

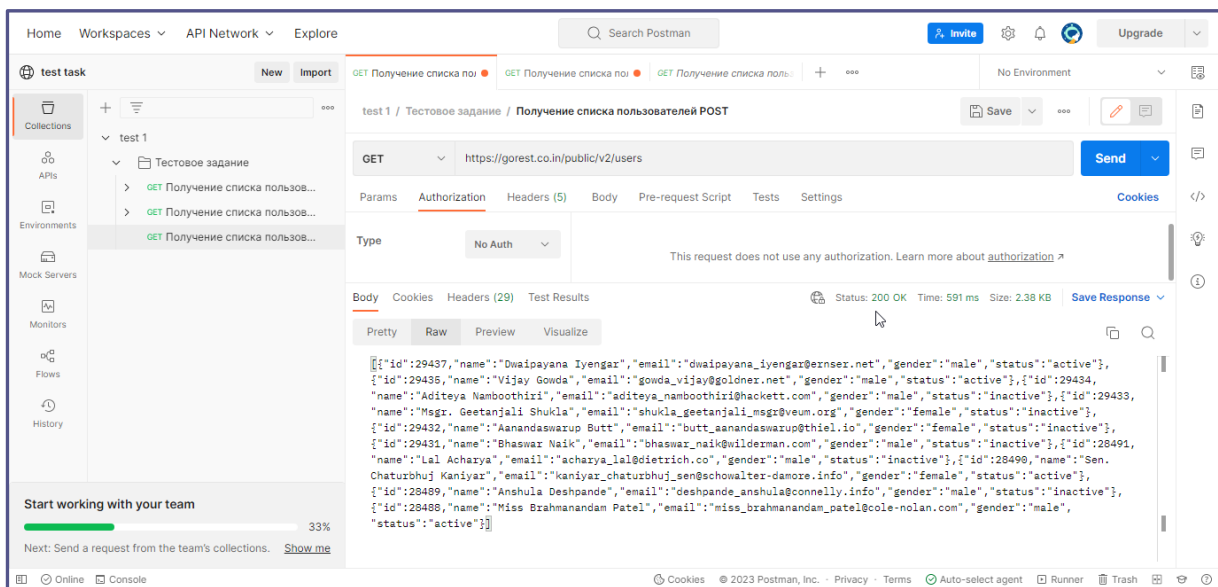
## Пример коллекции Postman

Перейти по ссылке для просмотра коллекции:

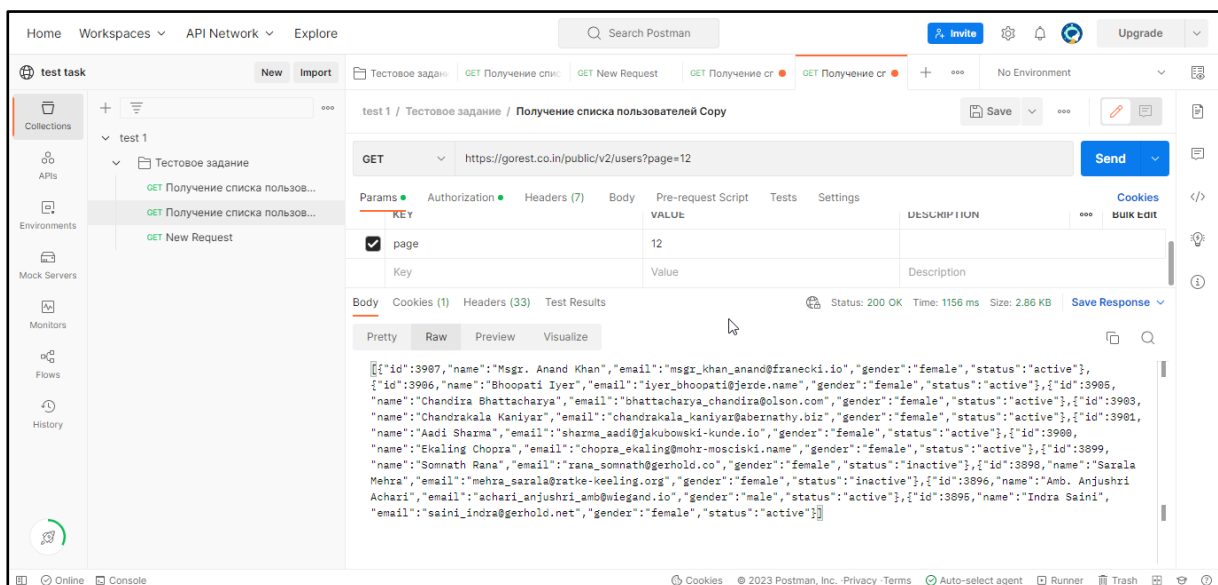
<https://www.postman.com/kosmonaft80/workspace/test-task/collection/25290198-0d43cc1a-faf2-413b-bad3-8c481778a9f1?action=share&creator=25290198>

В данной коллекции использованы переменные для окружения « тестовая коллекция 1 », получение id из ответа при создании пользователя с присвоением ее в переменную, переменные окружения, автотесты по статус кодам и времени на запрос, создание, просмотр, изменение и удаление данных о пользователе.

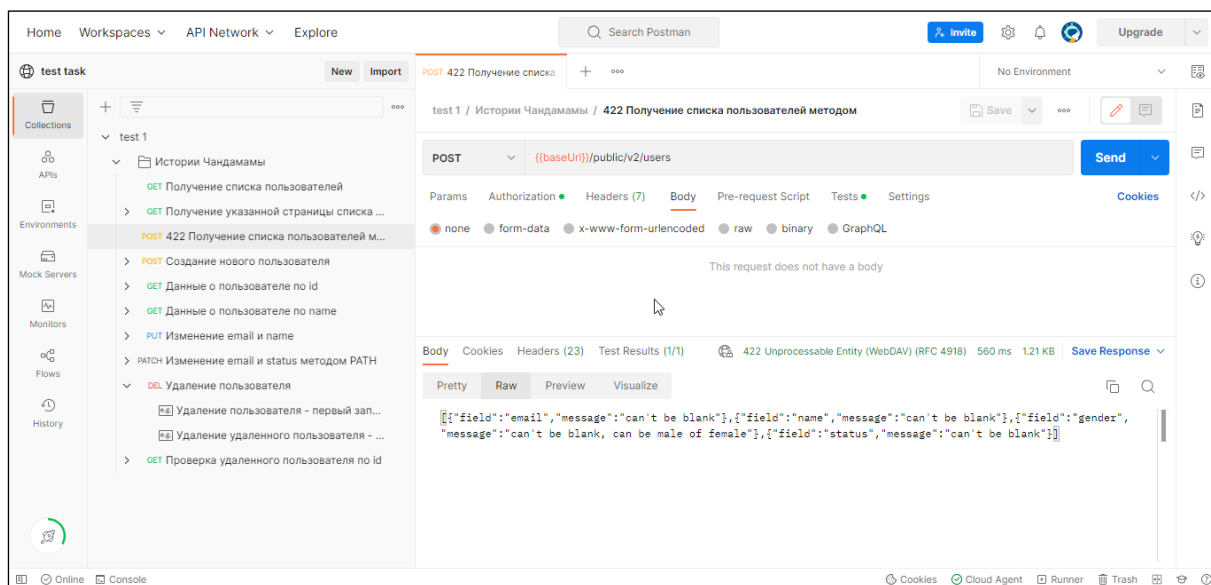
Запросом методом GET с вводом в вкладке Authorization токена, полученного при регистрации на сайте <https://gorest.co.in>, статус код ответа - "200 OK", в теле ответа список определенных пользователей (из страницы 1):



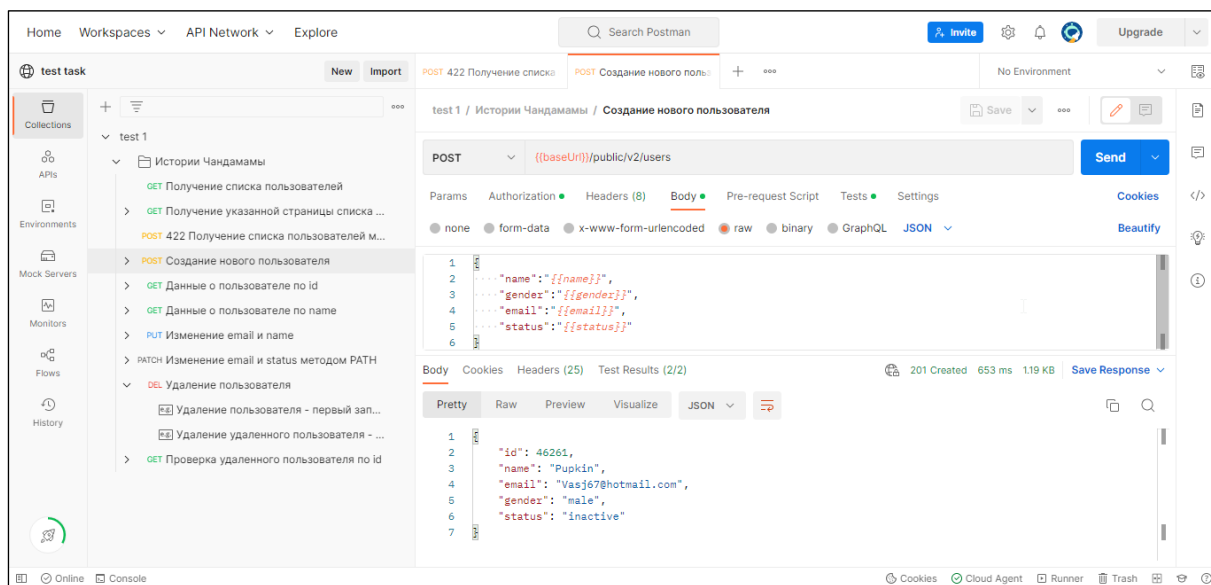
Другой список пользователей из страницы 12:



В ответ на запрос - пришел ответ с статус кодом "422 Unprocessable Entity" - Ошибка проверки данных (в ответ на запрос POST). Проверив тело ответа, видим сообщение об ошибке, потому что поля не могут быть пустыми:



Запрос ниже на рисунке - пришел ответ со статус кодом "201 Created" – пользователь создан, с данными пользователя в теле ответа:



На запросе ниже - данные по id :

The screenshot shows the Postman interface with a test collection named 'test task'. The selected request is a GET request titled 'test 1 / Истории Чандаммы / Данные о пользователе по id'. The URL is `{{baseUri}}/public/v2/users?id={{id}}`. The 'Query Params' section shows a parameter 'id' with the value `{{id}}`. The 'Body' tab is selected, showing a JSON response in 'Pretty' format:

```
1 {
2   "id": 46261,
3   "name": "Pupkin",
4   "email": "Vasj67@hotmail.com",
5   "gender": "male",
6   "status": "inactive"
7 }
8
```

The status bar at the bottom indicates a 200 OK response with a 578 ms duration and 1.34 KB of data.

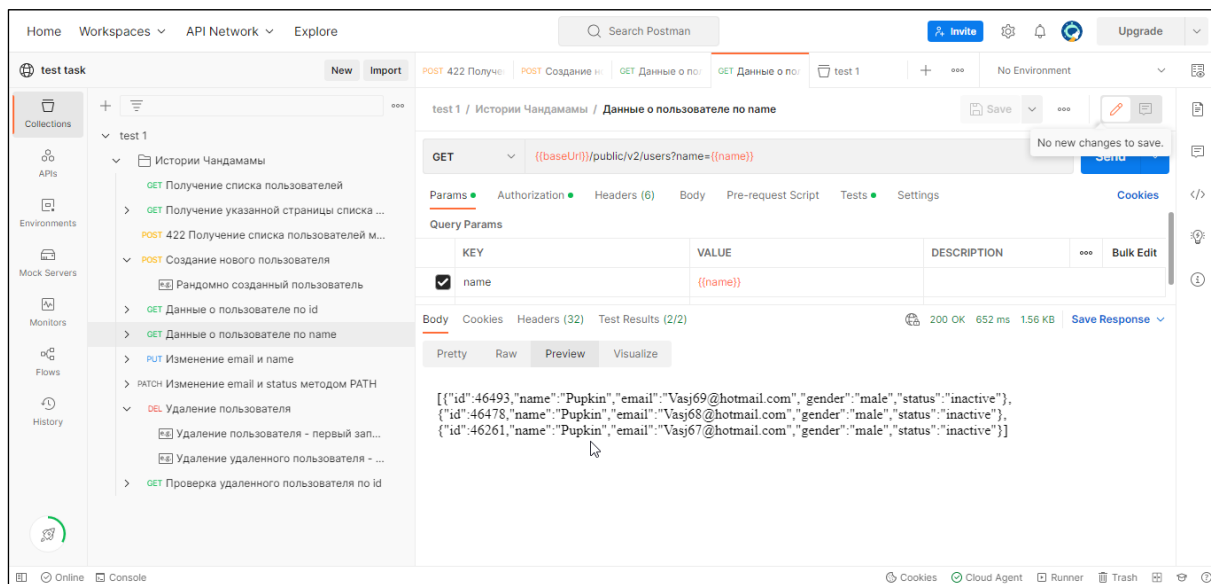
данные по имени:

The screenshot shows the Postman interface with the same test collection. The selected request is a GET request titled 'test 1 / Истории Чандаммы / Данные о пользователе по name'. The URL is `{{baseUri}}/public/v2/users?name={{name}}`. The 'Query Params' section shows a parameter 'name' with the value `{{name}}`. The 'Body' tab is selected, showing a JSON response in 'Pretty' format:

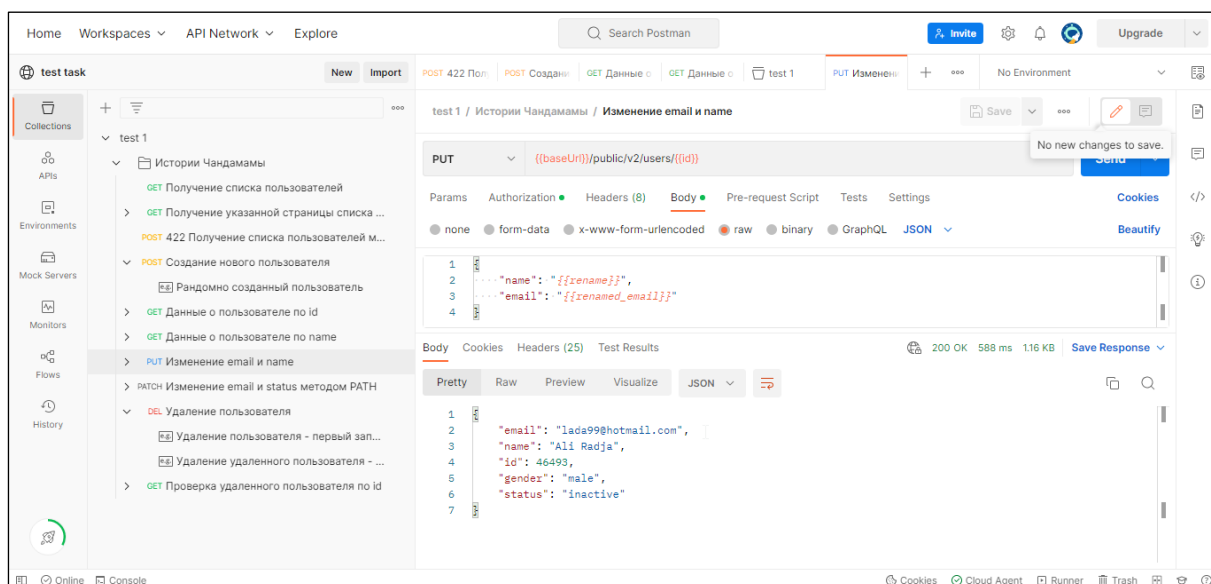
```
1 {
2   "id": 46261,
3   "name": "Pupkin",
4   "email": "Vasj67@hotmail.com",
5   "gender": "male",
6   "status": "inactive"
7 }
8
```

The status bar at the bottom indicates a 200 OK response with a 604 ms duration and 1.34 KB of data.

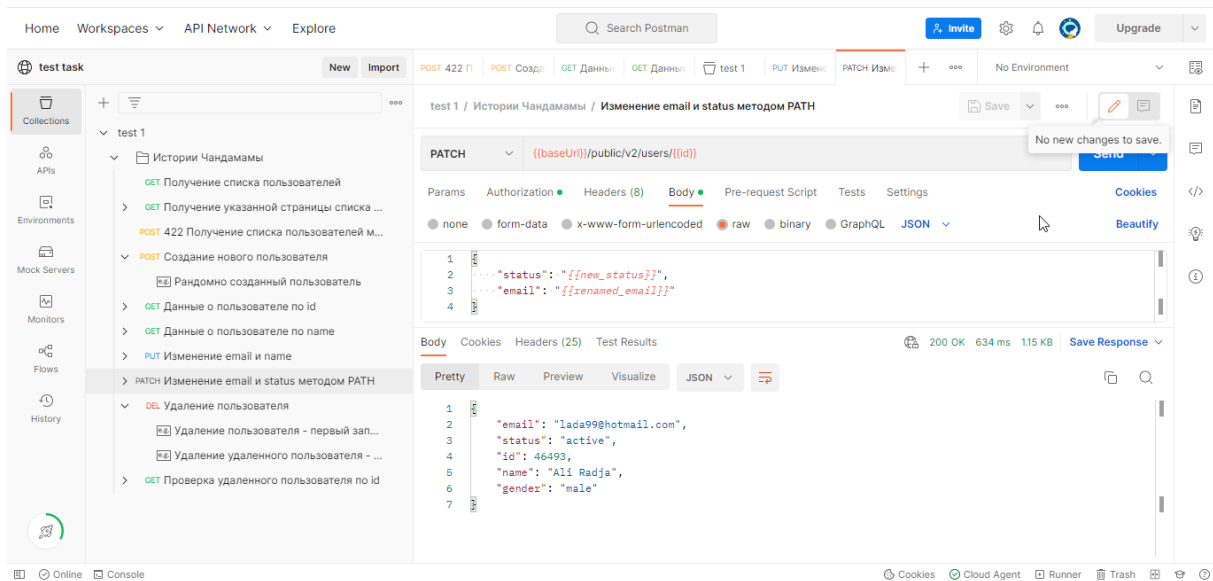
Запрос на рисунке - данные о пользователе можно получить методом GET передав в параметры значения- id, email - так как они уникальные у каждого пользователя:



Запрос на изменение данных:



Метод PATCH используется для частичного изменения ресурса, повторный запрос не меняет состояние сервера и приводит к тому же результату. PUT создает новый ресурс или заменяет представление целевого ресурса, данными представленными в теле запроса, перезаписывает целиком. В данном случае отличие только в изменяемых данных:



В запросе ниже – в теле ответа пустой список / массив, статус "204 NO CONTENT". При GET запросе удаленного пользователя статус код - "200 ОК" - такой ответ, потому что пользователя с таким id не существует, значит он удачно удален.

