

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Практическая работа № 4
“Документирование API из ЛР2 средствами Swagger,
Postman”

Выполнил:

Чан Дык Минь

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

Протестировать REST API из ЛР 2 средствами Postman. Необходимо написать 2-3 теста для конкретных эндпоинтов + 1 общий тест для всех эндпоинтов.

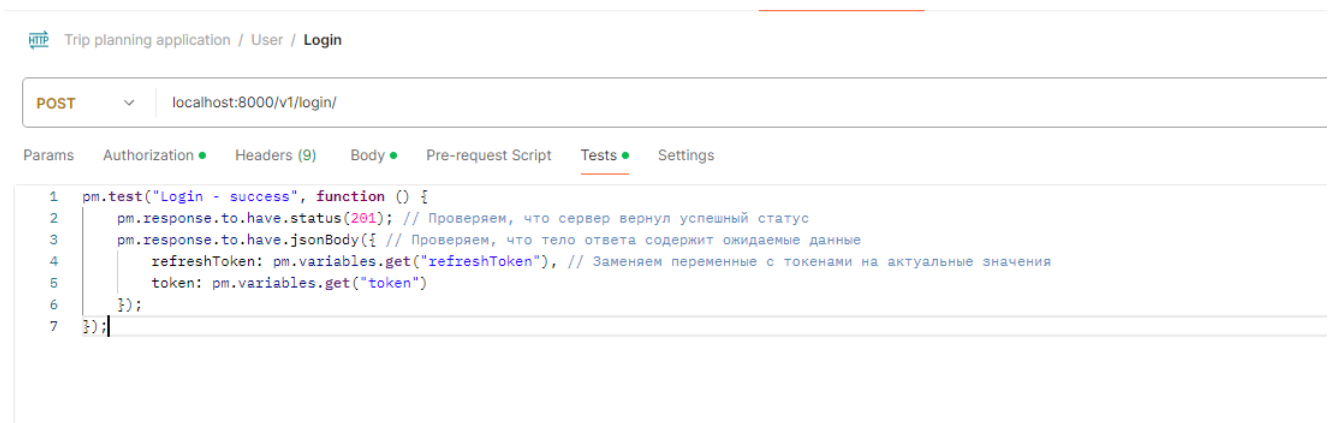
Ход работы

В ходе выполнения работы задокументированный REST API был импортирован в Postman.

Мы напишем тест для проверки функции регистрации. При регистрации с уже существующим адресом электронной почты система выдаст ошибку 400 с надписью "Сообщение": "Email already exists"



Далее мы проверим функцию входа в систему. Мы проверим, возвращает ли ответ полный обновленный токен и токен.



Мы присвоим вновь полученный Токен переменной authToken.

Trip planning application / User / **get current user**

GET localhost:8000/v1/users/me

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Type Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we've masked the values.

Token {{authToken}}

Теперь мы сможем протестировать текущий метод Get user. Ответ должен быть возвращен с полной информацией о пользователе.

Trip planning application / User / **get current user**

GET localhost:8000/v1/users/me

Params Authorization Headers (7) Body Pre-request Script Tests Settings

```
1 pm.test("Get current user - success", function () {
2   pm.response.to.have.status(200);
3   pm.response.to.have.jsonBody();
4   pm.response.to.have.jsonBody("id");
5   pm.response.to.have.jsonBody("name");
6   pm.response.to.have.jsonBody("email");
7   pm.response.to.have.jsonBody("createdAt");
8   pm.response.to.have.jsonBody("activities");
9 });
10
```

Результаты запуска тестов показывают, что API работает нормально. Мы можем написать аналогично для других конечных точек.

Trip planning application - Run results						Run Again	Automate Run	+ New Run	Export Results
Ran today at 01:21:17 · View all runs									
Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time				
Runner	API	1	551ms	3	24 ms				
All Tests Passed (3) Failed (0) Skipped (0)						View Summary			
Iteration 1						1			
POST Register									
localhost:8000/v1/register/						400 Bad Request 23 ms 307 B			
PASS Ошибка при регистрации с существующим email									
POST Login									
localhost:8000/v1/login/						201 Created 40 ms 526 B			
PASS Login - success									
GET get current user									
localhost:8000/v1/users/me						200 OK 10 ms 587 B			
PASS Get current user - success									

Вывод

В рамках домашнего задания №5 все API были успешно протестированы. Я также научился писать тесты для каждой конечной точки.