# 🧘‍♀️ Fitness Studio Booking API - Updated Documentation

This document provides a comprehensive overview of the Flask-based Booking API developed for a fictional fitness studio. The system includes support for class listing, booking, viewing user-specific bookings, robust validation, timezone support, structured logging and Swagger (OpenAPI) documentation.

## ✅ Features

* • Book fitness classes with slot tracking
* • Prevents duplicate bookings
* • Fetch bookings per user via email
* • Time zone aware class schedule
* • Structured logging to separate info and error log files
* • Swagger UI with live documentation and request validation
* • Validation using Pydantic
* • SQLite + SQLAlchemy for database and ORM

## ⚙️ Setup Instructions

1. Create a virtual environment and install dependencies:

python3 -m venv venv  
 source venv/bin/activate  
 pip install -r requirements.txt

2. Seed the database with initial class data: python seed\_data.py

3. Start the server: python main.py

Swagger UI: http://127.0.0.1:5000/docs

## 🌐 API Endpoint Specifications

### GET /classes

Returns all scheduled fitness classes. You can pass an optional 'timezone' query parameter to adjust the datetime.

Example:

| curl http://127.0.0.1:5000/classes?timezone=America/New\_York |
| --- |

Response: JSON list of classes with name, datetime (converted), instructor, and available\_slots.

### POST /book

Books a class for a user. Validates input using Pydantic and Swagger model definition. Prevents overbooking and duplicates.

Example:

| curl -X POST http://127.0.0.1:5000/book -H "Content-Type: application/json" -d '{  "class\_id": 1,  "client\_name": "Manoj",  "client\_email": "manoj@gmail.com" }' |
| --- |

Success Response: { "message": "Booking successful" }

Error Responses:

• 400 - No slots available  
• 404 - Class not found  
• 409 - Already booked  
• 422 - Invalid fields

### GET /bookings?email=...

Returns all bookings for a given client email.

Example:

| curl http://127.0.0.1:5000/bookings?email=manoj@gmail.com |
| --- |

If found: list of booking objects.  
If not found: { "message": "No bookings found for this email." }

## 📘 Swagger API Documentation

Visit http://127.0.0.1:5000/docs for live API documentation powered by Flask-RESTX Swagger UI.

You can view model definitions, try requests interactively, and see response codes and formats.

## 🪵 Logging Structure

All API activity is logged to two files under /logs:

• info.log – For successful operations and activity tracking

• error.log – For all raised exceptions and validation errors

Log format includes timestamp, level, message, filename, and line number.

## ✅ Final Remarks

This API is cleanly structured, modular, and easy to maintain. All important features including input validation, logging, documentation, and test coverage are implemented as per good practices.

## 🚀 Further Enhancements for Production-Grade deployment

* Setting up a docker container for production grade deployment.
* Adding Authentication & Authorization for enhanced security
* Adding Environment Configuration such as .env for secrets and DB configs
* Database Migrations using Alembic or Flask-Migrate for safe schema changes over time.