



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

Отчёт

по лабораторной работе № 7,8
по дисциплине «Программирование на языке Python»

Тема: «FastAPI, разработка API приложение»

Выполнил: Земляков А.А.,
студент группы ИУ8-13М

Проверил: Зотов М.

г. Москва, 2025 г.

1. ЦЕЛЬ РАБОТЫ

Реализовать API для функционала приложения по варианту

2. ПОСТАНОВКА ЗАДАЧИ

Для выполнения лабораторной работы необходимо:

- API использует асинхронное подключение к PostgreSQL через asyncpg
- API строит ORM-модели на SQLAlchemy 2.0
- API применяет валидацию и сериализацию через Pydantic (v2)
- REST API доступен через FastAPI
- API запускается через uvicorn (например: `uvicorn main:app --reload`),

либо поднимается в докер контейнере

3. ХОД РАБОТЫ

Для выполнения лабораторной работы используется среда программирования PyCharm, а также система контроля версий git.

Ход работы:

В задаче было необходимо создать API для резервирования переговоров в офисе на определенное время. Основные сущности — room и booking, основные эндпоинты:

- создание/удаление бронирования
- проверка доступности комнаты на время
- список бронирований пользователя

Для выполнения лабораторной работы был написан код, который реализует необходимое API. На рисунке 1 изображена документация, автоматически сгенерированная по всему API

комнаты		^
POST	/rooms/add_room	Добавление новой комнаты
GET	/rooms/get_rooms	Получение комнат
бронирования		^
POST	/booking/add_booking	Добавление бронирования
GET	/booking/get_users_booking	Получение бронирований пользователя
DELETE	/booking/delete_booking	Удаление бронирования пользователя
GET	/booking/check_room	Проверка доступности бронирования комнаты на время

Рисунок 1 — Документация API

На рисунках 2-5 изображена более подробная документация по всем необходимым в задаче ручкам.

POST

/booking/add_booking

Добавление бронирования

Parameters

Try it out

Name	Description
room_id required integer (query)	<input type="text" value="room_id"/>
user_name required string (query)	<input type="text" value="user_name"/>
start_time string(\$date-time) (query)	<input type="text" value="start_time"/>
end_time string(\$date-time) (query)	<input type="text" value="end_time"/>

Responses

Code	Description	Links
200	Successful Response <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div> <pre>{}</pre> </div>	No links
422	Validation Error <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div> <pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</pre> </div>	No links

Рисунок 2 - Документация по добавлению бронирования

DELETE

/booking/delete_booking

Удаление бронирования пользователя

Parameters

Try it out

Name	Description
booking_id required integer (query)	<input type="text" value="booking_id"/>
user_name required string (query)	<input type="text" value="user_name"/>

Responses

Code	Description	Links
200	Successful Response <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div> <pre>"string"</pre> </div>	No links
422	Validation Error <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div> <pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</pre> </div>	No links

Рисунок 3 — Документация по удалению бронирования

GET /booking/get_users_booking Получение бронирований пользователя

Try it out

Name	Description
user_name * required string (query)	<input type="text" value="user_name"/>

Responses

Code	Description	Links
200	Successful Response	No links
Media type <input type="text" value="application/json"/> Controls Accept header.		
Example Value Schema <pre>{ "room_id": 0, "user_name": "string", "start_time": "2025-12-18T07:09:58.658Z", "end_time": "2025-12-18T07:09:58.658Z", "id": 0 }</pre>		
422	Validation Error	No links
Media type <input type="text" value="application/json"/> Controls Accept header.		
Example Value Schema <pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</pre>		

Рисунок 4 — Документация по получению бронирований пользователя

GET /booking/check_room Проверка доступности бронирования комнаты на время

Try it out

Name	Description
id * required integer (query)	<input type="text" value="id"/>
start_time string(\$date-time) (query)	<input type="text" value="start_time"/>
end_time string(\$date-time) (query)	<input type="text" value="end_time"/>

Responses

Code	Description	Links
200	Successful Response	No links
Media type <input type="text" value="application/json"/> Controls Accept header.		
Example Value Schema <pre>true</pre>		
422	Validation Error	No links
Media type <input type="text" value="application/json"/> Controls Accept header.		
Example Value Schema <pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</pre>		

Рисунок 5 — Документация по проверки доступности комнтаы

4. ЗАКЛЮЧЕНИЕ

В ходе работы был реализован API для функционала приложения для резервирования переговоров в офисе на определенное время.