

Первая часть задания выполняется в Postman.

Загрузите решение в виде архива тестового файла с ответами на вопросы, скриншотами процесса и готовой коллекцией запросов Postman и пришлите ссылку на него.

Подготовительный этап

Авторизуйтесь на gorest.co.in и получите токен для аутентификации (http bearer tokens)

Задание

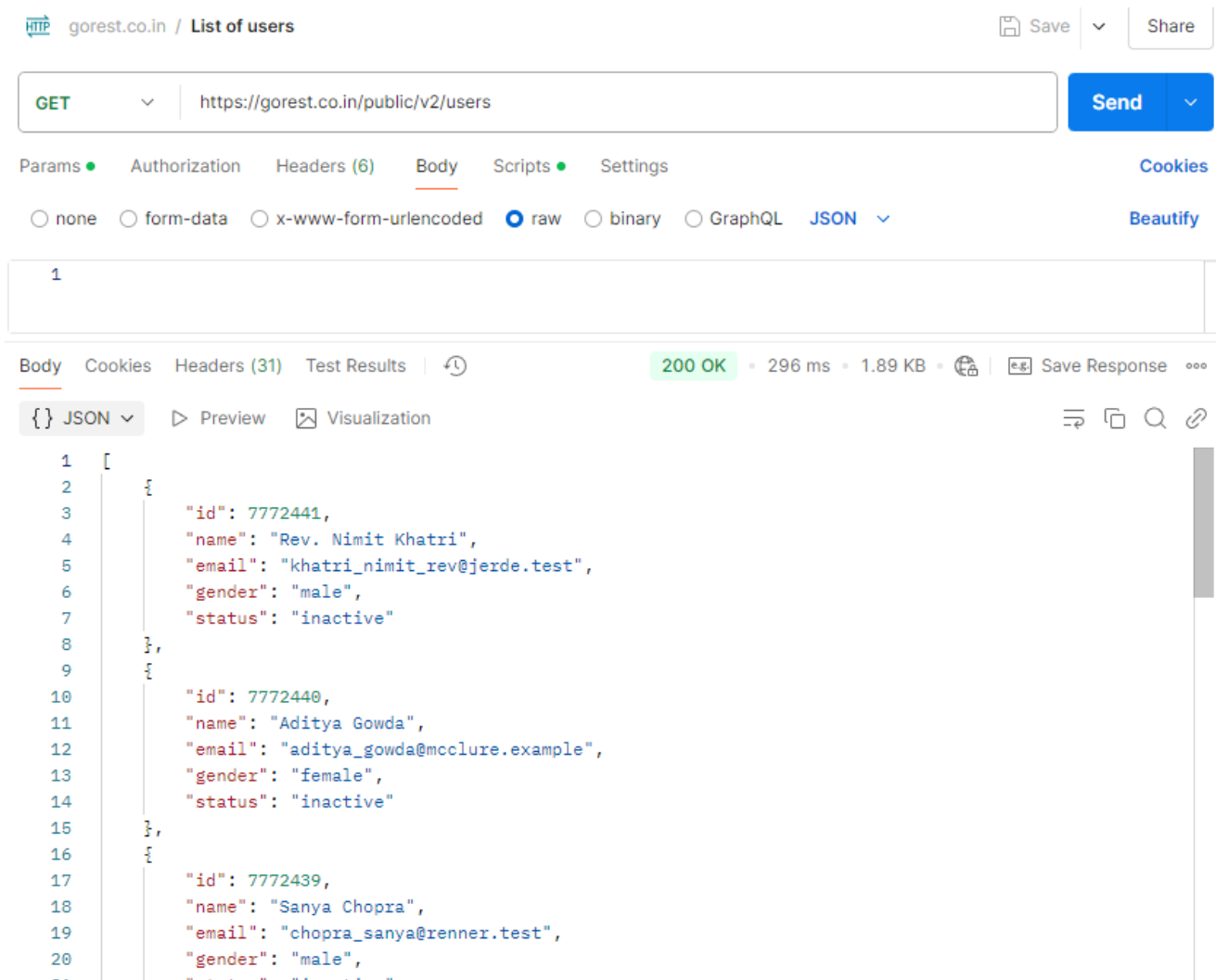
1. Получите список пользователей.

Каким запросом получили список пользователей?

GET <https://gorest.co.in/public/v2/users>

Какой статус ответа получили? 200 OK

Что содержится в теле ответа? Сделайте скриншот.



2. Модифицируйте запрос и получите страницу 12. Что содержится в теле ответа? Сделайте скриншот.

HTTP gorest.co.in / List of users Save Share

GET ▼ Send ▼

Params • Authorization Headers (6) Body Scripts • Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> page	12			

Body Cookies Headers (31) Test Results 200 OK • 825 ms • 1.93 KB • 🌐 📄 Save Response ⋮

{} JSON ▼ ▶ Preview 📊 Visualization 🔍 🔗

```
1 [
2   {
3     "id": 7772330,
4     "name": "Sarvin Rana DC",
5     "email": "dc_sarvin_rana@vandervort.test",
6     "gender": "male",
7     "status": "inactive"
8   },
9   {
10    "id": 7772329,
11    "name": "Fr. Hamsini Dwivedi",
12    "email": "fr_dwivedi_hamsini@dietrich-feeney.test",
13    "gender": "male",
14    "status": "inactive"
15  },
16  {
17    "id": 7772328,
18    "name": "Chitramala Saini",
19    "email": "chitramala_saini@king.test",
20    "gender": "male",
21    "status": "inactive"
22  }
23 ]
```

3. Какие отличия запроса в п.2 от запроса в п.1?

GET https://gorest.co.in/public/v2/users VS. GET https://gorest.co.in/public/v2/users?page=12
Добавлен параметр: ?page=12

4. Сделайте post запрос для получения пользователя с пустым телом запроса. Какой ответ пришел и почему? Сделайте скриншот.

401 Unauthorized. Метод POST посылает данные на сервер. Если нечего посылать, ответ запроса будет ошибкой

HTTP gorest.co.in / List of users Save Share

POST ▼ Send ▼

Params Authorization Headers (7) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

1

Body Cookies Headers (22) Test Results 401 Unauthorized • 795 ms • 1.2 KB • 🌐 📄 Save Response ⋮

{} JSON ▼ ▶ Preview 📊 Visualize 🔍 🔗

```
1 {
2   "message": "Authentication failed"
3 }
```

5. Напишите запрос для создания пользователя, используя метод raw => json. Какой ответ пришел? Сделайте скриншот.

201 Created

The screenshot shows a REST client interface for the URL `https://gorest.co.in/public/v2/users`. The method is set to **POST**. The request body is in raw JSON format:

```
1 {
2   "name": "Ilove chickennuggets",
3   "email": "pandey_oormila@ky",
4   "gender": "female",
5   "status": "active"
6 }
```

The response status is **201 Created** with a response time of 846 ms and a size of 1.46 KB. The response body is shown in JSON format:

```
1 {
2   "id": 7774894,
3   "name": "Ilove chickennuggets",
4   "email": "pandey_oormila@ky",
5   "gender": "female",
6   "status": "active"
7 }
```

6. Напишите запрос, возвращающий в ответе запись о созданном пользователе. Сделайте скриншот.

The screenshot shows a REST client interface for the URL `https://gorest.co.in/public/v2/users/7774894`. The method is set to **GET**. The response status is **200 OK** with a response time of 343 ms and a size of 1.42 KB. The response body is shown in JSON format:

```
1 {
2   "id": 7774894,
3   "name": "Ilove chickennuggets",
4   "email": "pandey_oormila@ky",
5   "gender": "female",
6   "status": "active"
7 }
```

7. Создайте еще несколько пользователей с одинаковыми значениями "name", но разными email. Получите список всех только что созданных пользователей. Сделайте скриншот. Какими ещё get запросами можно получить запись о пользователе?

GET `https://gorest.co.in/public/v2/users` + нужные параметры

HTTP gorest.co.in / **Get user details** Save Share

GET https://gorest.co.in/public/v2/users/?name=Ilove chickennuggets Send

Params • Authorization • Headers (7) Body Scripts Settings Cookies

Query Params

<input type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input type="checkbox"/>	page	7			
<input type="checkbox"/>	gender	female			
<input type="checkbox"/>	gender	male			
<input checked="" type="checkbox"/>	name	Ilove chickennuggets			
<input type="checkbox"/>	email	key			

Body Cookies Headers (31) Test Results 200 OK • 438 ms • 1.17 KB Save Response

JSON Preview Visualize

```
2 {
3   "id": 7774894,
4   "name": "Ilove chickennuggets",
5   "email": "pandey_oormila@ky",
6   "gender": "female",
7   "status": "active"
8 },
9 {
10  "id": 7772890,
11  "name": "Ilove chickennuggets",
12  "email": "pandey_oormila@key",
13  "gender": "female",
14  "status": "active"
15 }
```

8. Напишите запрос с методом put на изменение email и name на другое значение. Проверьте изменение get запросом. Сделайте скриншот.

HTTP gorest.co.in / **Update user details** Save Share

PUT https://gorest.co.in/public/v2/users/7774909 Send

Params Authorization • Headers (9) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   "name": "Ilove memes",
3   "email": "pandey_oormila@oky"
4 }
5
6 }
```

Body Cookies Headers (27) Test Results 200 OK • 383 ms • 1.41 KB Save Response

JSON Preview Visualize

```
1 {
2   "email": "pandey_oormila@oky",
3   "name": "Ilove memes",
4   "id": 7774909,
5   "gender": "male",
6   "status": "active"
7 }
```

gorest.co.in / Get user details

GET https://gorest.co.in/public/v2/users/7774909

Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Body Cookies Headers (27) Test Results 200 OK • 772 ms • 1.41 KB • Save Response

{ } JSON Preview Visualize

```
1 {
2   "id": 7774909,
3   "name": "Ilove memes",
4   "email": "pandey_oormila@oky",
5   "gender": "male",
6   "status": "active"
7 }
```

9. Чем отличаются запросы put и patch?

PUT требует для отправки все поля, даже если изменяется часть данных. PATCH – только те, которые нужно изменить.

PUT – идемпотентен. PATCH – неидемпотентен.

PUT – если поле не существует, может создать. PATCH – может завершиться ошибкой, если поля не существует.

Напишите запрос с методом patch для корректировки полей status и email. Сделайте скриншот. В чем отличие полученного результата, от результата запроса из п.8?

gorest.co.in / Update user details

PATCH https://gorest.co.in/public/v2/users/7774909

Send

Params Authorization Headers (9) Body Scripts Settings Cookies

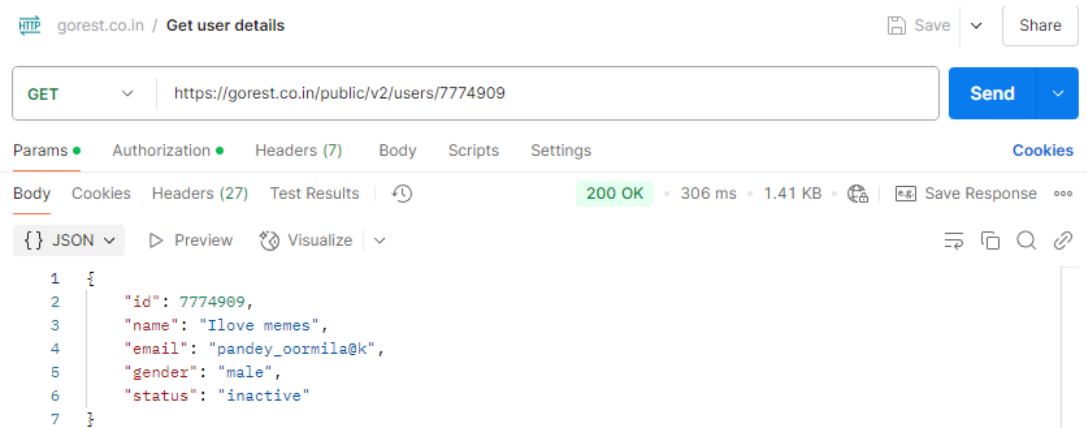
☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "status": "inactive",
3   "email": "pandey_oormila@k"
4 }
5
6 }
```

Body Cookies Headers (27) Test Results 200 OK • 299 ms • 1.41 KB • Save Response

{ } JSON Preview Visualize

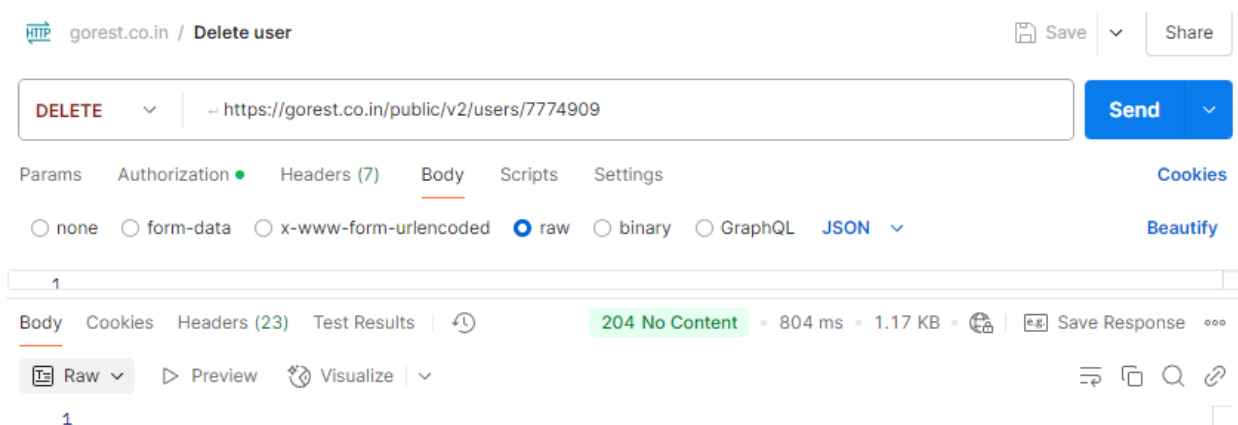
```
1 {
2   "email": "pandey_oormila@k",
3   "status": "inactive",
4   "id": 7774909,
5   "name": "Ilove memes",
6   "gender": "male"
7 }
```



Отличие только в полях status и email

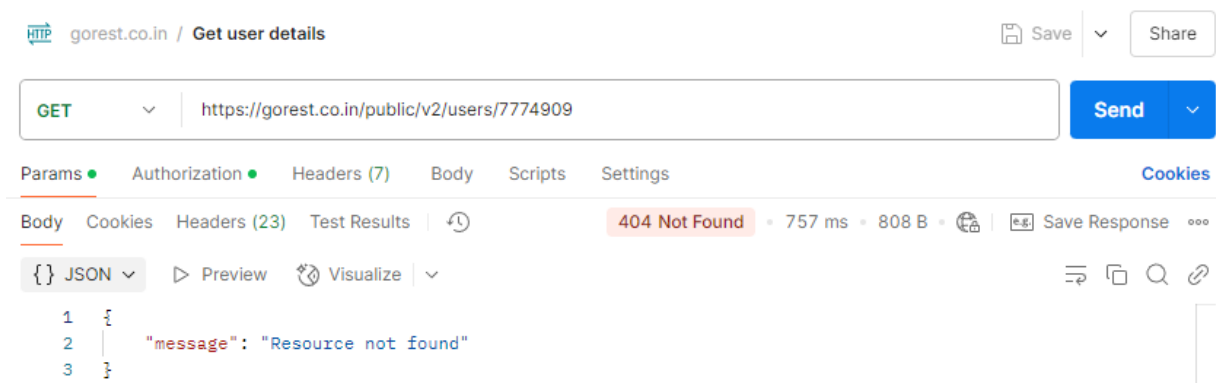
10. Удалите пользователя, созданного в п.5. Что содержится в теле ответа? Сделайте скриншот.

204 No Content. Запрос выполнен успешно



Выполните get запрос удаленного пользователя. Какой ответ пришел и почему? Сделайте скриншот.

404 Not Found. Не найдено. Пользователь не найден, т.к. был удалён



11. Удалите всех ранее созданных пользователей.

12. Сохраните коллекцию получившихся запросов в Postman.

<https://www.postman.com/dream-team-4768/workspace/for-cv/collection/42081846-4645908c-0dbb-4fee-ad16-6e7ed85ff138?action=share&creator=42081846>