

Appointment Service API Test Document

Introduction

This document outlines the testing procedures and test cases for the Appointment Service API.

Testing Environment

- **Framework:** Pytest
- **Server:** Flask Development Server
- **Test Client:** Flask Test Client
- **Rate Limiting:** 5 requests per second

Test Cases

1. Test Booking an Appointment

Description: This test case validates the functionality of booking an appointment through the API.

Pass criteria: Check the response code is 200(OK) and message is 'Appointment booked successfully'.

2. Test Booking Appointments with Rate Limiting

Description: This test case ensures that rate limiting works correctly for booking appointments.

Pass criteria: Check the response code is 200(OK) for request within rate limit, after that response code is 429 (too many requests).

3. Test Booking an Appointment Again (When Slot Is Not Available)

Description: This test case checks if booking an appointment again for an already booked slot correctly returns an error.

Pass criteria: Check the response code is 404 and message is 'Slot is not available'.

4. Test Booking an Appointment with an Invalid Slot

Description: This test case verifies that attempting to book an appointment with an invalid slot returns an error.

Pass criteria: Check the response code is 404 and message is 'Slot is not found'.

5. Test Checking Appointment Availability

Description: This test case validates the functionality of checking the availability of an appointment slot.

Pass criteria: Check the response code is 200 and message is 'Slot is available'.

6. Test Canceling an Appointment

Description: This test case ensures that canceling an appointment works correctly.

Pass criteria: Check the response code is 200 and message is 'Appointment cancelled successfully'.

7. Test Canceling an Invalid Appointment

Description: This test case checks if attempting to cancel an appointment with an invalid appointment ID returns an error.

Pass criteria: Check the response code is 404 and message is 'Slot not found'.

8. Test Listing All Appointments

Description: This test case verifies that listing all appointments returns a valid JSON response.

Pass criteria: Check the response code is 200 and valid JSON array.

Test Sample Output

```
C:\Users\megha\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packa
ges\Python311\site-packages\flask_limiter\extension.py:336: UserWarning: Using the in-memory storage for tracki
ng rate limits as no storage was explicitly specified. This is not recommended for production use. See: https:/
/flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage bac
kend.
  warnings.warn(
===== test session starts =====
platform win32 -- Python 3.11.4, pytest-7.4.0, pluggy-1.2.0
rootdir: C:\Users\megha\Covid_vaccine_appointment\tests
plugins: anyio-3.7.1
collected 9 items

test_appointment.py ..... [100%]

===== 9 passed in 2.17s =====
```

Conclusion

The Appointment Service API has been tested using the test cases mentioned in this document. The tests cover various aspects of the API, including booking, rate limiting, canceling, and checking availability of appointments.