

Склонируйте репозиторий и разверните его локально. Изучите документацию по работе с сервером. Выполните следующие задания, используя Postman:

1. Выведите список всех пользователей
2. Напишите тест на то, что существует пользователь "Nicholas Runolfsdottir V"
3. Создайте новый пост с помощью POST запроса
4. Удалите созданный пост
5. Напишите тест на удаление поста

## 1. Выведите список всех пользователей

The screenshot shows the Postman interface with a new collection named "Get users". A GET request is configured to `http://localhost:3000/users`. The response is a JSON object representing a user.

**Request:**

- Method: GET
- URL: `http://localhost:3000/users`

**Response:**

Status: 200 OK, Time: 44 ms, Size: 6.46 KB

```
{
  "id": 7,
  "name": "Kurtis Weissnat",
  "username": "Elwyn.Skiles",
  "email": "Telly.Hoeger@billy.biz",
  "address": {
    "street": "Rex Trail",
    "suite": "Suite 280",
    "city": "Howemouth",
    "zipcode": "58804-1099",
    "geo": {
      "lat": "24.8918",
      "lng": "21.8984"
    }
  }
}
```

## 2. Напишите тест на то, что существует пользователь "Nicholas Runolfsdottir V"

The screenshot shows the Postman interface for a REST client request. The request is a GET method to the URL `http://localhost:3000/users`. The **Tests** tab is active, displaying a JavaScript test script:

```
1 pm.test("Response data is correct", function () {
2   pm.expect(pm.response.text()).to.include("Nicholas Runolfsdottir V");
3 });
```

Below the script, the **Test Results (1/1)** section shows a single test that has passed:

- PASS** Response data is correct

The status bar at the bottom indicates the request was successful with a status of 200 OK, a time of 34 ms, and a size of 6.46 KB. The bottom navigation bar includes icons for Cookies, Capture requests, Runner, Trash, and a grid icon.

### 3. Создайте новый пост с помощью POST запроса

The screenshot displays the Postman application interface. At the top, the URL bar shows 'http://localhost:3000/posts' with a 'POST' method selected. The 'Body' tab is active, showing a JSON payload: 

```
{  "userId": 10,  "title": "attestation task",  "body": "Create a new post using a POST request"}
```

. The 'Send' button is visible. Below the request editor, the 'Body' tab of the response is selected, showing the received JSON: 

```
{  "userId": 10,  "title": "attestation task",  "body": "Create a new post using a POST request",  "id": 101}
```

. The status bar at the bottom indicates a '201 Created' status, a response time of '43 ms', and a size of '623 B'.

New Collection / Add post

Save

POST http://localhost:3000/posts Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

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

```
1 {
2   "userId": 10,
3   "title": "attestation task",
4   "body": "Create a new post using a POST request"
5 }
```

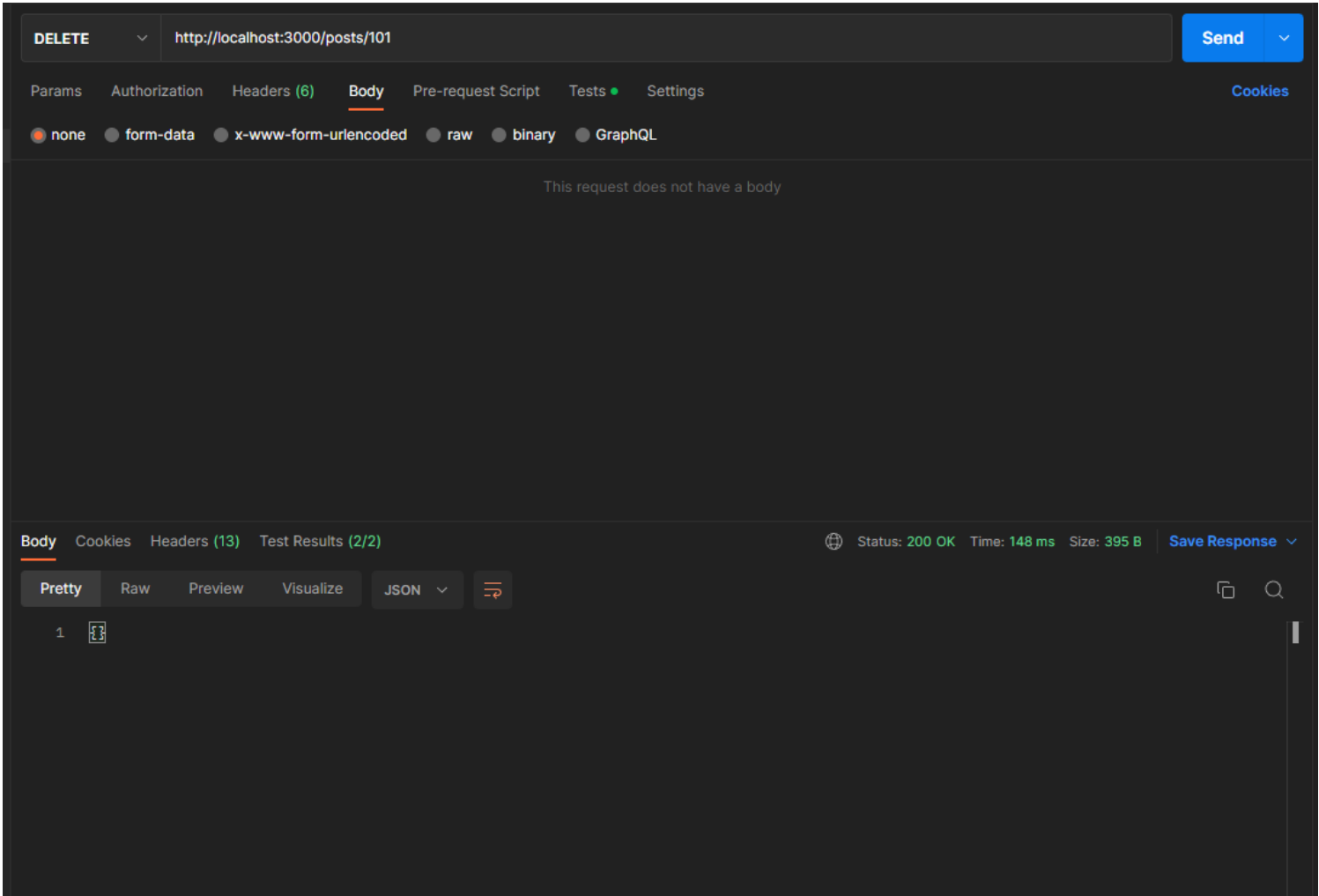
Body Cookies Headers (15) Test Results

Status: 201 Created Time: 43 ms Size: 623 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 10,
3   "title": "attestation task",
4   "body": "Create a new post using a POST request",
5   "id": 101
6 }
```

4. Удалите созданный пост



5. Напишите тест на удаление поста

DELETE

http://localhost:3000/posts/101

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

1

pm.test("Status code is 200", function () {

2

pm.response.to.have.status(200);

3

});

4

5

pm.test("Post deleted", function () {

6

var jsonData = pm.response.json();

7

pm.expect(jsonData.value).to.eql(undefined);

8

});

Test scripts are written in JavaScript, and are run after the response is received. Learn more about [tests scripts](#)

Snippets

Get an environment variable

Get a global variable

Get a variable

Get a collection variable

Set an environment variable

Set a global variable

Set a collection variable

Body

Cookies

Headers (13)

Test Results (2/2)

Status: 200 OK

Time: 148 ms

Size: 395 B

Save Response

All

Passed

Skipped

Failed

PASS

Status code is 200

PASS

Post deleted