# Отчет по тестированию API JSONPlaceholder

**Цель:** Провести тестирование API сервиса JSONPlaceholder, выполняя операции создания, обновления и удаления постов.  
**Дата тестирования:** 25.03.2025  
**Протестировал:** Осипенко А.С.

**Используемые инструменты:  
Postman**: для выполнения запросов к API и автоматического тестирования.  
**Python + Requests + Pytest**: для написания и выполнения тестов с использованием Python.

### 1. Тестирование с помощью Postman

**Запросы, проведенные в Postman:**

**Получить список постов:**

**Метод**: GET  
**URL**: <https://jsonplaceholder.typicode.com/posts>  
**Проверки**:  
Статус-код: 200Ответ: JSON должен содержать массив постов.  
Все посты должны иметь обязательные поля:  
**id (тип: number)  
title (тип: string)  
body (тип: string)  
userId (тип: number)**

**Создать пост:**

**Метод**: POST  
**URL**: <https://jsonplaceholder.typicode.com/posts>  
**Тело запроса**:

{

"title": "Test",

"body": "Test for the task",

"userId": 1

}

**Проверки**:  
Статус-код: 201  
Ответ должен содержать ID нового поста.  
Ответ должен содержать отправленные данные: title, body, userId.

**Удалить пост:**

**Метод**: DELETE  
**URL**: <https://jsonplaceholder.typicode.com/posts/1>  
**Проверки**:  
Статус-код: 200 или 404 (если пост не найден).

**Примечание:**

В Postman для каждого запроса добавлены тесты на проверку статус-кодов и структуры ответа с использованием JavaScript.

### 2. Тестирование с помощью Python

**Тесты с использованием библиотеки requests и pytest:**

**Тест на создание поста**:

**Метод**: POST  
**URL**: <https://jsonplaceholder.typicode.com/posts>  
**Проверки**:  
Статус-код: 201  
Ответ должен содержать отправленные данные: title, body, userId.

**Пример кода:**

def test\_create\_post():

url = "https://jsonplaceholder.typicode.com/posts"

new\_post = {

"title": "Test",

"body": "Test for the task",

"userId": 1

}

response = requests.post(url, json=new\_post)

# Проверка на статус-код 201

assert response.status\_code == 201

# Проверка, что ответ содержит правильные данные

response\_data = response.json()

assert response\_data["title"] == new\_post["title"]

assert response\_data["body"] == new\_post["body"]

assert response\_data["userId"] == new\_post["userId"]

**Тест на обновление поста**:

**Метод**: PUT  
**URL**: <https://jsonplaceholder.typicode.com/posts/1>  
**Проверки**:  
Статус-код: 200  
Ответ должен содержать обновленные данные: id, title, body, userId.

**Пример кода:**

def test\_update\_post():

url = "https://jsonplaceholder.typicode.com/posts/1"

updated\_post = {

"id": 1,

"title": "Updated Test Post",

"body": "This post has been updated",

"userId": 1

}

response = requests.put(url, json=updated\_post)

# Проверка на статус-код 200

assert response.status\_code == 200

response\_data = response.json()

# Проверка, что ответ содержит корректные данные

assert response\_data["id"] == updated\_post["id"]

assert response\_data["title"] == updated\_post["title"]

assert response\_data["body"] == updated\_post["body"]

assert response\_data["userId"] == updated\_post["userId"]

**Тест на удаление поста**:

**Метод**: DELETE  
**URL**: <https://jsonplaceholder.typicode.com/posts/1>  
**Проверки**:  
Статус-код: 200 (если пост был успешно удален).

**Пример кода:**

def test\_delete\_post():

url = "https://jsonplaceholder.typicode.com/posts/1"

response = requests.delete(url)

# Проверка на статус-код 200

assert response.status\_code == 200

### 3. Результаты тестов:

**1.** Все тесты в Postman и Python прошли успешно.  
 2. Запросы POST и PUT возвращают статус-коды 201 и 200 соответственно, как и ожидалось.  
 3. Ответы содержат правильные данные, которые соответствуют отправленным значениям.

4. Тесты DELETE подтверждают успешное удаление поста с кодом 200.

### 4. Заключение:

Процесс тестирования API JSONPlaceholder прошел успешно. Все основные операции (создание, обновление, удаление) были корректно протестированы как через Postman, так и через Python. Тесты подтвердили правильность работы API и соответствие возвращаемых данных ожиданиям.