

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 4

Выполнил:

Якунин Андрей

Группа
К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

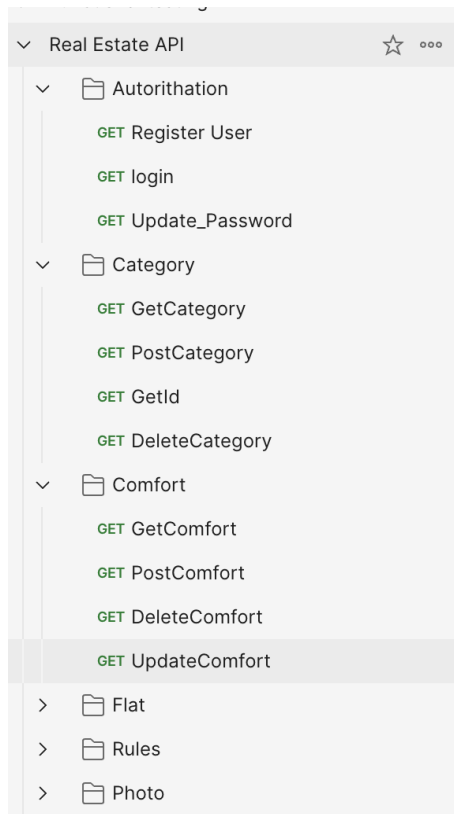
2025 г.

Задача

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

Ход работы

Нужно написать тесты в Postman



Регистрация

```
pm.test("Статус 201", function () {  
  
    pm.response.to.have.status(201);  
  
});
```

```
pm.test("Ответ содержит 'id' и 'email'", function () {  
  
    const res = pm.response.json();  
  
    pm.expect(res).to.have.property("id");  
  
    pm.expect(res).to.have.property("email");});
```

Авторизация

```
pm.test("Статус 200", function () {

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

});

pm.test("Ответ содержит токен", function () {

    const res = pm.response.json();

    pm.expect(res).to.have.property("token");

});
```

Обновление пароля

```
pm.test("Статус 200", function () {

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

});

pm.test("Ответ содержит сообщение", function () {

    const res = pm.response.json();

    pm.expect(res).to.have.property("message", "Password updated successfully");

});
```

Получение категории

```
pm.test("Status code is 200", () => {

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

});

pm.test("Response is an array", () => {

    const json = pm.response.json();
```

```
    pm.expect(json).to.be.an("array");  
  });
```

Создание категории

```
pm.test("Status code is 201", () => {  
  
    pm.response.to.have.status(201);  
});
```

```
pm.test("Response contains valid id, name, and ownerId", () => {  
  
    const json = pm.response.json();  
  
    pm.expect(json).to.have.property("id").that.is.a("number").and.to.be.above(0);  
  
    pm.expect(json).to.have.property("name").that.is.a("string").and.to.have.length.above(0);  
});
```

Получение категории по id

```
pm.test("Status code is 200 or 404", () => {  
  
    pm.expect(pm.response.code === 200 || pm.response.code === 404).to.be.true;  
});  
  
if (pm.response.code === 200) {  
  
    pm.test("Response has category object with id", () => {  
  
        const json = pm.response.json();
```

```

    pm.expect(json).to.have.property("id");

    pm.expect(json).to.have.property("name");

  });

} else if (pm.response.code === 404) {

  pm.test("Response has 'Category not found' message", () => {

    const json = pm.response.json();

    pm.expect(json.message).to.eql("Category not found");

  });

}

```

Удаление категории

```

pm.test("Status code is 200 or 404", () => {

  pm.expect(pm.response.code === 200 || pm.response.code === 404).to.be.true;

});

if (pm.response.code === 200) {

  pm.test("Response has success message", () => {

    const json = pm.response.json();

    pm.expect(json.message).to.eql("Category deleted successfully");

  });

} else if (pm.response.code === 404) {

  pm.test("Response has 'Category not found' message", () => {

    const json = pm.response.json();

    pm.expect(json.message).to.eql("Category not found");

  });

}

```

Вывод

В ходе работы мной были разработаны и реализованы автоматизированные тесты для API с использованием Postman. Были созданы тесты, проверяющие корректность ответов сервера, статус кодов HTTP, а также наличие и правильность ключевых полей в теле ответа.

Особое внимание уделялось проверке:

- Статуса ответа (например, 201 Created для успешного создания ресурса);
- Наличия обязательных полей в ответе (например, `id`, `name`, `ownerId`);
- Типов и значений этих полей для подтверждения их корректности.

Такой подход позволяет оперативно выявлять ошибки на уровне API и обеспечивает стабильность и надежность работы сервиса. Тесты в Postman удобны для быстрой проверки как в процессе разработки, так и при интеграционном тестировании.

В дальнейшем планируется расширить покрытие тестов, добавив проверки на негативные сценарии и валидацию данных, что повысит качество и устойчивость продукта