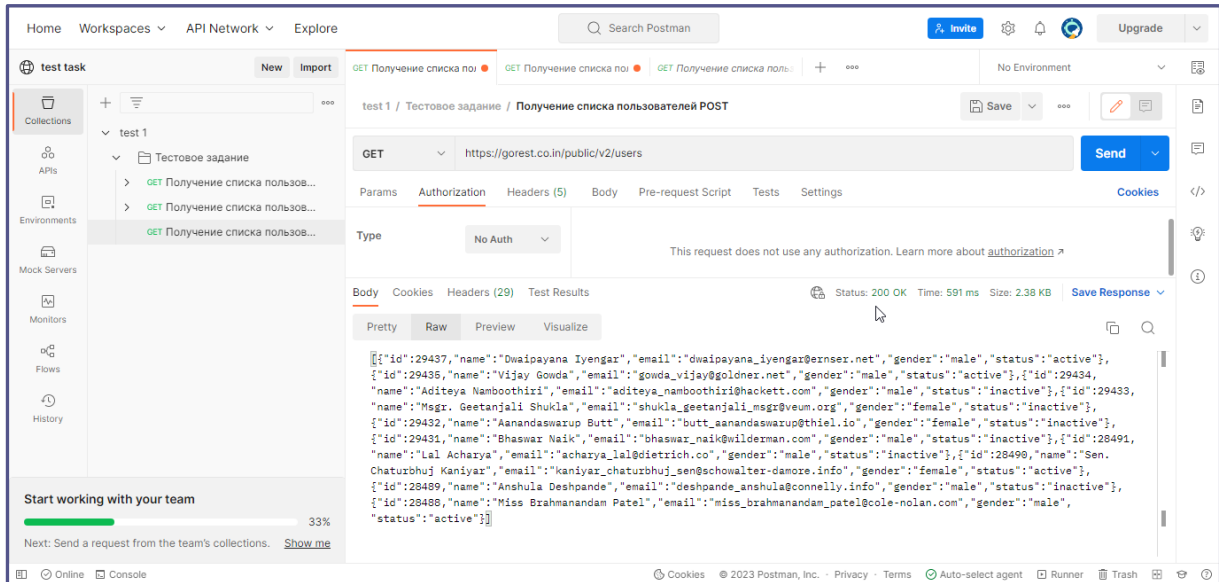
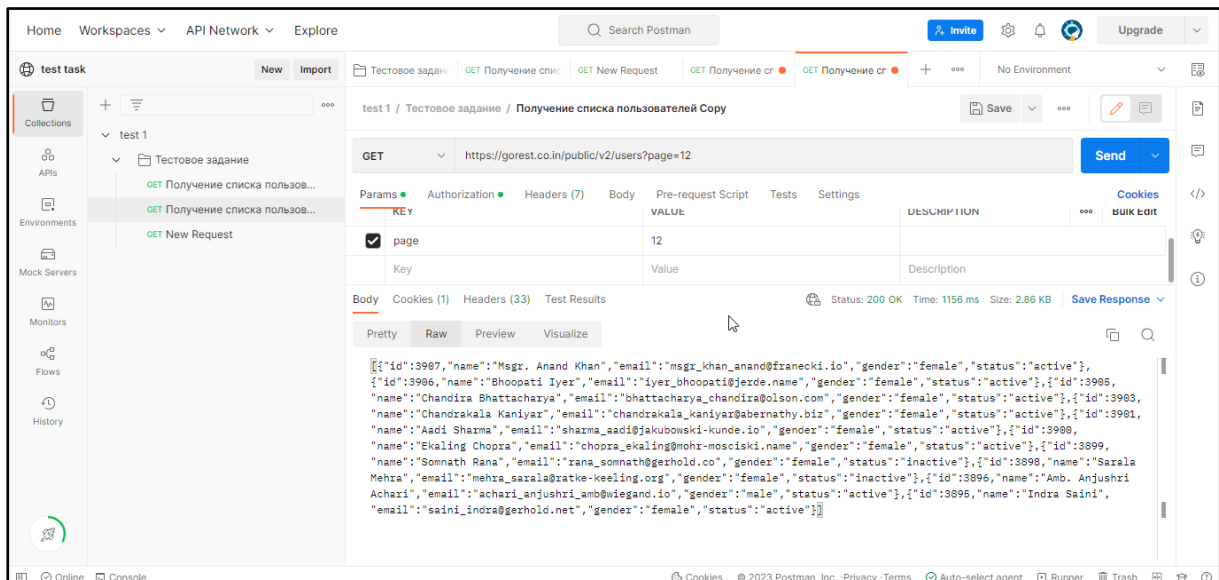


В данной коллекции использованы переменные для окружения test1, получение id из ответа при создании пользователя с присвоением ее в переменную, сниппеты по статус кодам и времени на запрос.

Вопрос 1. Запросом методом GET с вводом в вкладке Authorization токена, полученного при регистрации, статус код ответа - "200 OK", в теле ответа список определенных пользователей (из страницы 1).

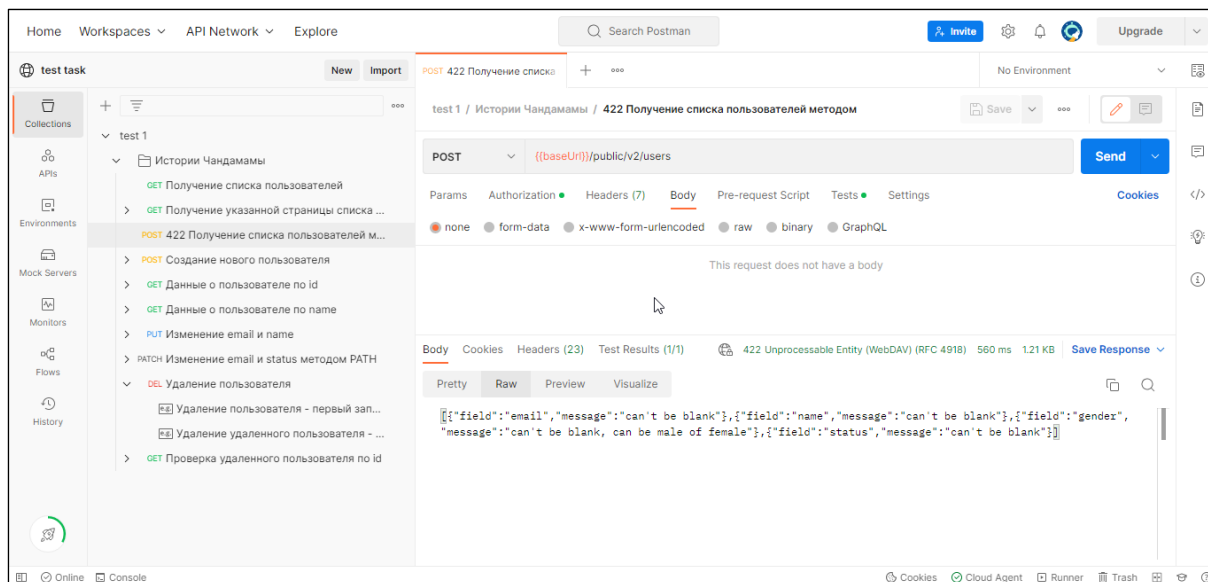


Вопрос 2. Другой список пользователей из страницы 12.

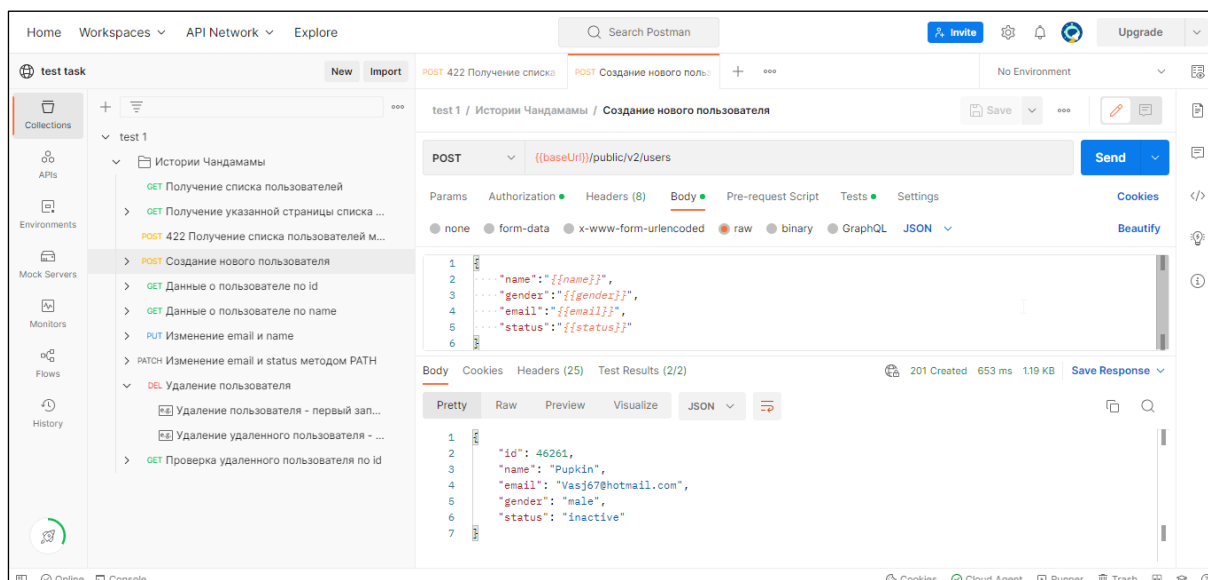


Вопрос 3. Отличия в том, что в п.2 задаем параметры страницы `page = 12`, а в первом запросе условно идет страница 1.

Вопрос 4. 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.



Вопрос 5. Пришел ответ со статус кодом "201 Created" – пользователь создан, с данными пользователя в теле ответа.



Вопрос 6.

Данные по id :

The screenshot shows the Postman interface with a test collection named 'test task'. The selected test is 'test 1 / Истории Чандаммы / Данные о пользователе по id'. The request is a GET method to the endpoint `{{baseUri}}/public/v2/users?id={{id}}`. The 'Query Params' section shows a parameter 'id' with the value `{{id}}`. The response is a 200 OK status with a 578 ms response time and a 1.34 KB body. The response body is displayed in the 'Pretty' view as a JSON object:

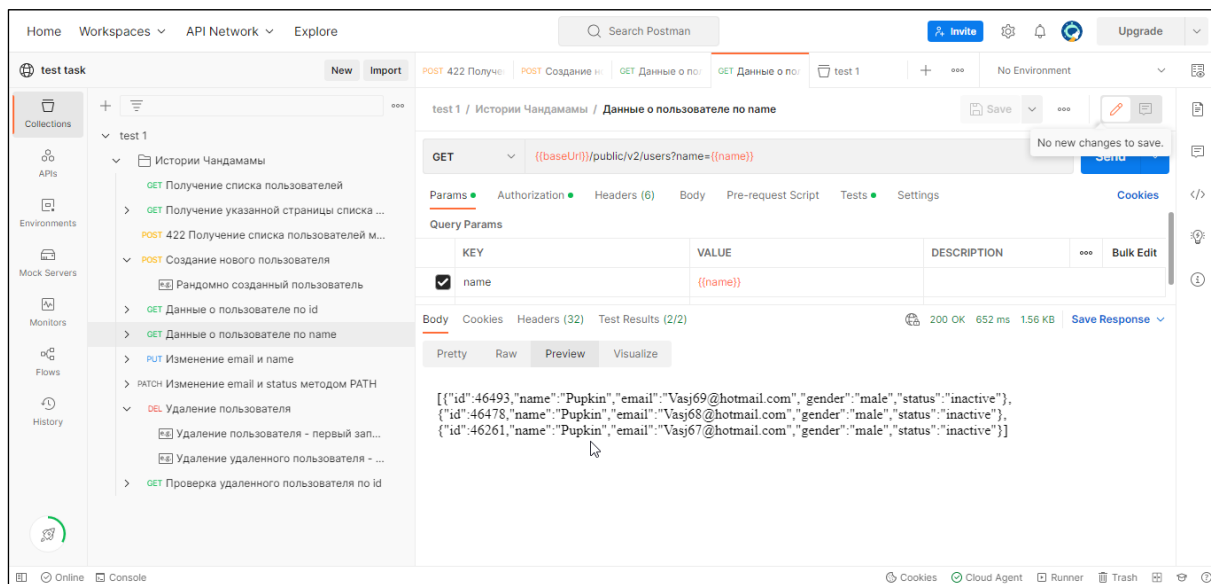
```
{
  "id": 46261,
  "name": "Pupkin",
  "email": "Vasj67@hotmail.com",
  "gender": "male",
  "status": "inactive"
}
```

Данные по имени:

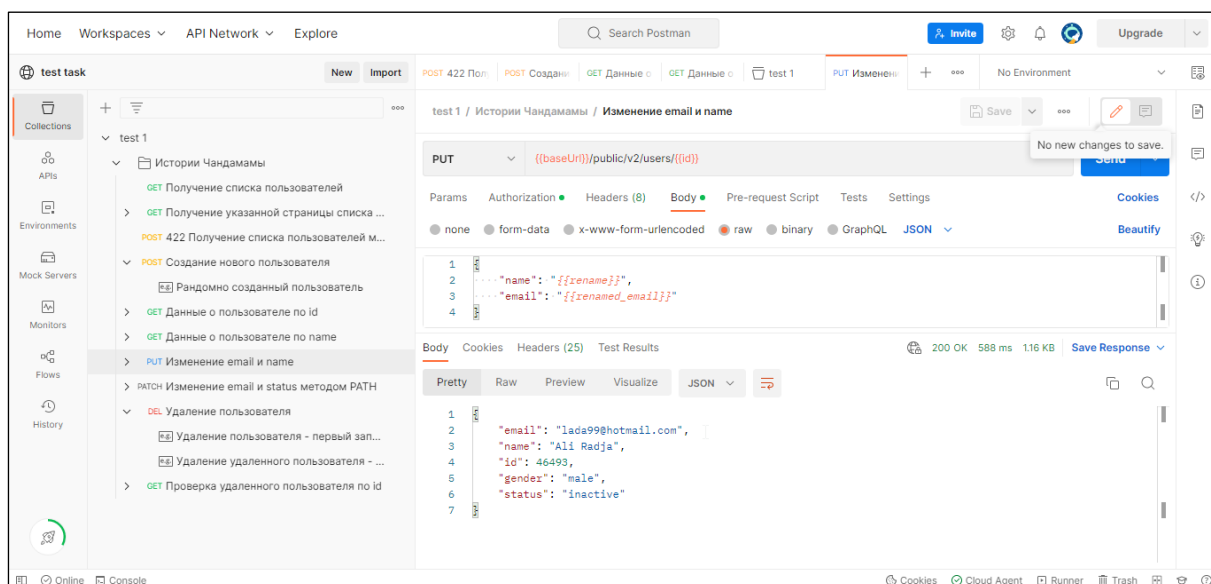
The screenshot shows the Postman interface with the same test collection 'test task'. The selected test is 'test 1 / Истории Чандаммы / Данные о пользователе по name'. The request is a GET method to the endpoint `{{baseUri}}/public/v2/users?name={{name}}`. The 'Query Params' section shows a parameter 'name' with the value `{{name}}`. The response is a 200 OK status with a 604 ms response time and a 1.34 KB body. The response body is displayed in the 'Pretty' view as a JSON object:

```
{
  "id": 46261,
  "name": "Pupkin",
  "email": "Vasj67@hotmail.com",
  "gender": "male",
  "status": "inactive"
}
```

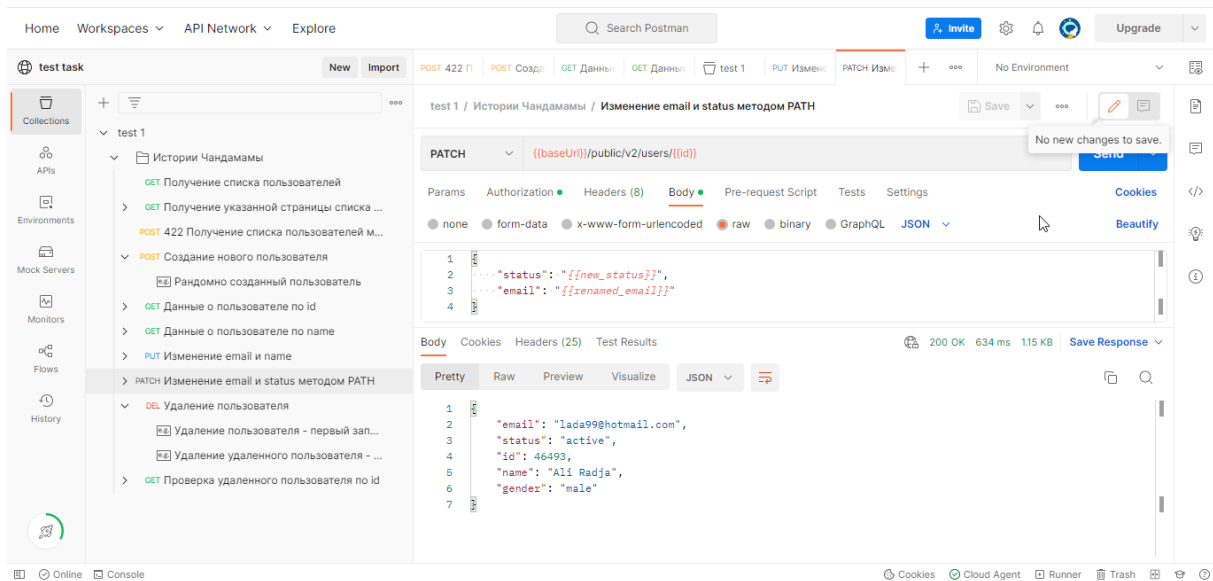
Вопрос 7. Данные о пользователе можно получить методом GET передав в параметры значения- id, email - так как они уникальные у каждого пользователя.



Вопрос 8.



Вопрос 9. 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.



Вопрос 10. В теле ответа пустой список / массив, статус "204 NO CONTENT". При GET запросе удаленного пользователя статус код - "200 OK" - такой ответ, потому что пользователя с таким id не существует, значит он удачно удален.

