Запуск коллекции из командной строки:

Запускать коллекции Postman из командной строки можно с помощью инструмента под названием Newman. Newman — это CLI-инструмент для Postman, который позволяет выполнять запросы и тесты коллекций Postman.

- 1. Скачать Node.js (скачать можно с официального сайта https://nodejs.org/)
- 2. После установки Node.js нужно установить Newman с помощью npm (менеджер пакетов Node.js). Нужно ввести следующую команду в терминале (эта команда установит Newman глобально на вашей системе):

npm install -g newman

3. Экспорт коллекции Postman:

Перед тем как запустить коллекцию, нужно ее экспортировать из Postman:

- Открыть Postman.
- Найти коллекцию, которую хотим экспортировать, в левой панели.
- Щелкнуть правой кнопкой мыши по коллекции и выбрать "Export".
- Выбрать нужный формат (обычно JSON) и сохранить файл на вашем компьютере.
- 4. Запуск коллекции с помощью Newman.

Выполнить следующую команду в терминале, заменив your-collection.json на путь к экспортированному файлу:

newman run your-collection.json

5. Проверка результатов.

После завершения выполнения команды в терминале мы увидим результаты запуска коллекции, включая информацию о пройденных и непройденных тестах.

Информация о запуске коллекции:

Your environment has been set up for using Node.js 20.17.0 (x64) and npm.

C:\Users\Юлия Самусева>npm install -g newman npm warn deprecated har-validator@5.1.5: this library is no longer supported npm warn deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

changed 120 packages in 13s

9 packages are looking for funding run `npm fund` for details

C:\Users\Юлия Самусева>

C:\Users\Юлия Caмусева>newman run "C:\Users\Юлия Caмусева\Downloads\Petstore Final Project.postman_collection (1).json" newman

Petstore Final Project

→ User (Operations about user)

POST https://petstore.swagger.io/v2/user/createWithList [200 OK, 370B, 731ms]

- √ Response status code is 200
- $\sqrt{}$ Response has the required fields code, type, and message
- √ Type is a non-empty string
- √ Message is a non-empty string
- √ Content-Type header is application/json

→ Get user by user name

GET https://petstore.swagger.io/v2/user/string8888 [200 OK, 486B, 142ms]

- √ Response status code is 200
- $\sqrt{}$ Response has the required fields
- 1. Email is in a valid format
- 2. Phone number is in a valid format
- $\sqrt{}$ UserStatus is a non-negative integer

→ Update user

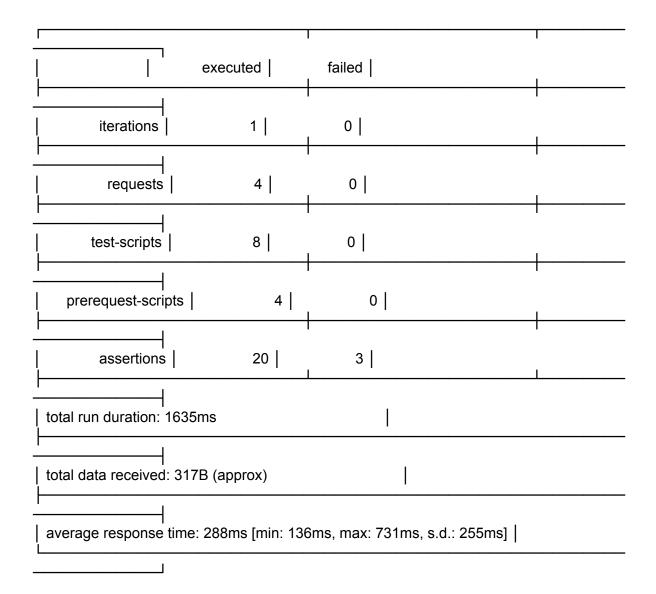
PUT https://petstore.swagger.io/v2/user/string8888 [200 OK, 387B, 146ms]

- √ Response status code is 200
- √ Content-Type is application/json
- $\sqrt{}$ Validate the structure of the response object
- √ Code is a non-negative integer
- $\sqrt{}$ Type and message are non-empty strings

→ New Request

DELETE https://petstore.swagger.io/v2/user/string8888 [200 OK, 378B, 136ms]

- √ Response status code is 200
- √ Content-Type header is application/json
- 3. Code field exists and is not empty
- $\sqrt{}$ Type field in the response should exist and be a string
- $\sqrt{}$ Message field in the response should exist

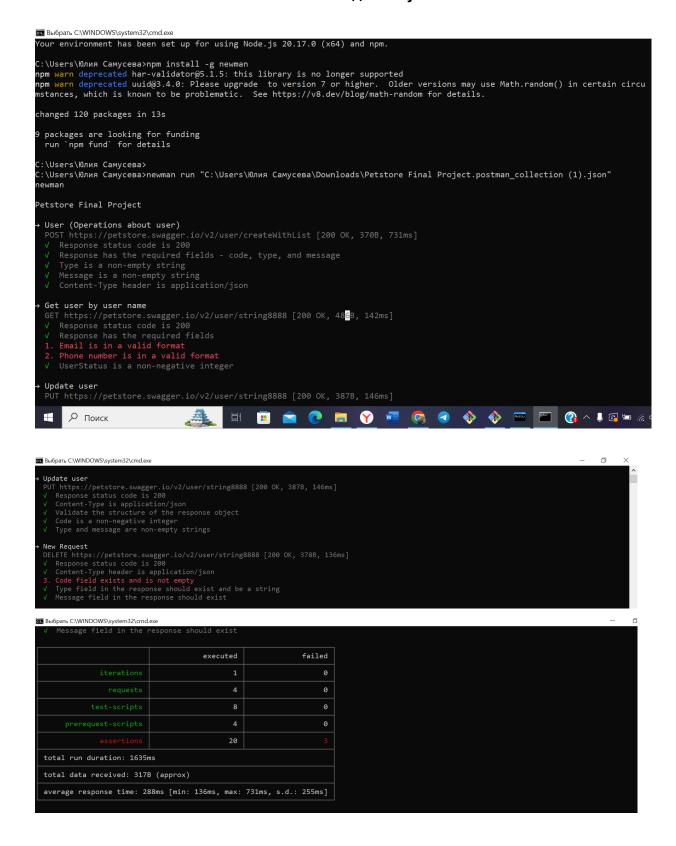


failure detail

1. AssertionError Email is in a valid format expected 'string' to match /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$/ at assertion:2 in test-script inside "Get user by user name"

- 2. AssertionError Phone number is in a valid format expected 'string' to match /^\d{3}-\d{3}-\d{4}\$/ at assertion:3 in test-script inside "Get user by user name"
- 3. AssertionError Code field exists and is not empty
 .empty was passed non-string primitive 200
 at assertion:2 in test-script
 inside "New Request"

Как это выглядит визуально:



👞 Выбрать C:\WINDOWS\system32\cmd.exe

| # | | detail |
|-----|-----------------------|---|
| 1. | AssertionError | Email is in a valid format expected 'string' to match /^[a-zA-Z0-9%+-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,}\$/ at assertion:2 in test-script inside "Get user by user name" |
| 2. | AssertionError | Phone number is in a valid format expected 'string' to match /^\d{3}-\d{3}-\d{4}\$/ at assertion:3 in test-script inside "Get user by user name" |
| 3. | AssertionError | Code field exists and is not empty .empty was passed non-string primitive 200 at assertion:2 in test-script inside "New Request" |
| C:\ | Users\Юлия Самусева>_ | |