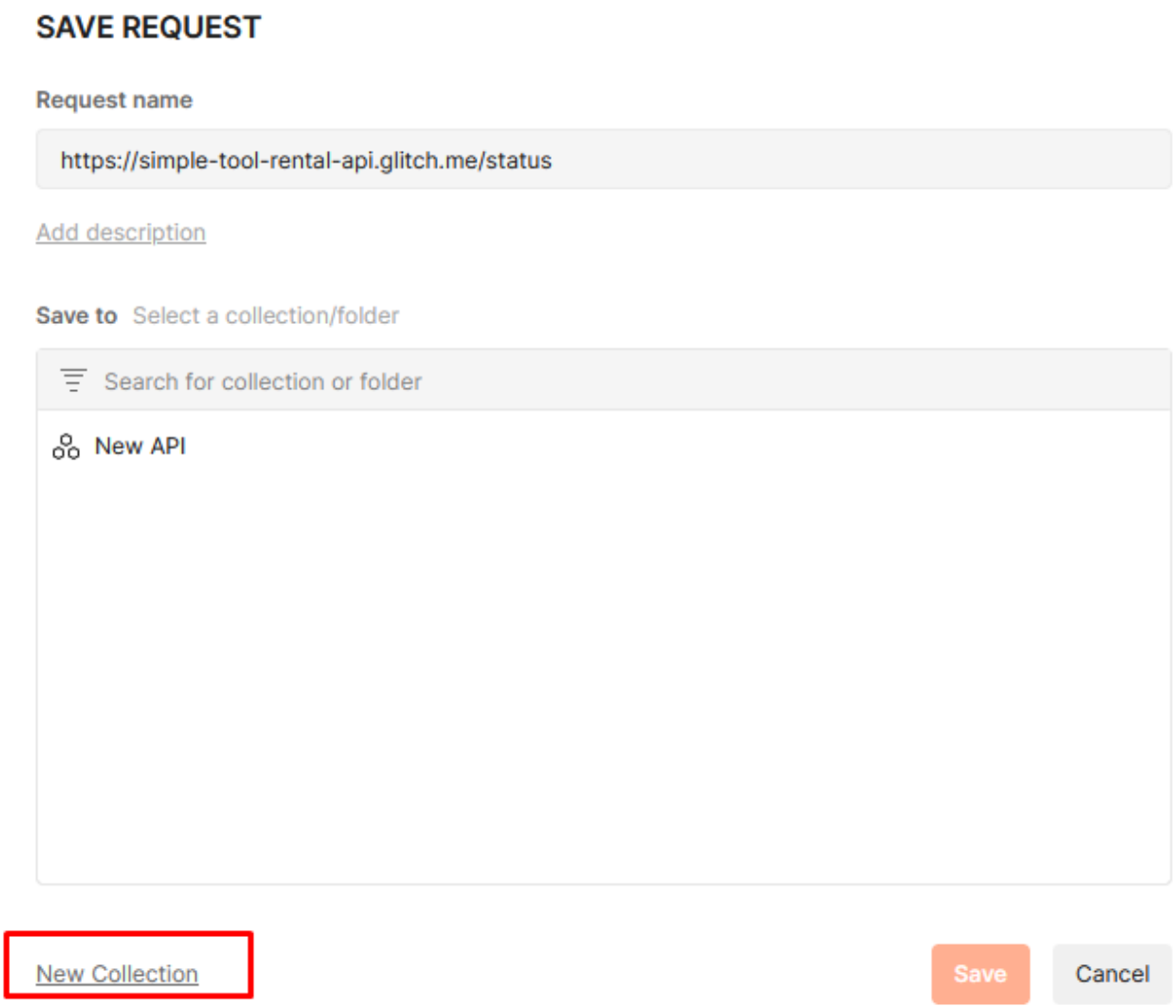

Рис. 2.1

Назвемо колекцію Tool Rental API та натиснемо «створити» та зберігаємо запит. (Рис. 2.2, Рис.2.3)

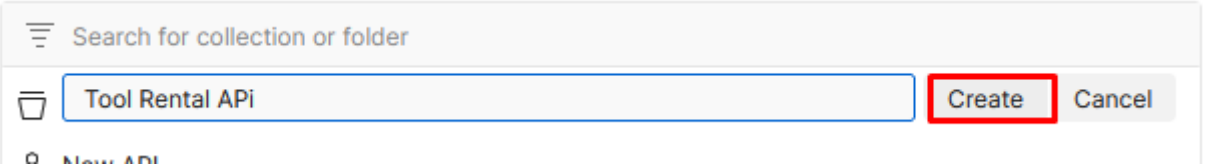

Рис. 2.2

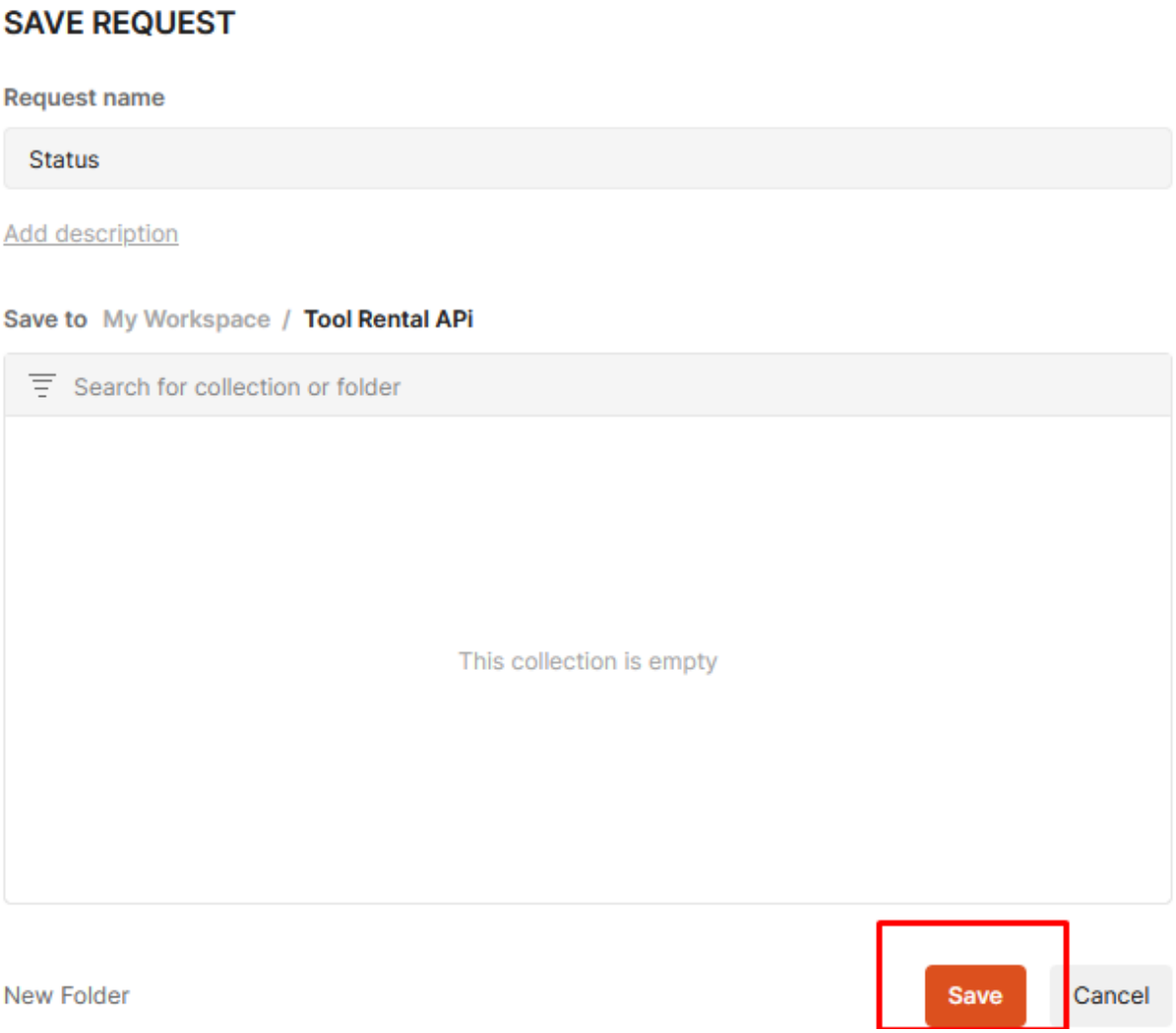

Рис. 2.3

Працюючи з запитами Postman, можна використати концепцію базової URL, що допоможе зробити запит конфігураційним. Для цього помістимо посилання в змінну та дамо змінній ім’я, а в графі «область видимості» обираємо нашу колекцію. Створити змінну можна лише, якщо запит збережено.(Рис. 2.4)

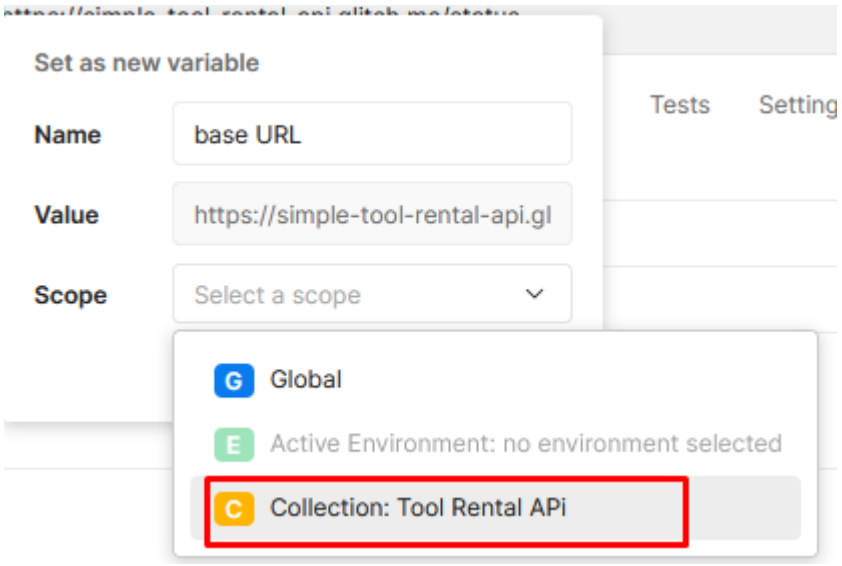

Рис. 2.4

Після цього адреса зміниться. Імя змінної буде знаходитися між двома фігурними дужками. Це синтакс для змінної в Postman . Запит буде працювати правильно. (Рис. 2.5)

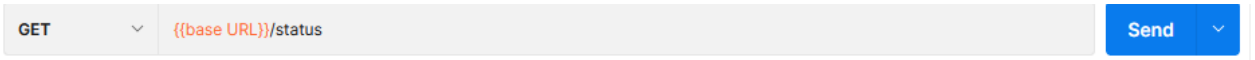

Рис. 2.5

Якщо перейти до вкладки «змінні», ми побачимо, що в кожної змінної є початкове та поточне значення. (Рис. 2.6)

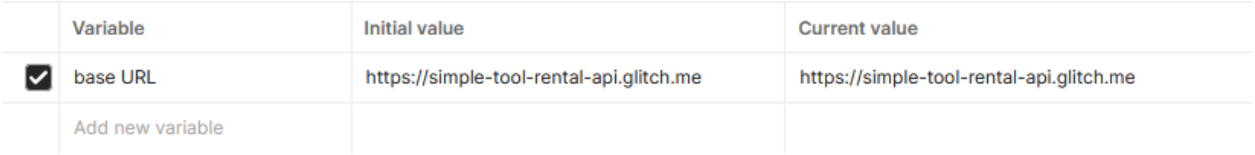

Рис. 2.6

Початкове значення:

- Використовується Postman при підтвердженні запиту
- Приватне для вас / вашого акаунту Postman
- Немає можливості поділитися з іншими

Поточне значення:

- Можна змінювати, коли ділитися колекцією
- Не використовується застосунком Postman

2.2 Параметри Query

Параметри Query – один із способів відправити дані в API. Вони можуть бути обовязковими або опціональними (згідно з документацією). Якщо змінна не

збережена, вона підсвічується червоним кольором та при наведенні видно, що змінна не вирішена. В такому випадку відправити дані неможливо.. (Рис. 2.7)

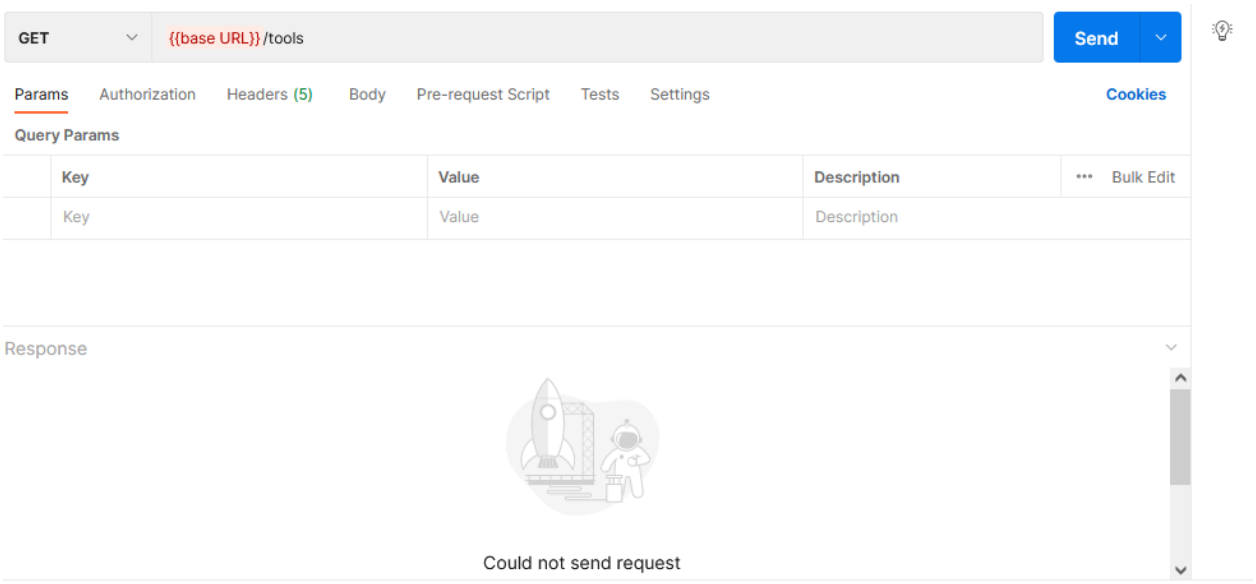

Рис. 2.7

Щоб вирішити проблему, достатньо зберегти змінну. Змініть назву та помістіть її в існуючу колекцію, або створіть нову. Дані прийдуть в форматі JSON. (Рис. 2.8)

Рис. 2.8

Щоб додати параметри Query, впишіть ключ та значення з JSON файлу, які Ви отримали у відповідному місці. (Рис. 2.9)

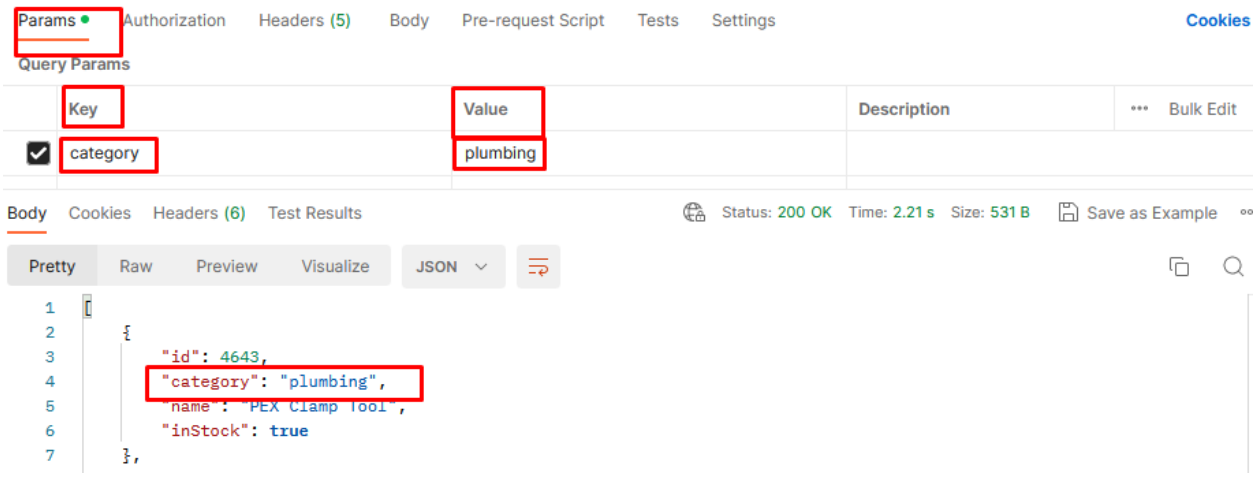

Рис. 2.9

Якщо значення вписати інше, ніж прийшло, буде помилка 400 Bad Request. (Рис. 2.10)

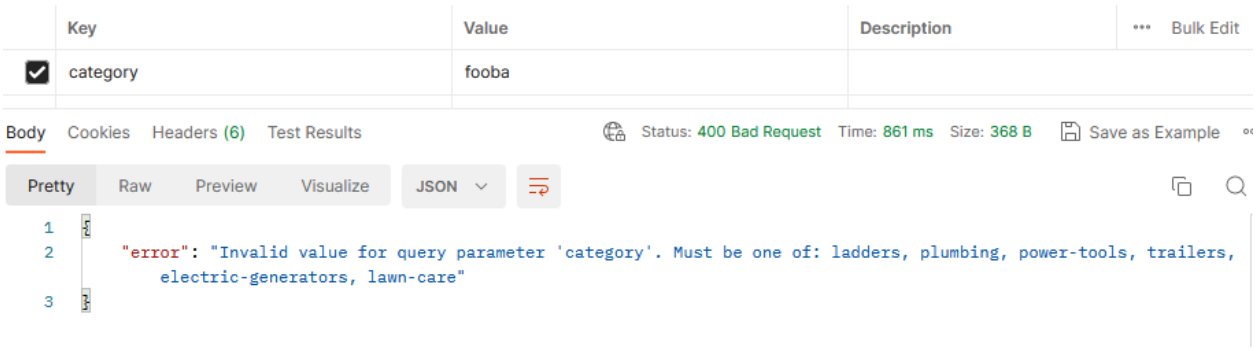

Рис. 2.10

2.3 Змінні Path

Змінні Path в Postman це всього на всього placeholder. На відміну від параметрів Query, змінні Path повертають 1 набір даних в форматі JSON. (Рис. 2.11)

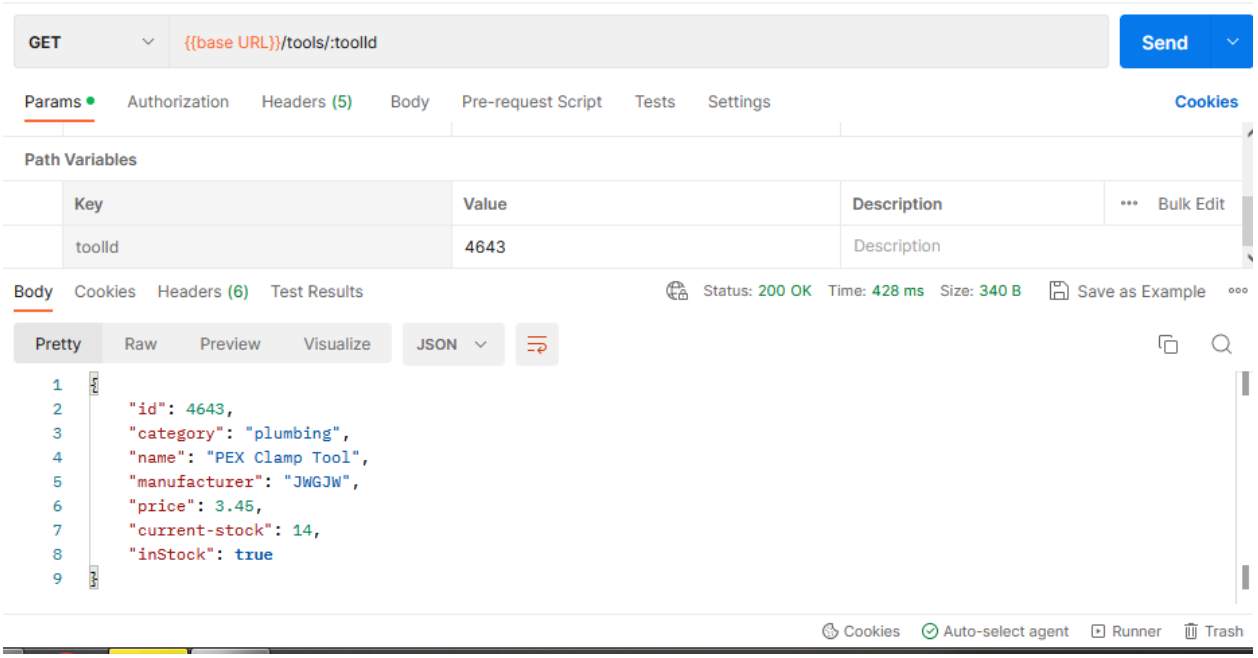

Рис. 2.11

Для відслідкування запитів та відповідей, можна використовувати консоль. Рис. (2.12)

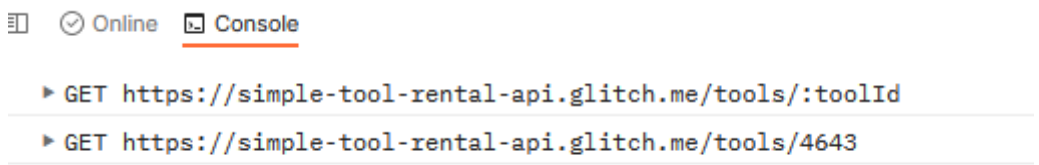

Рис. 2.12

2.4 Авторизація API

Деякі ендпоінти можуть вимагати авторизації. Для того, щоб підтвердити або подивитися ордери необхідно зареєструвати.Для цього продублюємо минулий запит, змінемо імя та метод на POST. А до змінної додамо **/api-clients.** Перейдемо в боді, оберемо raw та формат JSON. Отримаємо accessToken **(Рис. 2.13)**

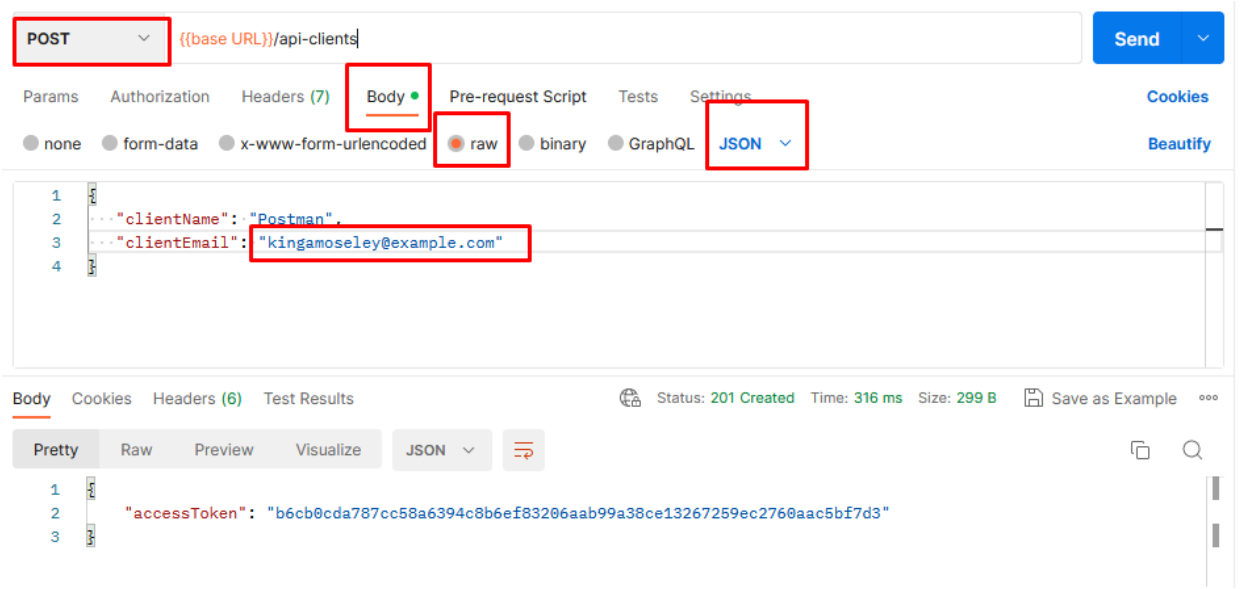

Рис. 2.13

Переходимо у вкладку авторизації, та обираємо токен та тип (Рис. 2.14)

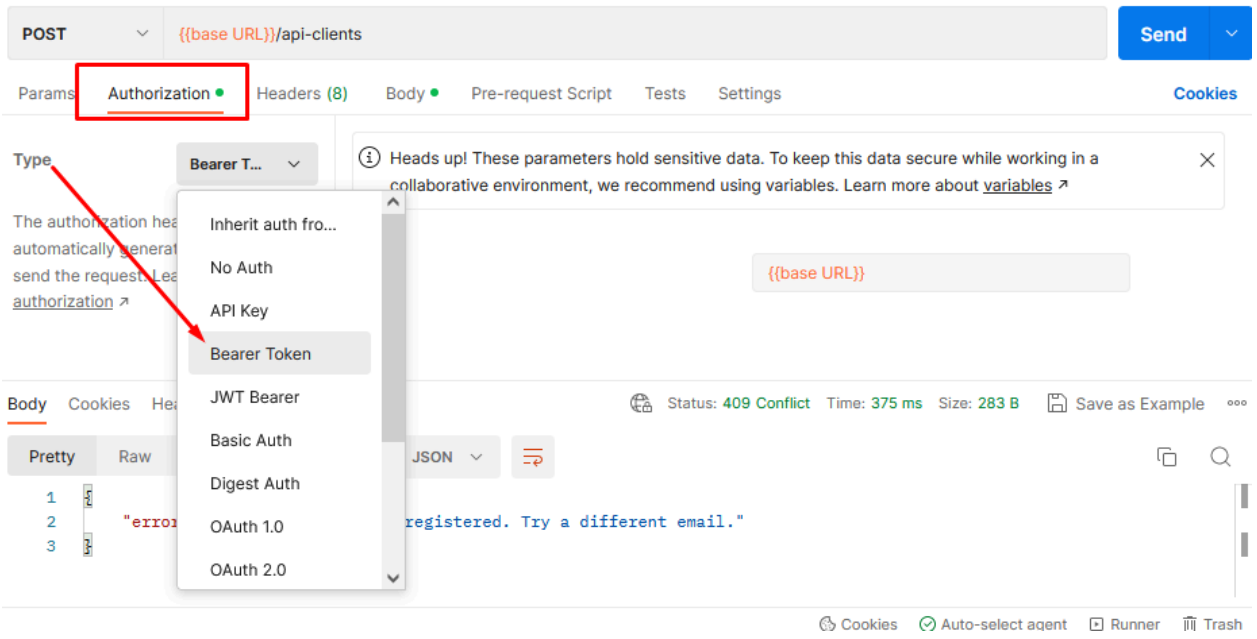

Рис. 2.14

2.5 Проблема статус-кодів HTTP

Типічні помилки:

404 – перевірте URL або метод запитів HTTP

400 – перевірте request body та переконайтеся що правильний формат JSON

409 – клієнт зареєстрований

2.6 HTTP заголовки

HTTP заголовки можна знайти в запиті та у відповіді.

Для перегляду заголовків перейдіть в розділ «заголовки» та натисніть на «скриті». Відкриється перелік. (Рис. 2.15)

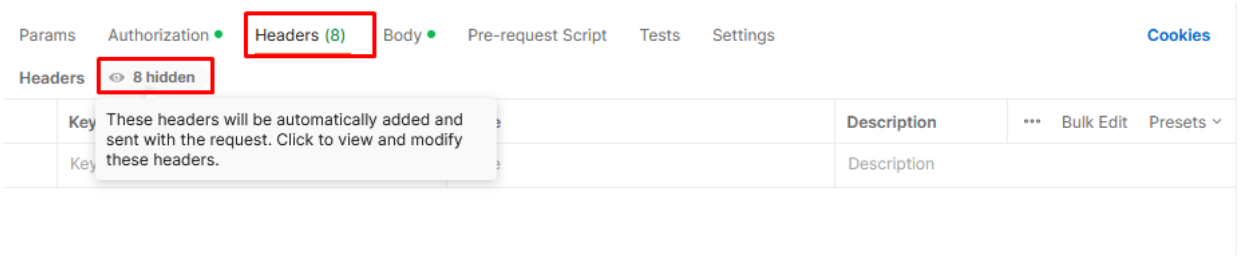

Рис. 2.15

До типових заголовків відносять:

- Content Type – передає API інформацію про те, що request body в форматі JSON.
- Authorization – містить інформацію про автентифікацію.

До типічних хедерів відповіді відносять:

- Content Type – говорить клієнту (в даному випадку програмі Postman) що request body в форматі JSON.

2.7 JSON формат

Ми використовуємо JSON для передачі даних. (Рис. 2.16)

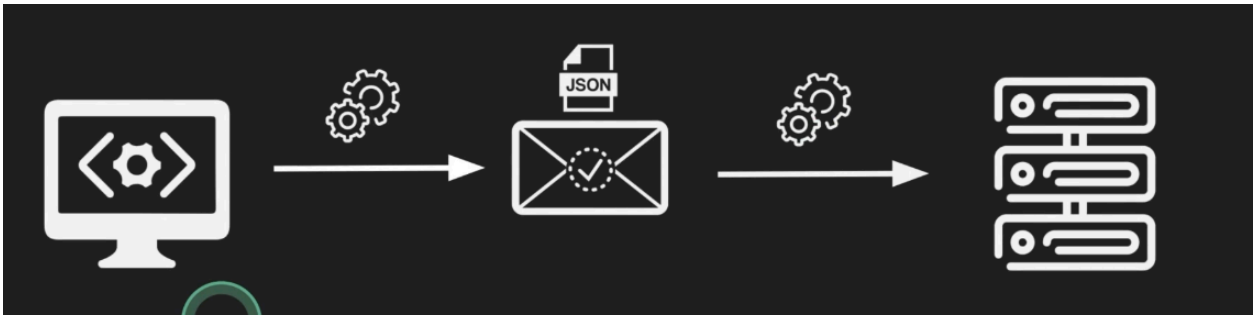

Рис. 2.16

JSON представлений у форматі «ключ-значення». Ключі та значення пишуться в середині фігурних скобок в лапках Значення в цифрах та булінах пишуться без лапок. (Рис.2.17)

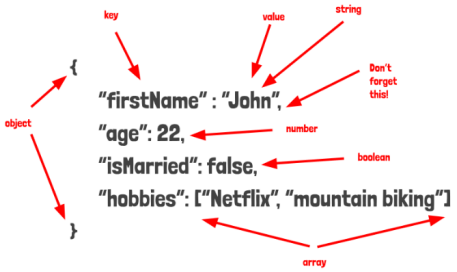

Рис.2.17

Якщо формат буде написаний неправильно, буде помилка, як на Рис. 2.18

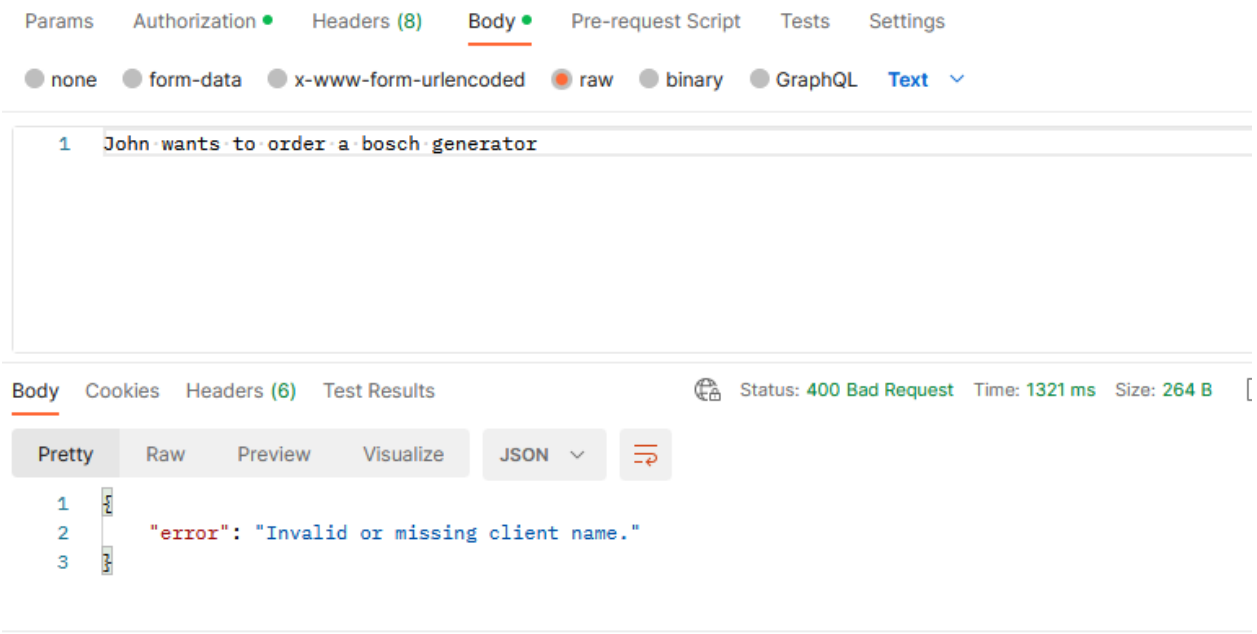

Рис. 2.18

Типічні помилки формату JSON включають наступні:

- Відсутні лапки для ключів та значень
- Відсутні коми після рядків
- Кома після останнього рядка

2.8 GET vs POST

Метод GET:

- Лише отримує дані
- Можна викликати кілька раз без побічних даних, тобто нічого не зміниться.
- Немає request body
- Можна додавати параметри в хедерах та URL

Метод POST:

- Створює нові дані

- При кожному виклику даного методу, будуть створюватися нові дані.
- Має request body
- Можна додавати параметри в баді, хедерах та URL

2.9 В яких випадках Postman не використовується

- Немає взаємодії з користувачем, такої як заповнення форм або кліки на кнопки.
- Краще не використовувати при тестуванні продуктивності, або будь-яких інших видах тестування в яких Ви здійснюєте багато запитів за короткий час.

2.10 Метод запитів PATCH

Метод запитів PATCH дозволяє змінювати існуючі дані.

2.11 Метод запитів DELETE

Метод запитів DELETE використовується для видалення даних. Зазвичай, request body не потрібний. Для того, щоб перевірити чи видалення пройшло успішно, використовуйте запит GET.

Розділ 3. ПІДГОТОВКА ДО АВТОМАТИЗАЦІЇ

3.1 Основи автоматизації.

Тестувати API вручну – великий обсяг роботи. Коли хтось вносить зміни до API нам треба вручну протестувати всі ендпоінти та параметри та перевірити чи API працює як і раніше.

Postman дозволяє нам тестувати API за допомогою написання API тестів. Автоматизація це процес за якого Postman виконує тестування, а тестувальник вмішується лише якщо щось іде не так.

3.2 Перший API тест

Postman має вбудовану функцію, яка дозволяє писати тести.

Зазвичай ми пишемо тести щоб перевірити чи буде відповідь така, як ми очікуєм, чи ні. Для написання тестів Postman використовує JavaScript. Напишемо базовий тест для перевірки статусу коду. Для цього перейдемо у вкладку «тести» та справа під сніпетами знайдемо «статус коду: код 200». Після вибору в нас зявиться скрипт для тесту. Натискаємо «відправити» та перевіряємо тест на успішність у вкладці «результати тестів». (Рис.3.1)

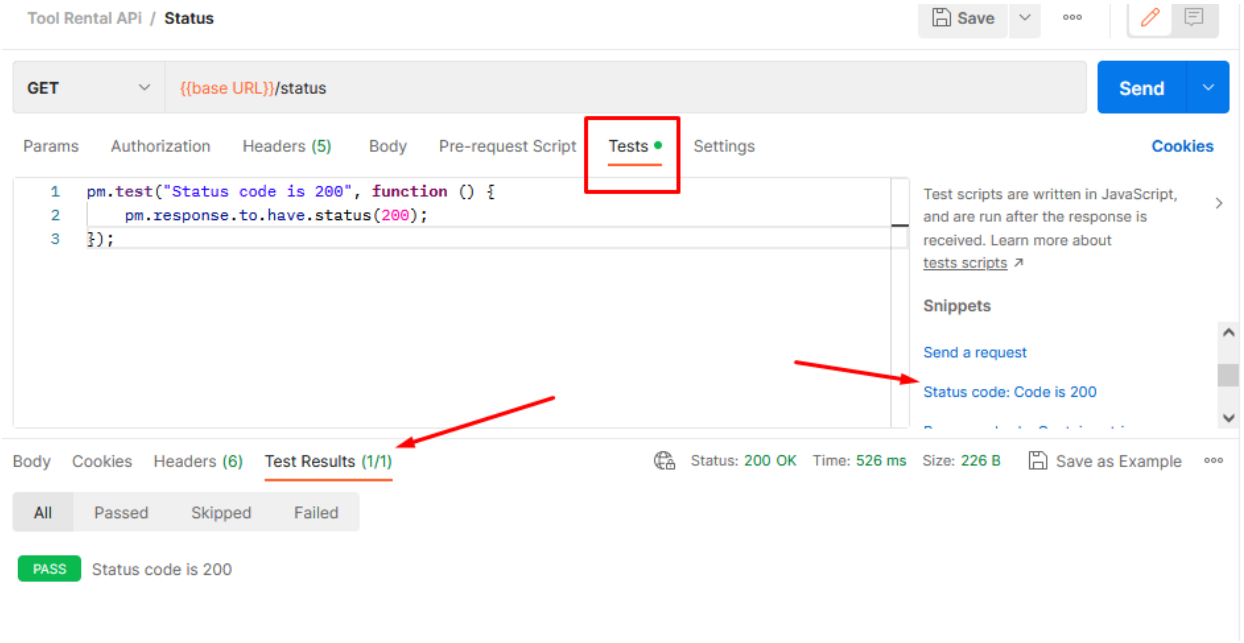

Рис.3.1

В даному скрипті ми зазначили, що очікуємо статус 200. Якщо ж в очікуваному результаті ми вкажемо інший статус – висвітиться помилка (Рис. 3.2), в якій говориться, що в результаті отримали статус 200, а очікували 400.

Рис. 3.2

Якщо в посиланні допустити помилку, тести не пройдуть. (Рис. 3.3). Нам буде повідомлено, що замість статусу 200, ми отримали статус 404.

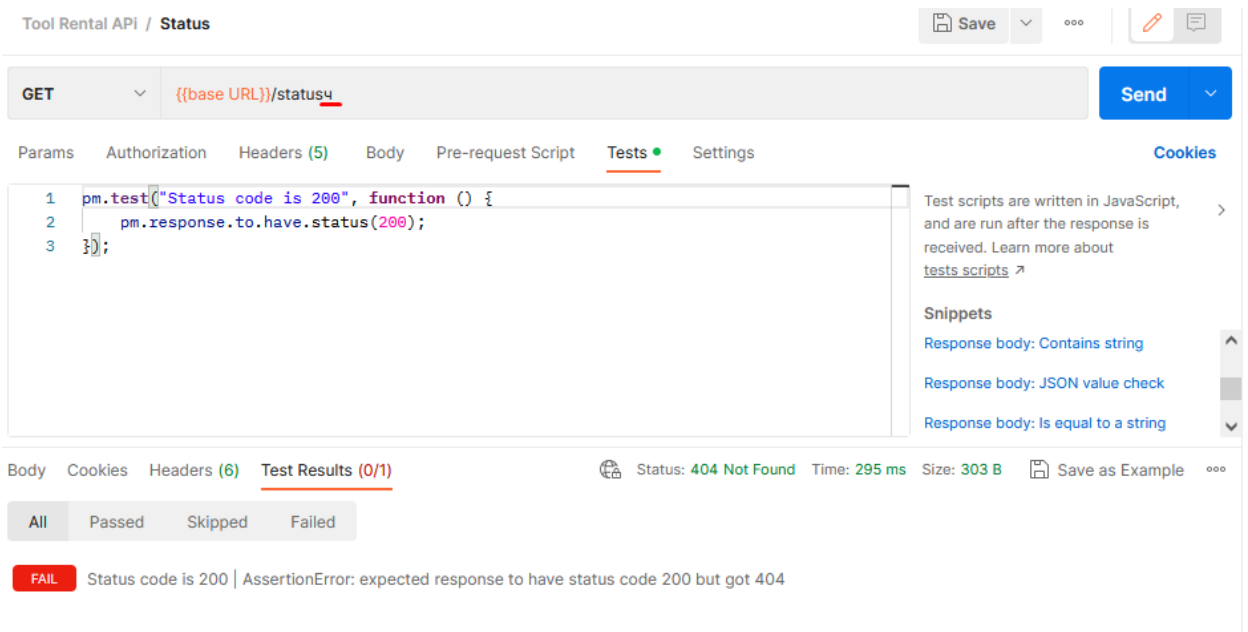

Рис. 3.3

А що, як в нас буде статус “down” замість ‘up’ в body? Напишемо ще один тест, щоб ви краще розуміли структуру. Ми завжди починаємо писати тест з функції pm.test(), яка приймає 2 параметри. Перший параметр – імя тесту, другий – функція колбек. В середині фігурних скобок пишемо твердження.(Рис. 3.4)

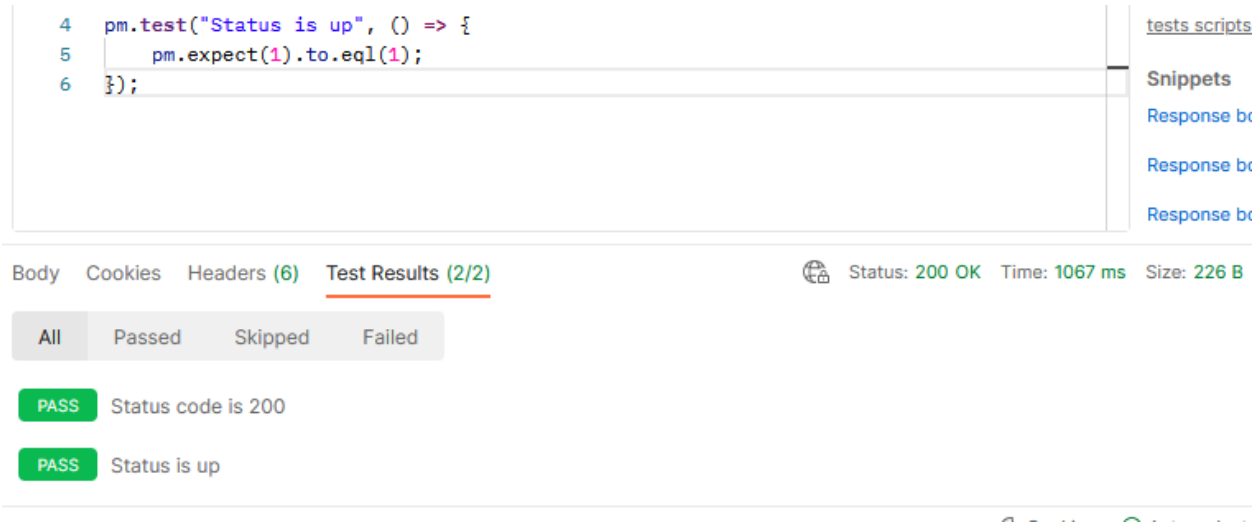

Рис. 3.4

Будьте обережними при написанні тесту, адже якщо допустити синтаксичну помилку (наприклад,закрити функцію раніше), тести будуть завжди проходити.(Рис. 3.5)

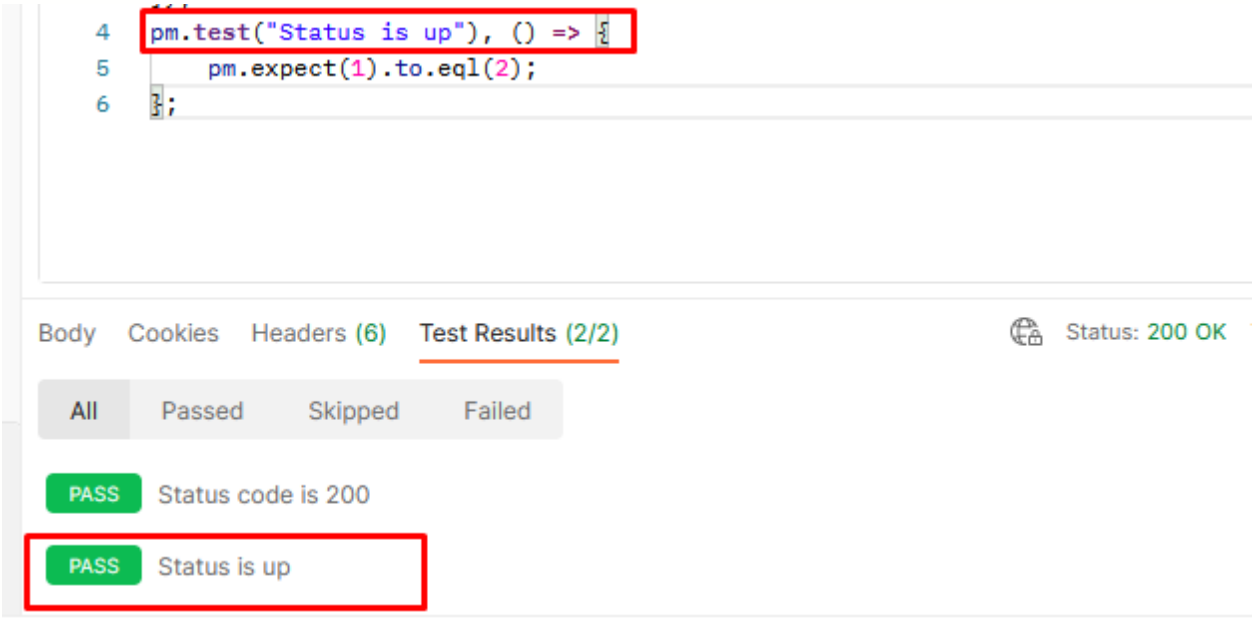

Рис. 3.5

3.3 Змінні postman

Для того, що зберегти час та уникнути копіювання та вставлення змінних з одного запиту в інший, Postman дозволяє створювати різні типи змінних:

1. Змінні колекції – доступні лише для колекції.(Рис. 3.6)

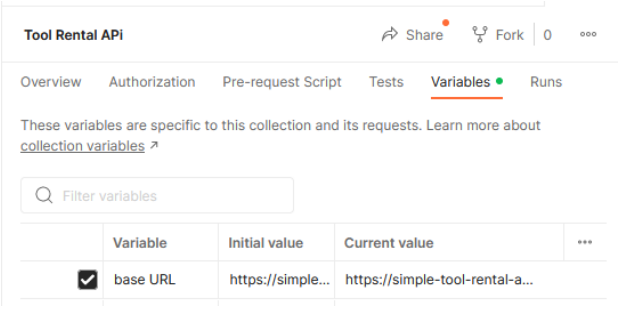

Рис. 3.6

2. Змінні середовища:
    - Доступні лише для середовища
    - Корисні, коли ви хочете перевикористати одну й ту ж саму колекцію для різних серверів, таких як localhost, тестування та продакшн.
3. Глобальні змінні – доступні для всього робочого середовища.(Рис. 3.7)

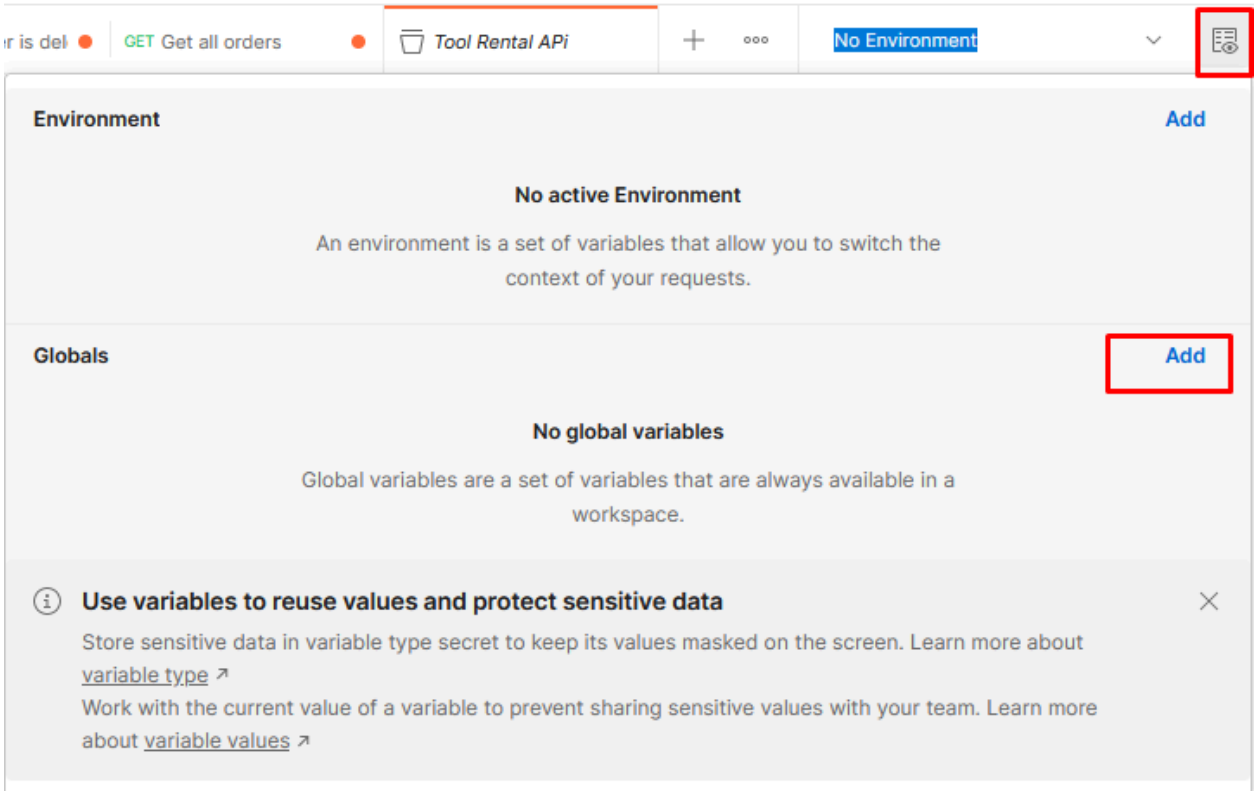

Рис. 3.7

Глобальні змінні дуже зручні у використанні(Рис. 3.8)

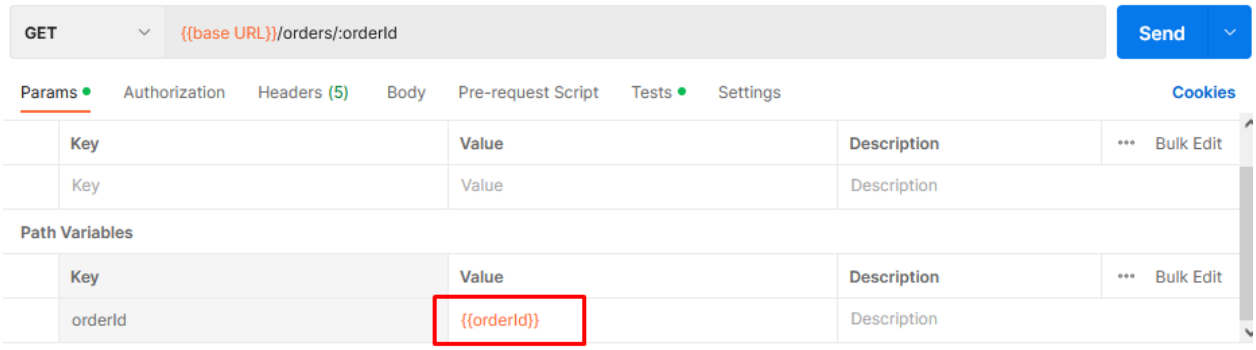

Рис. 3.8

3.4 Робота зі змінними Postman зі скриптів

За допомогою скриптів можна створити нову або знайти існуючу змінну. Для цього використайте наступний скрипт: pm.collectionVariables.get(“apiToken”). Якщо ввести змінну, якої не існує і подивитися в консоль, виведе undefined(Рис. 3.9)

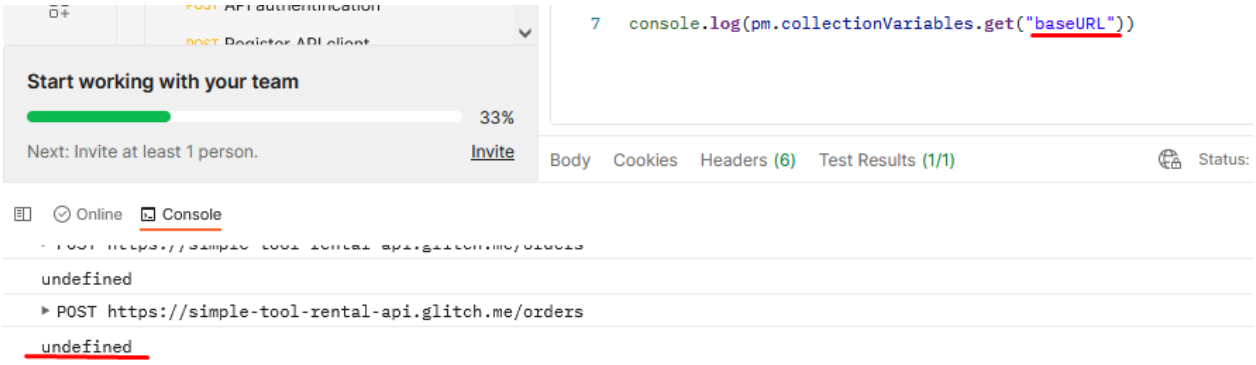

Рис. 3.9

Для створення нової змінної колекції, використовуйте наступний синтаксис:

pm.collectionVariables.set(“firstName”, “John”). Крім того, за допомогою цього синтаксису можна змінити значення у змінної (Рис. 3.10)

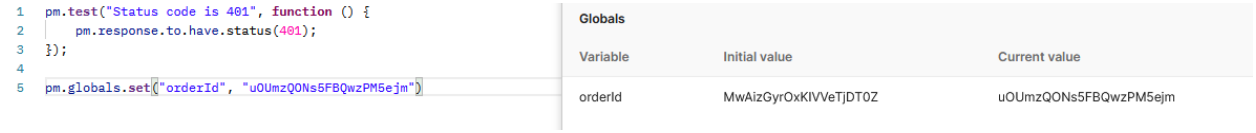

Рис. 3.10

Розділ 4. ЗАПУСК АВТОМАТИЗОВАНОГО ЗБОРУ

4.1 Collection Runner

Collection Runner – інструмент postman, який дозволяє запускати всю колекцію одним кліком, замість того, щоб проходити по кожному запиту. Є два шляхи, щоб здійснити запуск колекції:

Натиснути на «runner» внизу сторінки або за допомогою налаштувань колекції. (Рис. 4.1)

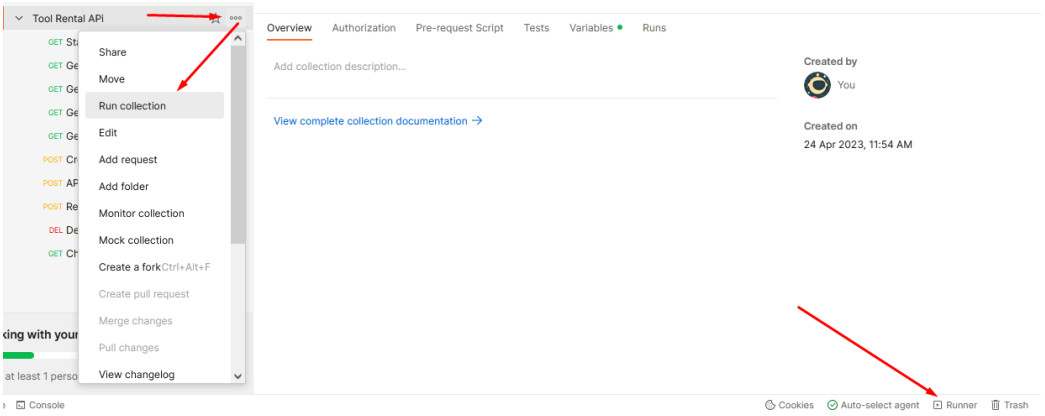

Рис. 4.1

Для того, щоб запити почали виконуватися, оберіть «запустити» та назву колекції(Рис. 4.2)

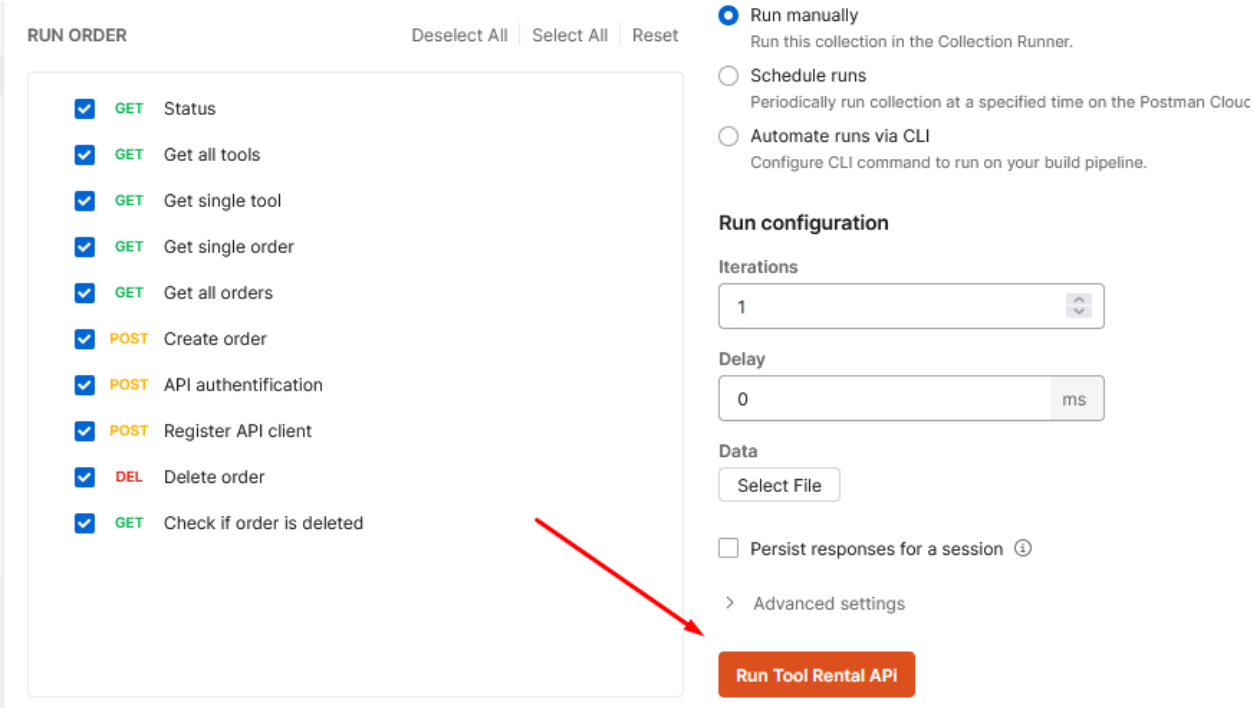

Рис. 4.2

Після виконання ви зможете побачити всю інформацію з виконання запитів, включаючи тести, які пройшли, та тести, які впали. (Рис. 4.3)

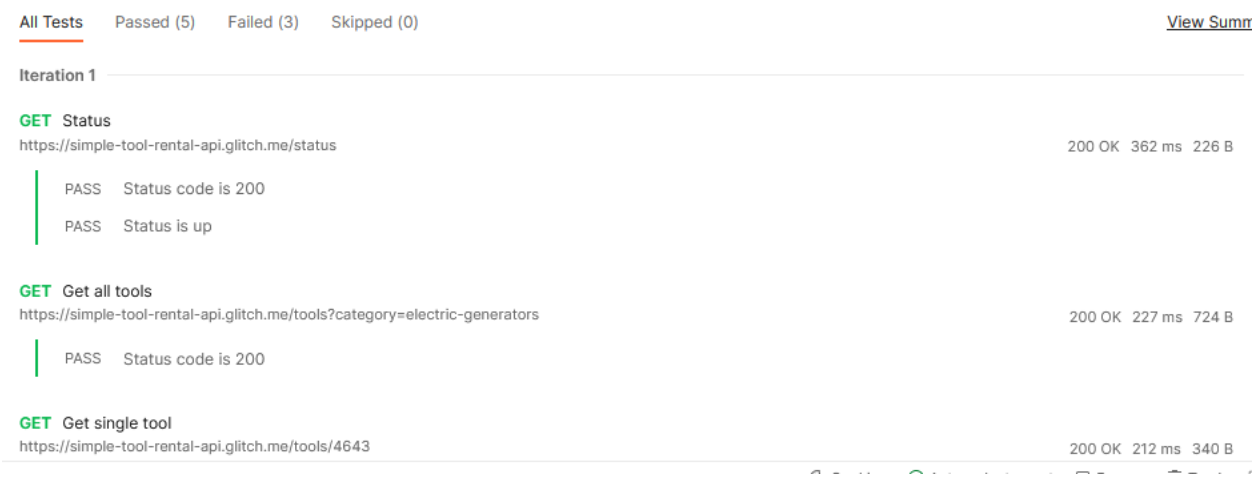

Рис. 4.3

4.2 Моніториг Postman.

Автоматизувати запуск колекції можна за допомогою моніторингу postman. Для створення перейдіть до своєї колекції(Рис. 4.4)

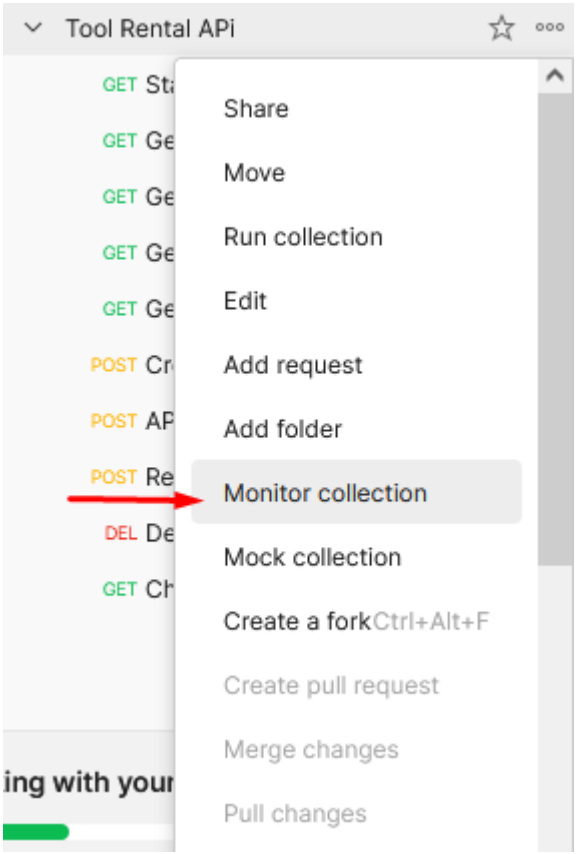

Рис. 4.4

Далі вписуєте імя, а дефолтні налаштування можна залишити без змін. Та натискаєте «створити моніторинг» (Рис. 4.5)

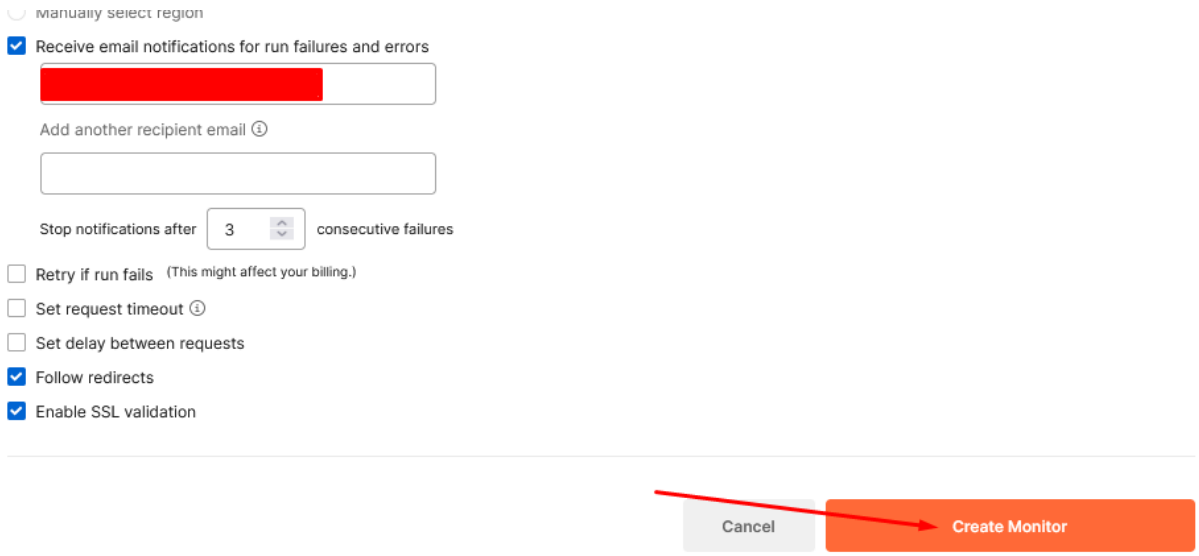

Рис. 4.5

Після виконання, якщо ви бачите “unhealthy”, значить десь є помилка. Відстежити її можна за допомогою інформації. (Рис. 4.6)

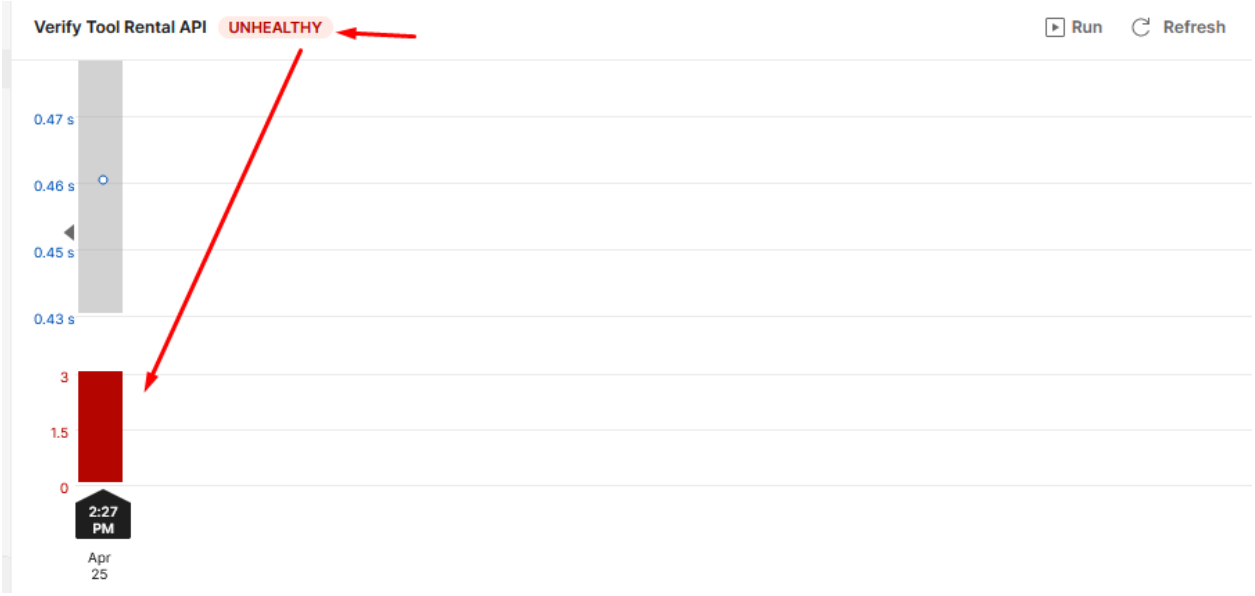

Рис.4.6

Нижче можна побачити кількість тестів, які пройшли, та тих – які впали(Рис. 4.7)

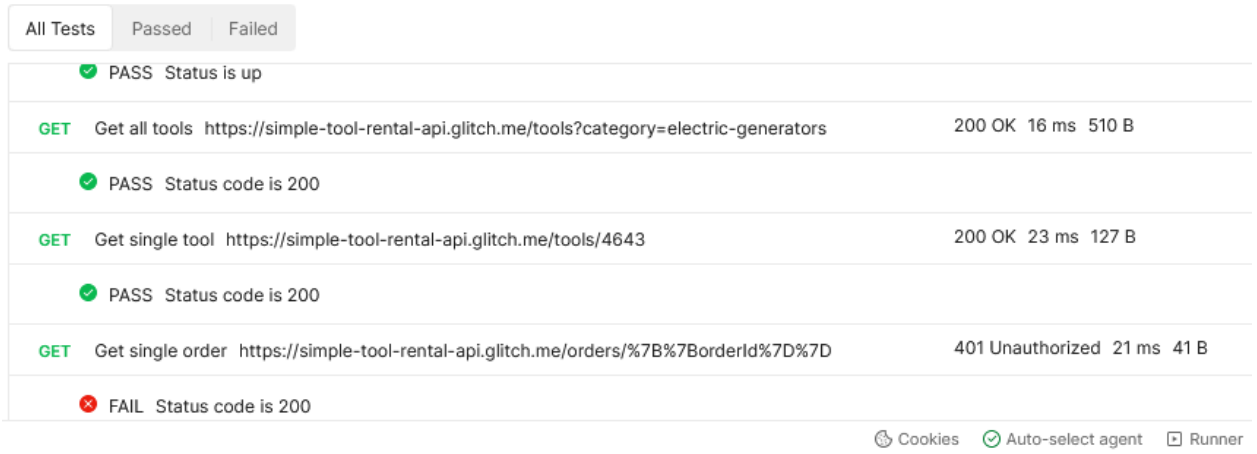

Рис. 4.7

4.3 Newman – інструмент CLI в Postman

Newman інструмент CLI в Postman, за допомогою якого можна запускати колекцію postman з терміналу. Для того, щоб використовувати цей інструмент має бути встановлений Node.js. Встановити newman можна вводячи в терміналі наступну команду: npm install -g newman

Запустити newman можна за допомогою команди newman run + посилання на Вашу колекцію. (Рис. 4.8)

Рис. 4.8

Важливо! Якщо ви запускаєте за допомогою посилання на колекцію, при зміні в запитах вам треба буде знову брати посилання та вставляти його в newman.

Після запуску буде показано результати виконання запитів, а також можна буде подивитися кількість успішних тестів та кількість тестів, що впали (Рис. 4.9)

	executed	failed
iterations	1	0
requests	9	0
test-scripts	18	0
prerequest-scripts	9	0
assertions	12	0
total run duration: 2.7s		
total data received: 3.27kB (approx)		
average response time: 277ms [min: 131ms, max: 1155ms, s.d.: 311ms]		

Рис. 4.9

БІБЛІОГРАФІЯ:

1. https://simple-tool-rental-api.glitch.me/
2. https://github.com/vdespa/quick-introduction-to-postman/blob/main/simple-tool-rental-api.md#Get-all-tools
3. Despa V. Quick Introduction to Postman and API Testing for Beginners [Електронний ресурс] / Valentin Despa. – 2023. – Режим доступу до ресурсу: file:///C:/Users/Professional/Downloads/Introduction+to+Postman+and+API+v2023-1+(course+notes)-1.pdf.
4. https://www.postman.com/downloads/
5. https://www.postman.com/release-notes/
6. https://github.com/postmanlabs/newman