

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Домашняя работа 4

Выполнила:

Красюк Карина

Группа К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

## Задача

- реализовать тестирование API средствами Postman;
- написать тесты внутри Postman.

## Ход работы

1. Реализован тест авторизации (для user). Этот тест проверяет, что регистрация возвращает статус 200 и содержит в ответе поле 'token'.

```
pm.test("Статус 200", function () {  
    pm.response.to.have.status(200);  
});  
  
pm.test("Содержит 'token'", function () {  
    const response = pm.response.json();  
    pm.expect(response).to.have.property('token');  
});
```

2. Реализован тест регистрации. Этот тест проверяет, что регистрация возвращает статус 201 и содержит в ответе поля 'id' и 'email'.

```
pm.test("Статус 201", function () {  
    pm.response.to.have.status(201);  
});  
  
pm.test("Содержит 'id' и 'email'", function () {  
    const res = pm.response.json();  
    pm.expect(res).to.have.all.keys('id', 'email');  
});
```

3. Реализован тест получения всех записей, который проверяет успешный статус ответа (200) и убеждается, что ответ является массивом.

```
pm.test("Статус 200", function () {  
    pm.response.to.have.status(200);  
});  
  
pm.test("Ответ — массив", function () {  
    pm.expect(pm.response.json()).to.be.an('array');  
});
```

4. Реализован тест получения записей по id. Этот тест проверяет, что сервер возвращает успешный статус-код 200 (ОК) и тело ответа содержит поле 'id'.

```
pm.test("Успешный статус 200", function() {  
    pm.response.to.have.status(200);  
});
```

```
pm.test("Наличие поля 'id' в ответе", function() {  
    const jsonData = pm.response.json();  
    pm.expect(jsonData).to.have.property('id');  
});
```

5. Реализован тест создания записей. Этот тест проверяет успешную регистрацию, подтверждая статус 201 и наличие ID в ответе сервера.

```
pm.test("Регистрация: успешное создание (статус 201)",  
function() {  
    pm.response.to.have.status(201);  
});
```

```
pm.test("Регистрация: ответ содержит идентификатор", function()  
{  
    const response = pm.response.json();  
    pm.expect(response).to.have.property('id');  
});
```

6. Реализован тест обновления записей. Этот тест проверяет успешный запрос (статус 200) и наличие поля 'message' в ответе сервера.

```
pm.test("Успешный статус 200", function () {  
    pm.response.to.have.status(200);  
});
```

```
pm.test("Ответ содержит поле 'message'", function () {  
    const response = pm.response.json();  
    pm.expect(response).to.have.property('message');  
});
```

7. Реализован тест удаления записей. Этот тест проверяет успешное удаление (статус 200) и наличие подтверждающего сообщения "успешно удалена" в ответе сервера.

```
pm.test("Успешный статус удаления (200)", function() {  
    pm.response.to.have.status(200);  
});
```

```
pm.test("Подтверждение удаления", function() {  
    pm.expect(pm.response.json().message).to.include("успешно  
удалена");  
});
```

## **Вывод**

В данной домашней работе были реализованы тесты в Postman для ранее написанного приложения. Тесты нацелены на помощь в проверке работоспособности отдельных частей приложения.