



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное 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

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01

О Т Ч Е Т

по лабораторной работе № 8

Название: Организация клиент-серверного взаимодействия между
Golang и PostgreSQL

Дисциплина: Языки интернет-программирования

Студент

ИУ6-31Б

(Группа)

(Подпись, дата)

Н.Е. Мамаев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

Цель работы: получение первичных навыков в организации долгосрочного хранения данных с использованием PostgreSQL и Golang.

Задание.

Перекопировать код сервисов, полученный в ходе выполнения 6-й лабораторной работы, в соответствующие поддиректории в директории cmd (кроме кода сервиса hello, т.к. он уже реализован в качестве примера). Доработать сервисы таким образом, чтобы они использовали для хранения данных СУБД PostgreSQL. Каждый сервис должен как добавлять новые данные в БД (insert/update), так и доставать их для предоставления пользователю (select).

1. Создаем БД lw8_web и в не 3 таблицы.

На рисунке 1 изображен скрипт для создания трех таблиц:

```
22  CREATE TABLE hello (  
23      id_hello SERIAL PRIMARY KEY,  
24      name_hello VARCHAR(255)  
25  )  
26  CREATE TABLE query (  
27      id_number SERIAL PRIMARY KEY,  
28      record VARCHAR(255)  
29  )  
30  CREATE TABLE counter (  
31      id_number SERIAL PRIMARY KEY,  
32      number INT  
33  )  
34  INSERT INTO counter (number)  
35  VALUES(0);
```

Рисунок 1. Создание таблиц.

2. Работа микросервиса Counter.

Примеры работы с микросервисом через Postman.

На рисунке 2 изображен результат возврата счетчика, т.е. Get-запроса:

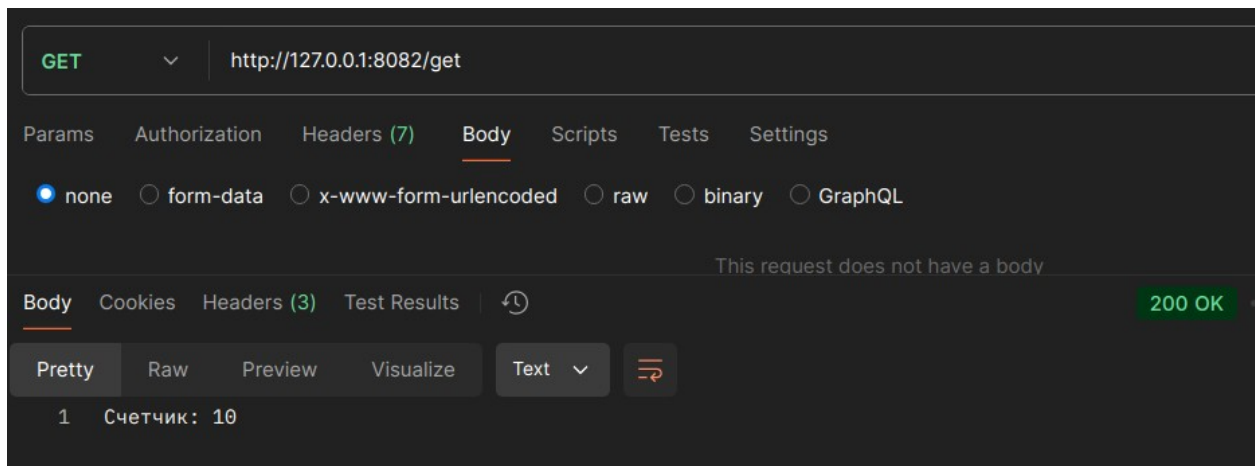


Рисунок 2. Get-запрос, вывод счетчика.

На рисунке 3 изображен результат изменения счетчика, т. е. Post-запроса:

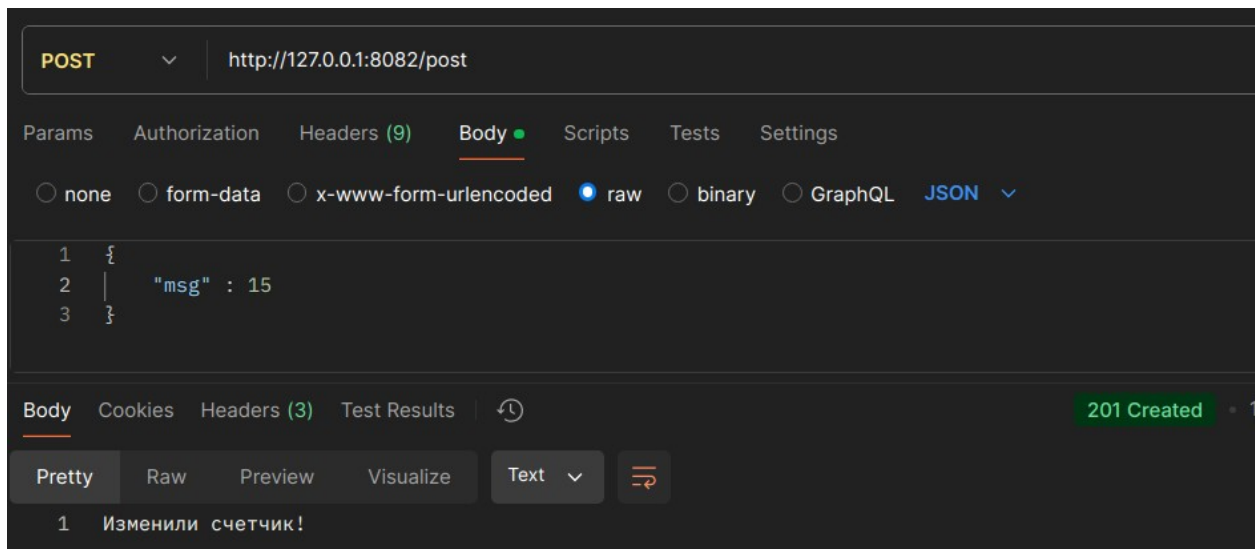


Рисунок 3. Post-запрос, изменение счетчика.

На рисунке 4 изображен результат возврата счетчика, т. е. Get-запроса после изменения:

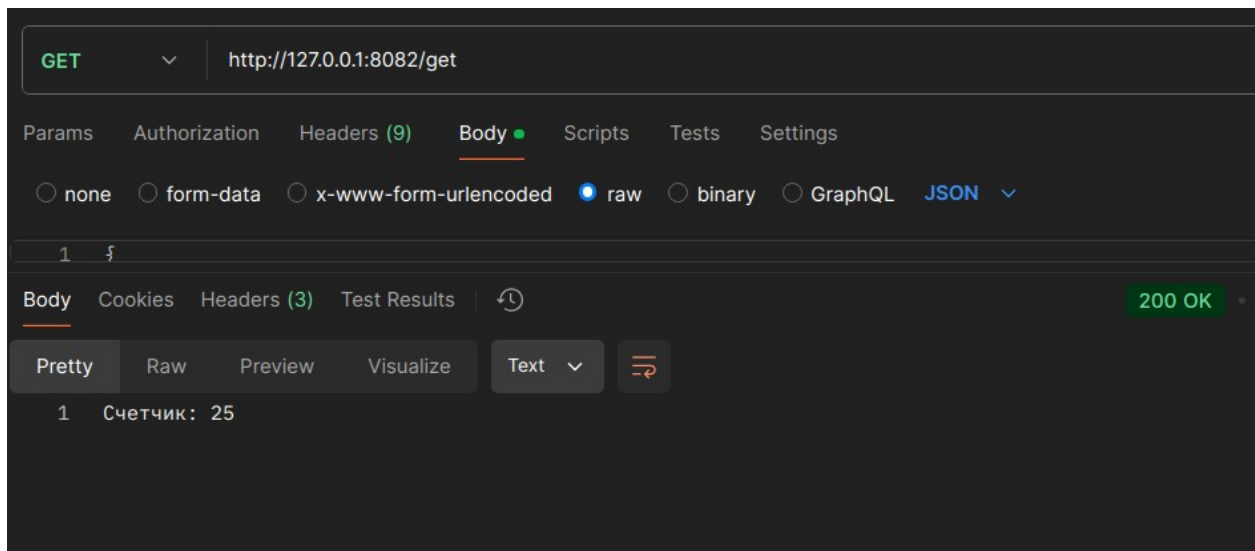


Рисунок 4. Get-запрос, демонстрация изменения счетчика.

На рисунке 5 продемонстрирована таблица counter:

16 `SELECT * FROM counter;`

17

Data Output Сообщения Notifications

	id_number [PK] integer	number integer
1	1	25

Рисунок 5. Таблица counter.

3. Работа микросервиса Query.

Примеры работы с микросервисом через Postman.

На рисунке 6 изображен результат get-запроса со значением параметра name, которое содержится в таблице:

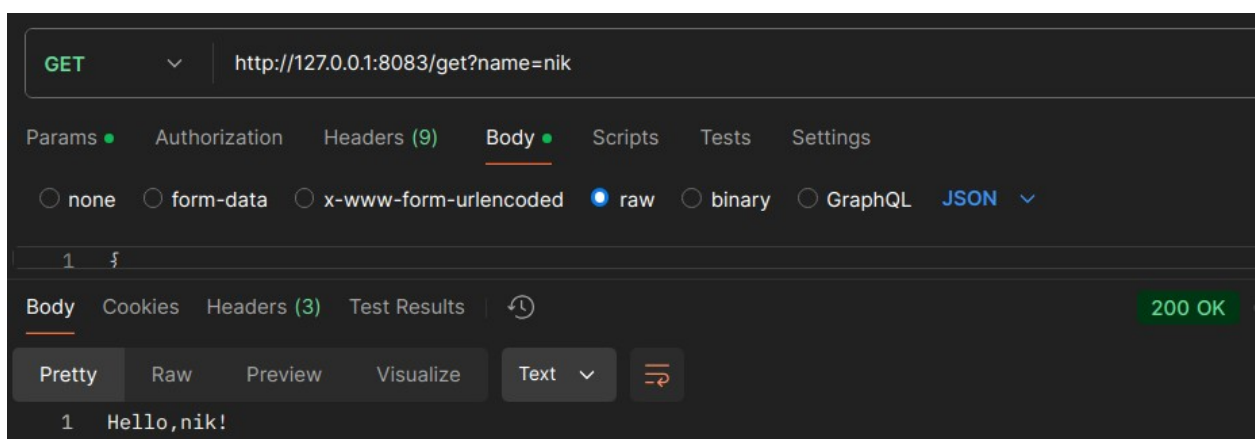


Рисунок 6. Get-запрос с существующими данными.

На рисунке 7 изображен результат get-запроса со значением параметра name, который не содержится в таблице:

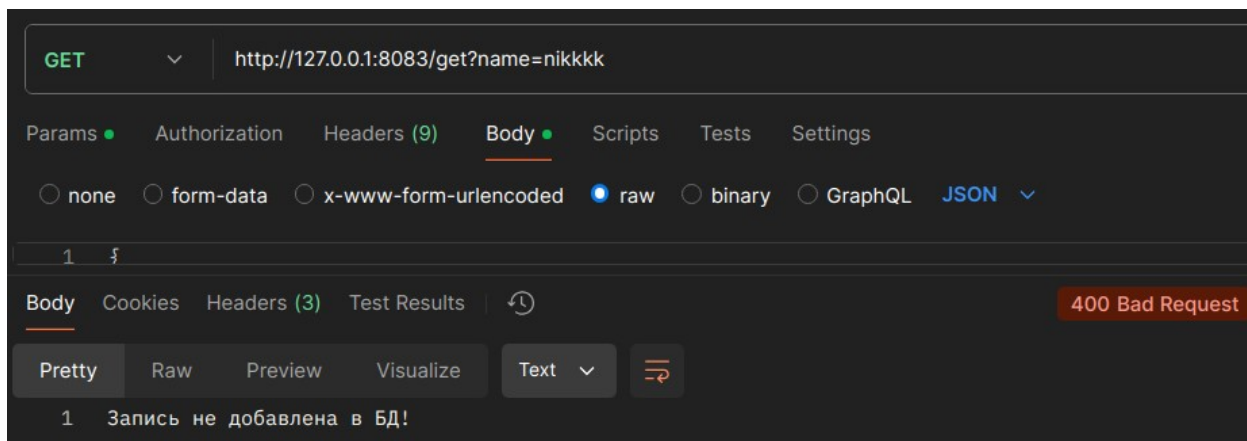


Рисунок 7. Get-запрос с несуществующими данными.

На рисунке 8 изображен результат get-запроса без значения параметра name:

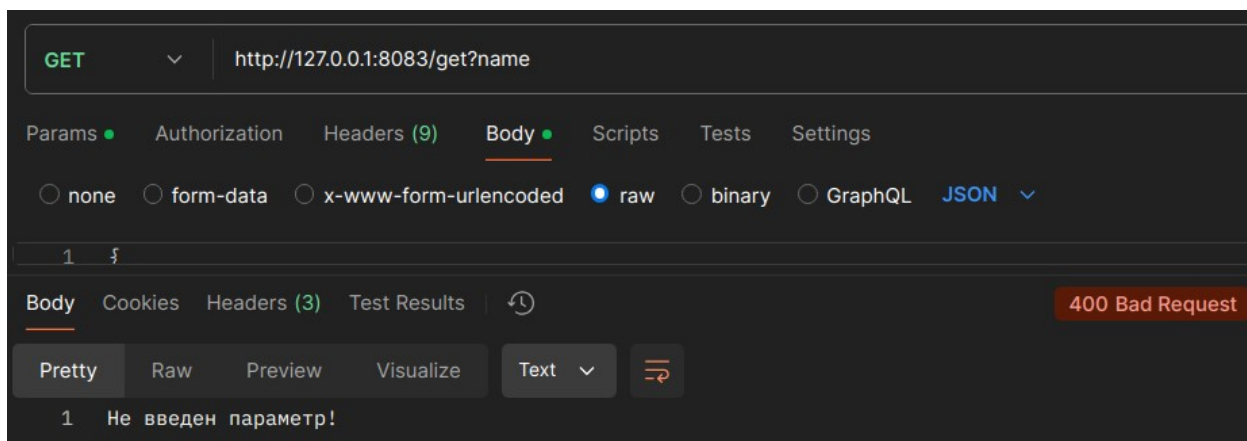


Рисунок 8. Get-запрос без данных.

На рисунке 9 изображен результат post-запроса со значением параметра name, который содержится в таблице:

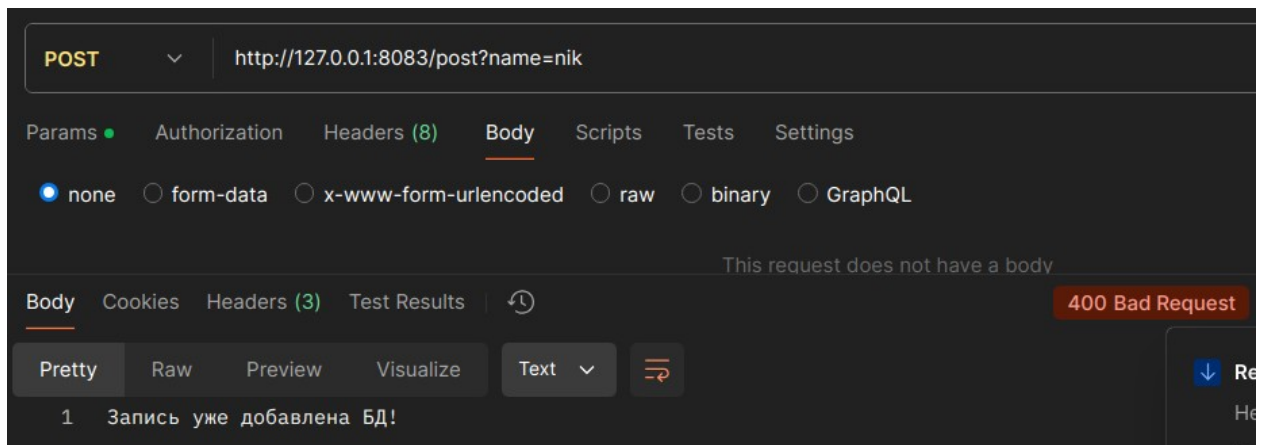


Рисунок 9. Post-запрос с существующими данными.

На рисунке 10 изображен результат post-запроса со значением параметра name, который не содержится в таблице:

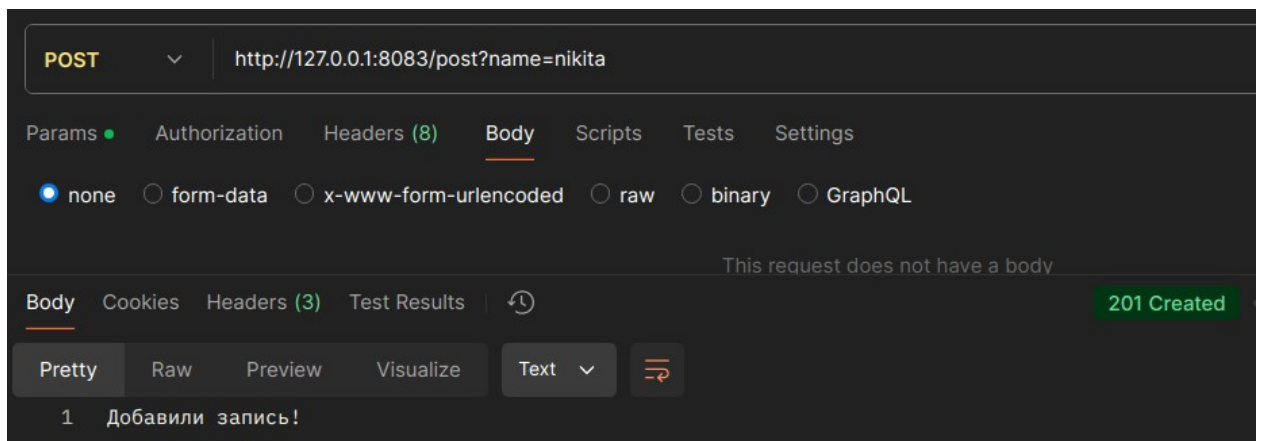


Рисунок 10. Post-запрос с несуществующими данными.

На рисунке 11 изображен результат post-запроса без значения параметра name:

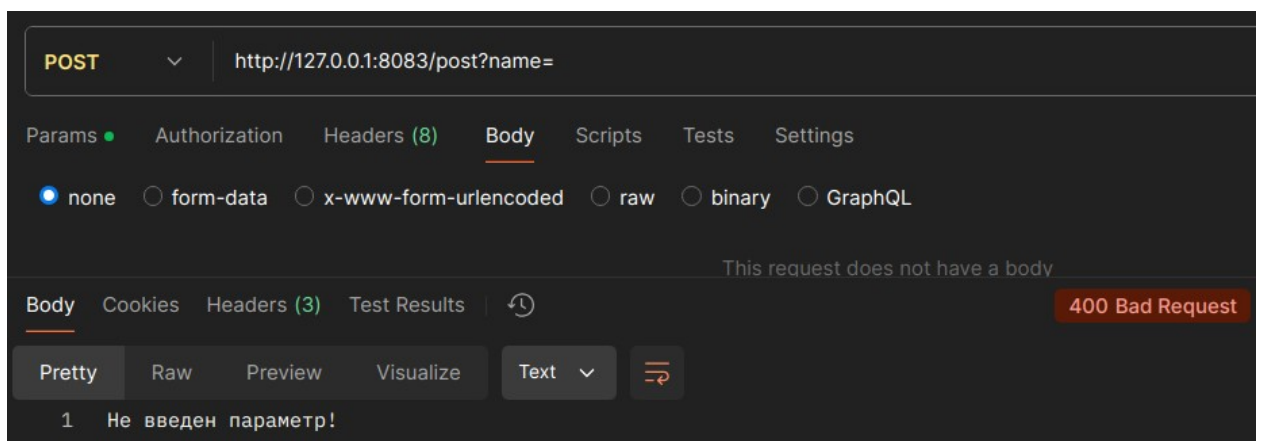
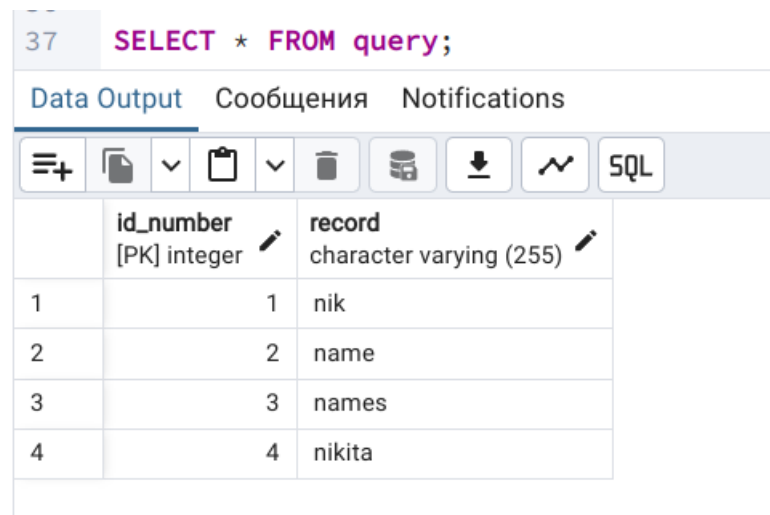


Рисунок 11. Post-запрос без данных.

На рисунке 12 продемонстрирована таблица query:



	id_number [PK] integer	record character varying (255)
1	1	nik
2	2	name
3	3	names
4	4	nikita

Рисунок 12. Таблица query.

4. Работа микросервиса hello.

На рисунке 13 изображен результат Post-запроса:

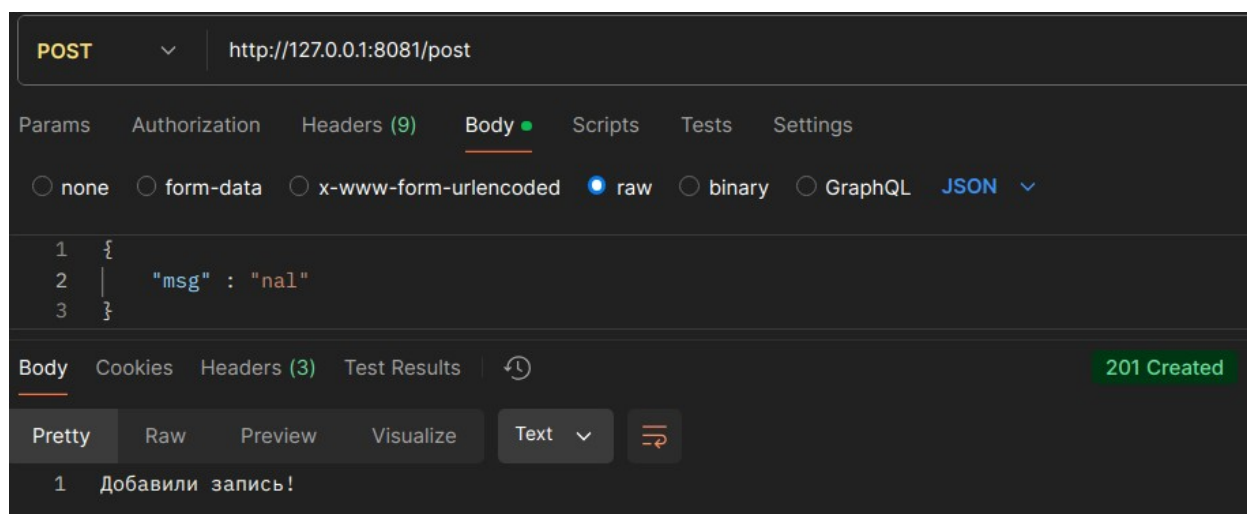


Рисунок 13. Post-запрос, добавление записи.

На рисунке 14 изображен результат Get-запроса:

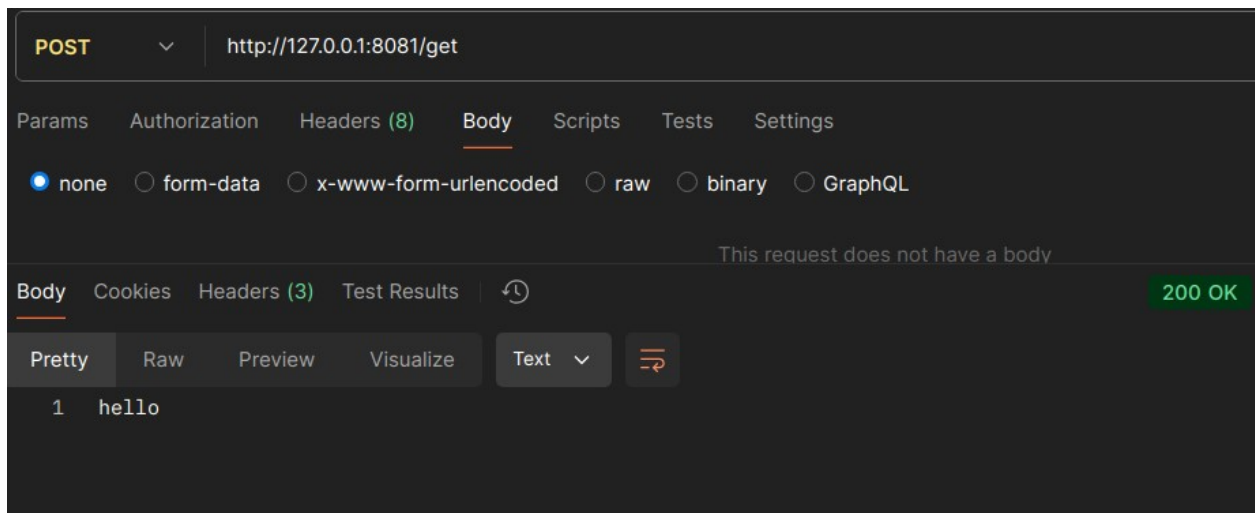


Рисунок 14. Get-запрос.

На рисунке 15 изображена таблица hello:

```
9  SELECT * FROM hello;
10
```

Data Output			Сообщения	Notifications
<div>SQL</div>				
	id_hello [PK] integer	name_hello character varying (255)		
1	1	nik		
2	2	hello		
3	3	nal		

Рисунок 15. Таблица hello.

Вывод: в ходе лабораторной работы получены навыки в организации долгосрочного хранения данных с использованием PostgreSQL и Golang.