



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 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 8

**Название: Организация клиент-серверного взаимодействия
между Golang и PostgreSQL**

Дисциплина: Основы web-программирования

Студент

ИУ6-31Б

(Группа)

Минбулатов
А.А.

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

Шульман В.Д.

(Подпись, дата)

(И.О. Фамилия)

Москва, 2024

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

Порядок выполнения:

Для успешного выполнения лабораторной работы необходимо проделать следующие шаги:

1. Установить и настроить PostgreSQL
2. Ознакомиться с теоретическими сведениями
3. Сделать форк данного репозитория в GitHub, клонировать получившуюся копию локально, создать от мастера ветку dev и переключиться на неё
4. 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)
5. 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)
6. Проверить свой код линтерами с помощью команды `make lint`
7. Сделать отчёт и поместить его в директорию docs
8. Зафиксировать изменения, сделать коммит и отправить получившееся состояние ветки dev в личный форк данного репозитория в GitHub

9. Через интерфейс GitHub создать Pull Request dev --> master

10. На защите лабораторной работы продемонстрировать открытый Pull Request. PR должен быть направлен в master ветку форка, а не исходного репозитория

Ход работы

Создаем бд под названием LAB8 с соответствующими таблицами.

```
CREATE TABLE hello(  
    id_hello    SERIAL PRIMARY KEY,  
    name_hello  VARCHAR(255)  
)  
  
CREATE TABLE query(  
    id_query    SERIAL PRIMARY KEY,  
    name_query  VARCHAR(255)  
)  
  
CREATE TABLE count(  
    id_number SERIAL PRIMARY KEY,  
    number INT  
)
```

Рисунок 1 – Создание таблиц

Пример работы микросервиса counter через postman:

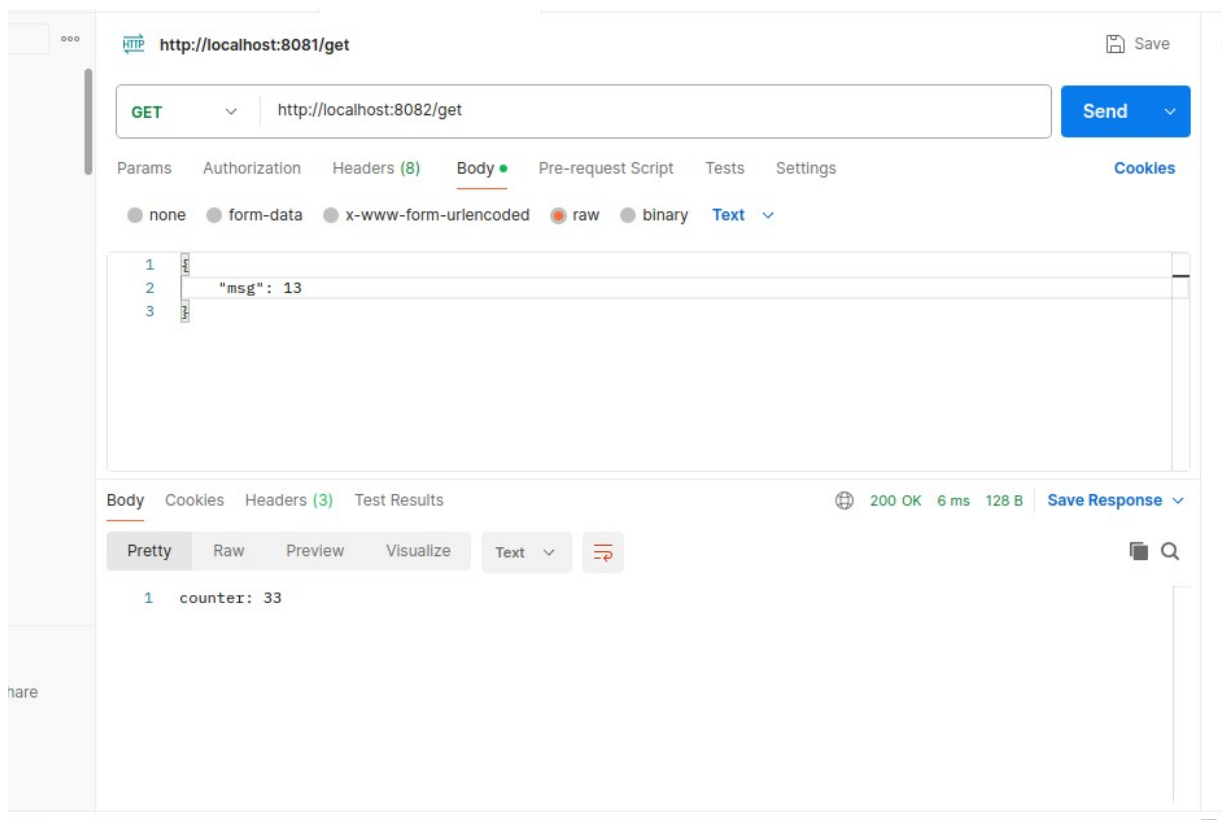


Рисунок 2 – Get запрос, вывод счетчика

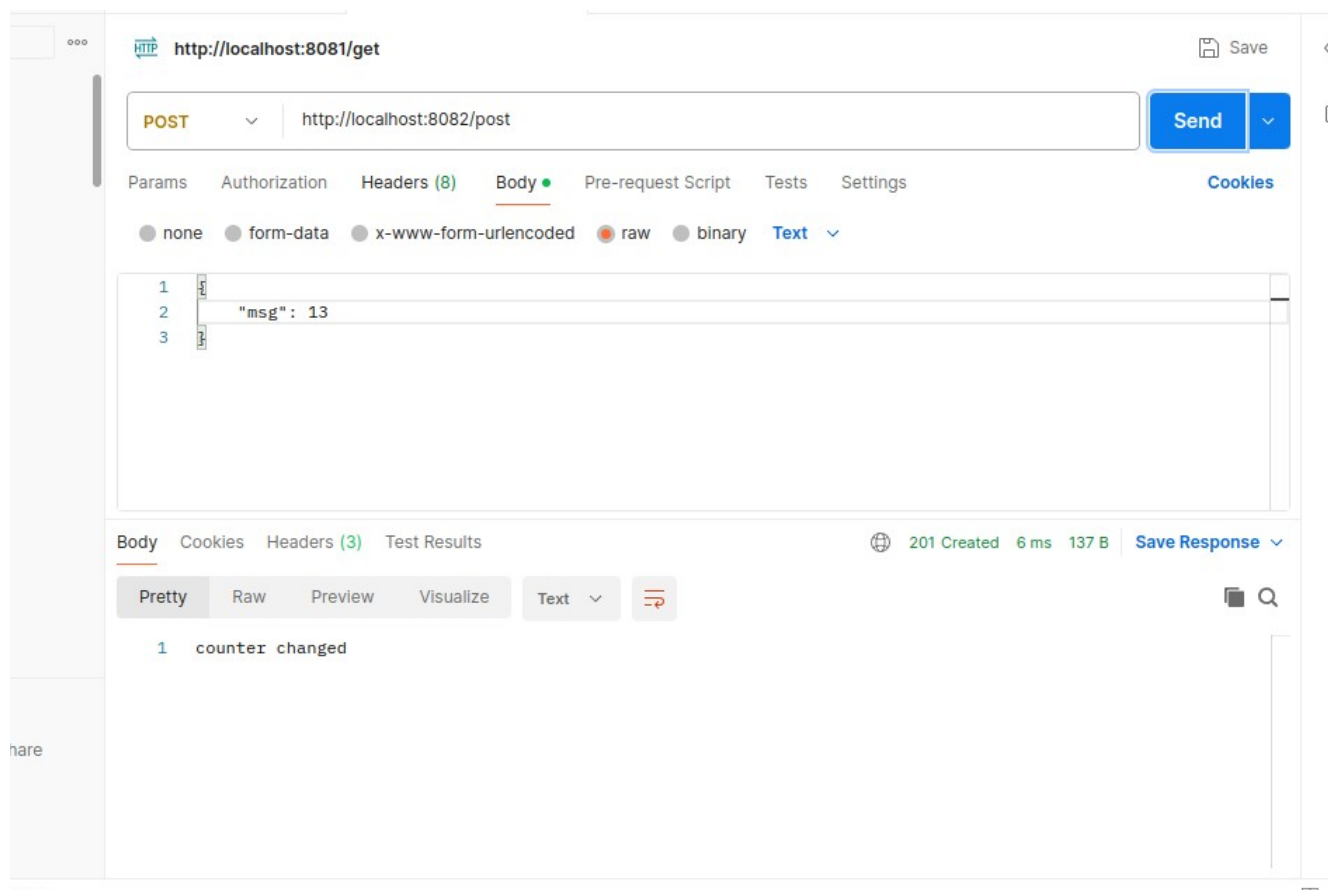


Рисунок 3 – Post запрос, изменение счетчика

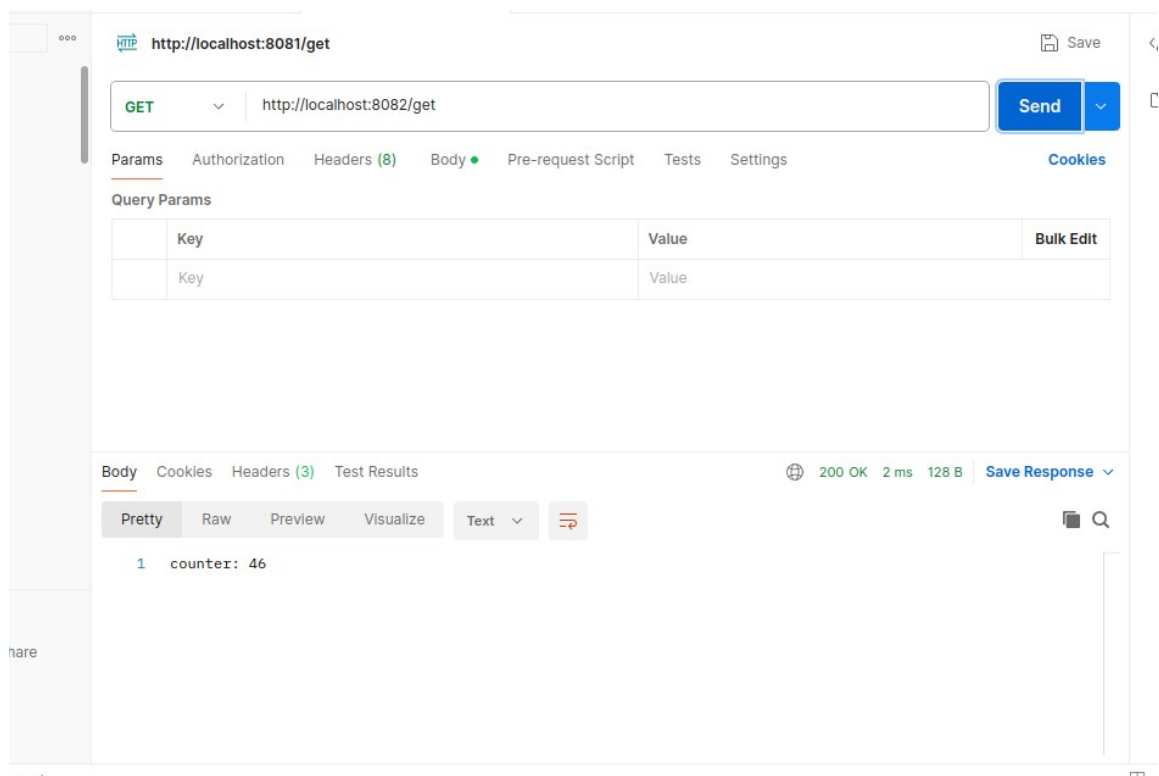


Рисунок 4 – Get запрос, результат изменения счетчика

Showing rows: 1 to 1		
	id_number [PK] integer	number integer
1	1	46

Рисунок 5 – Таблица counter

Пример работы микросервиса query через postman:

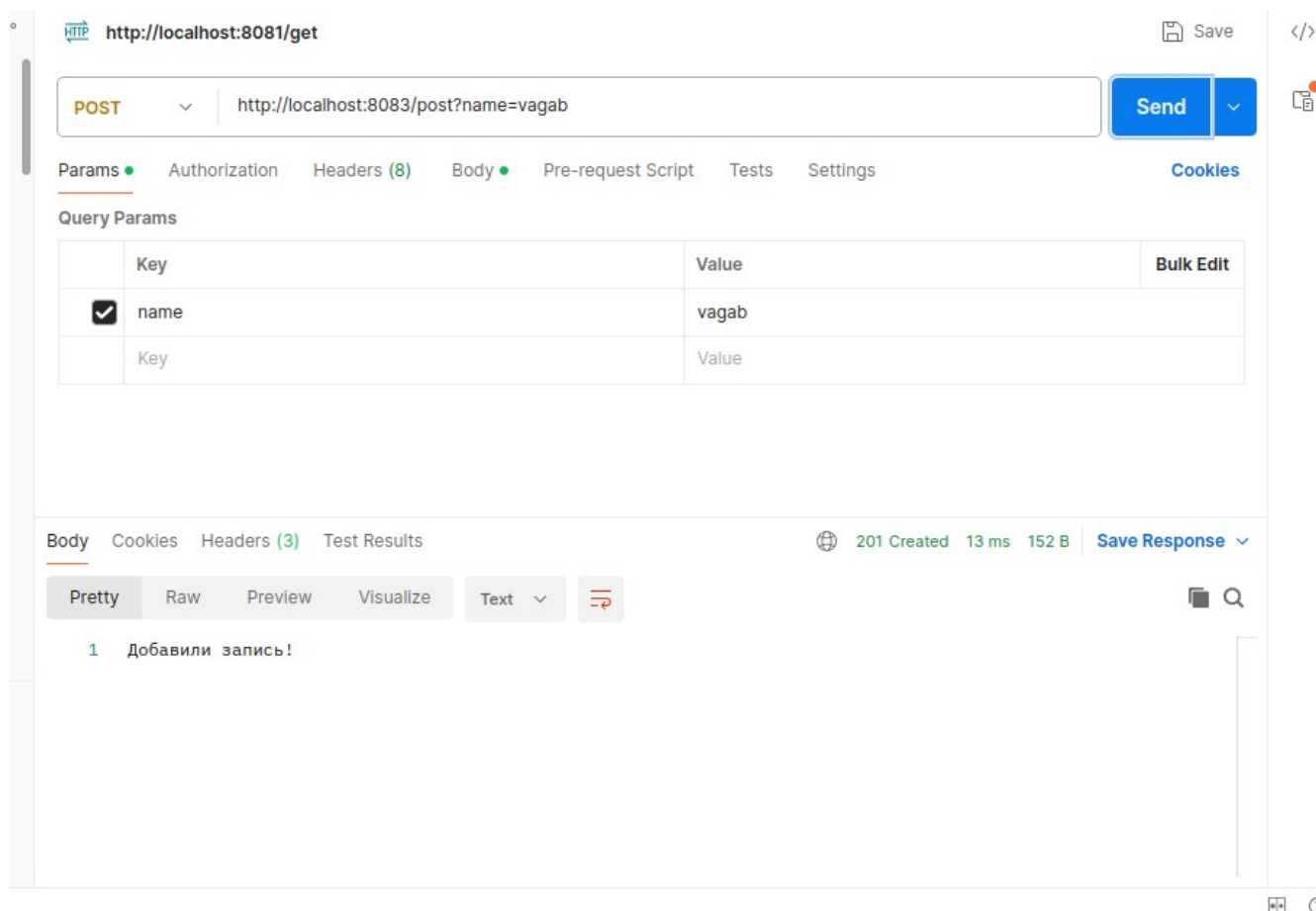


Рисунок 6 – Post запрос добавление записи

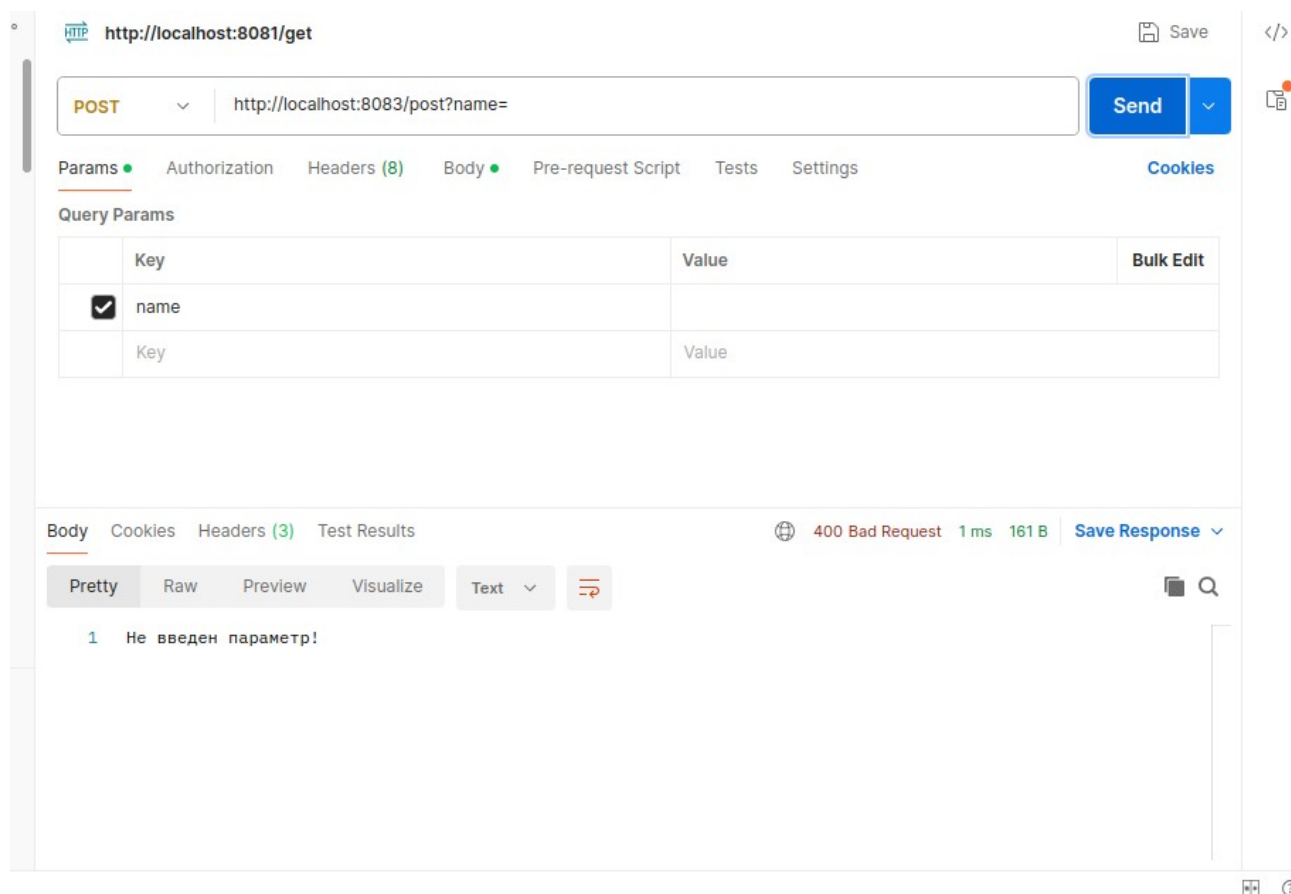


Рисунок 7 – Post запрос без параметров

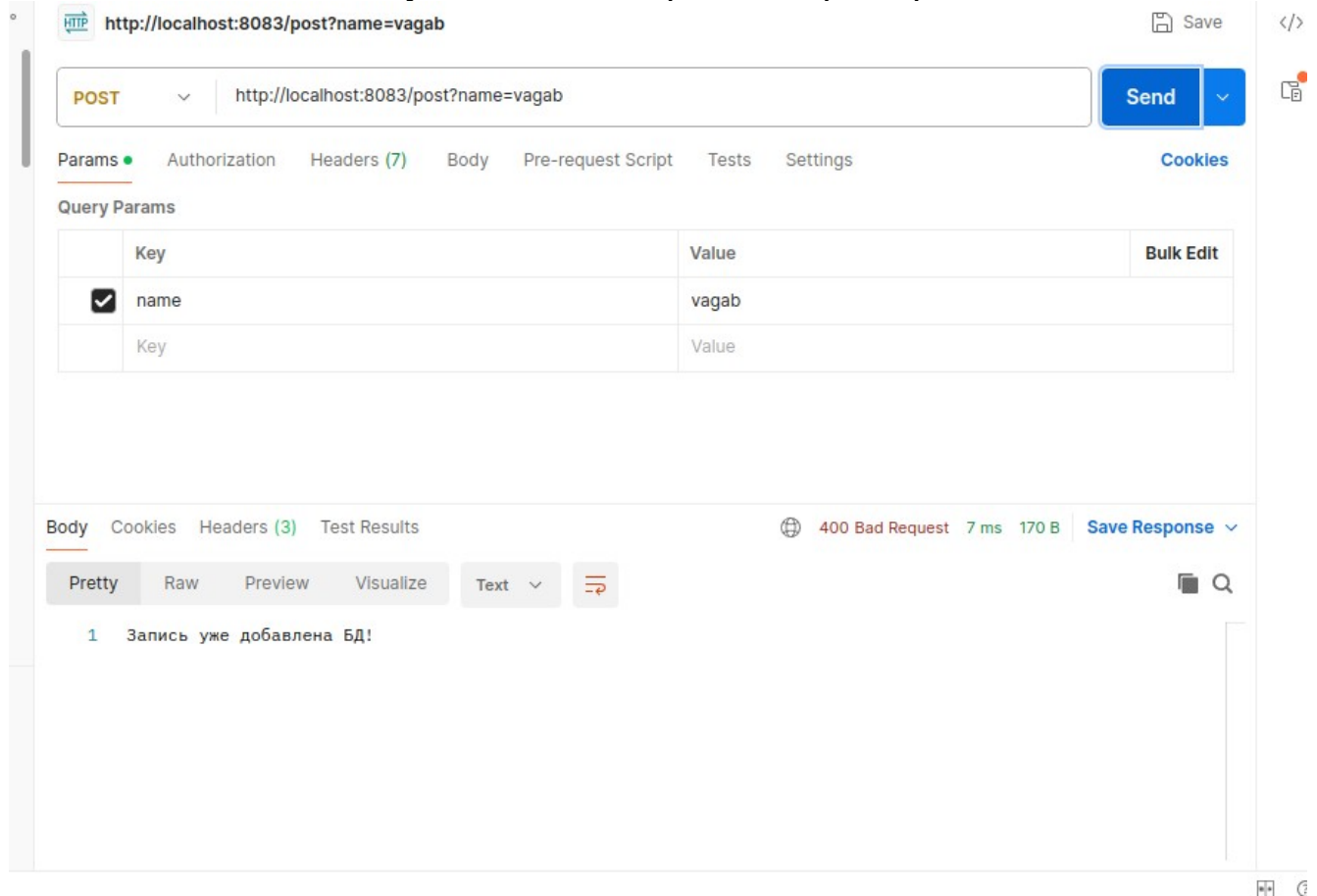


Рисунок 8 – Post запрос с существующим параметром

HTTP

http://localhost:8083/post?name=vagab

Save

</>

GET

http://localhost:8083/post?name=vagab

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	name	vagab	
	Key	Value	

body

Cookies

Headers (3)

Test Results

400 Bad Request

11 ms

170 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1 Запись уже добавлена БД!

Рисунок 9 – Get запрос с существующими данными

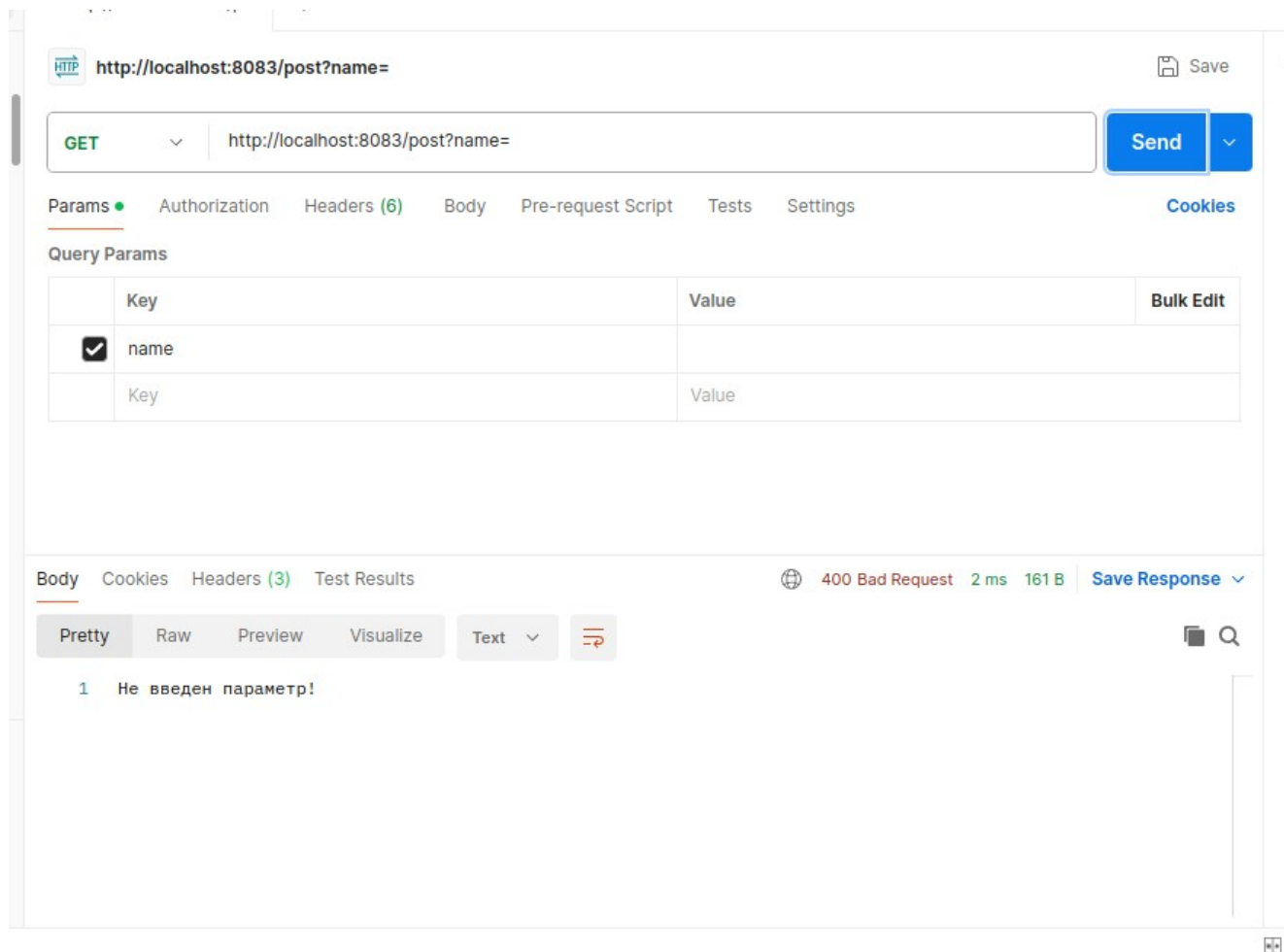


Рисунок 10 – Get запрос с несуществующими параметрами

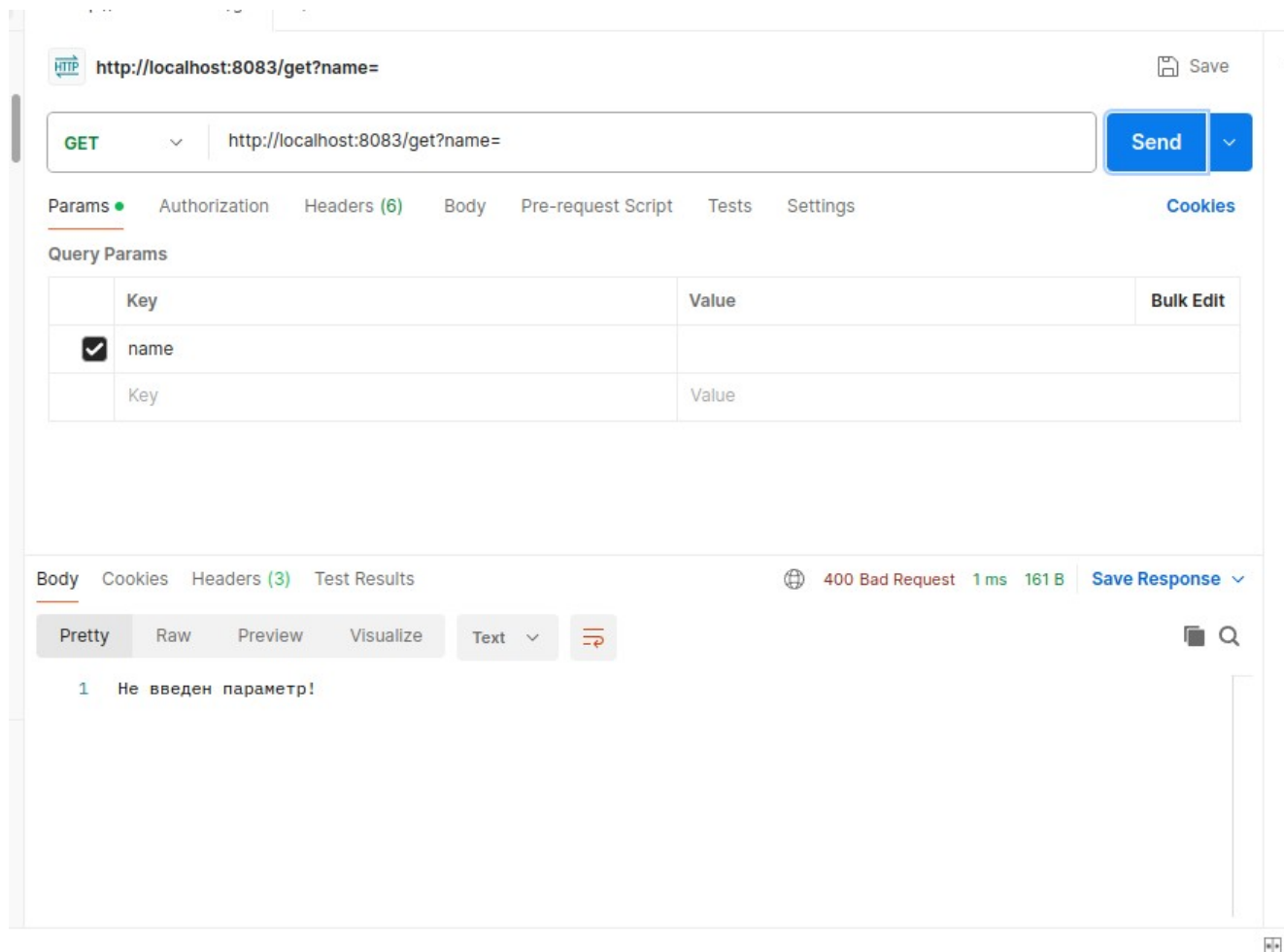


Рисунок 11 – Get запрос без параметров

	id_query [PK] integer	name_query character varying (255)
1	9	vagab
2	10	maga
3	11	abukurax

Рисунок 12 – Таблица query

Пример работы микросервиса hello через postman:

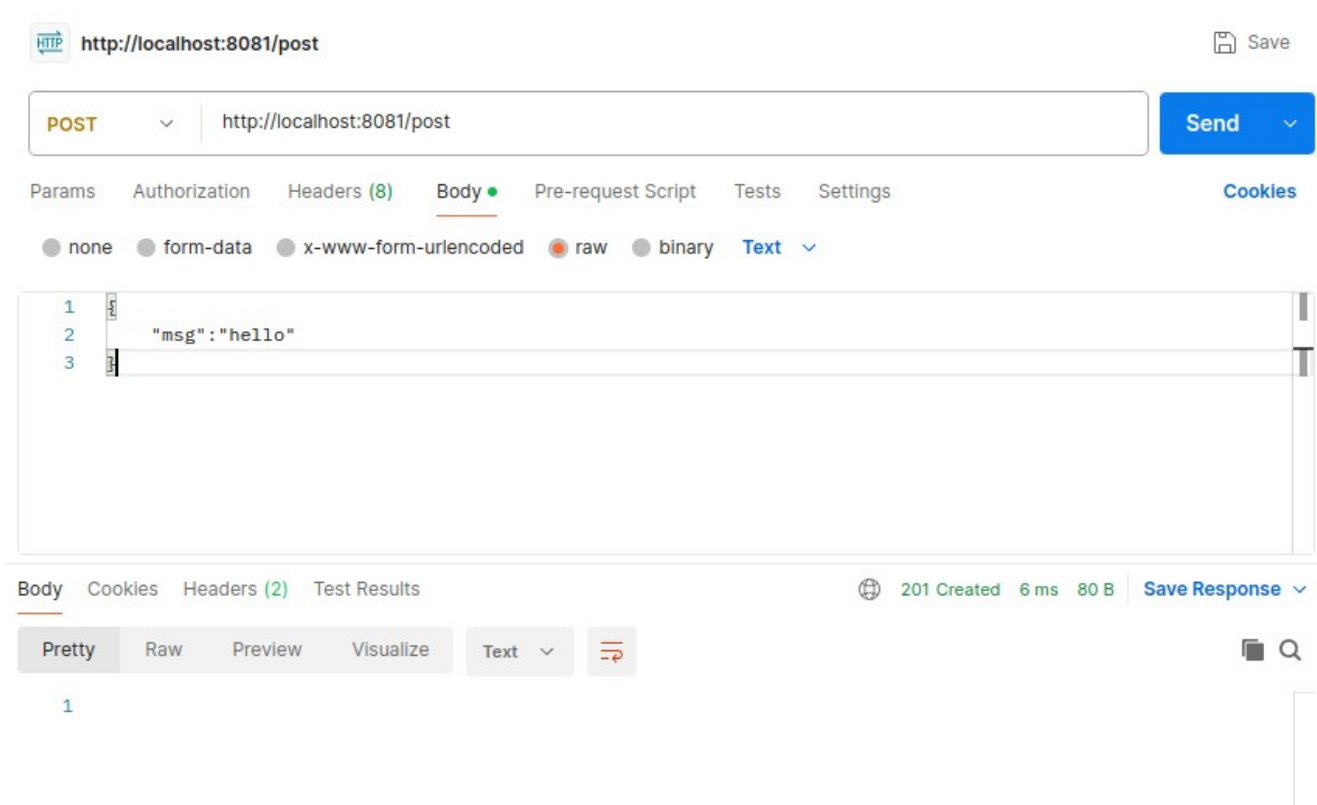


Рисунок 14 – Post запрос добавление записи

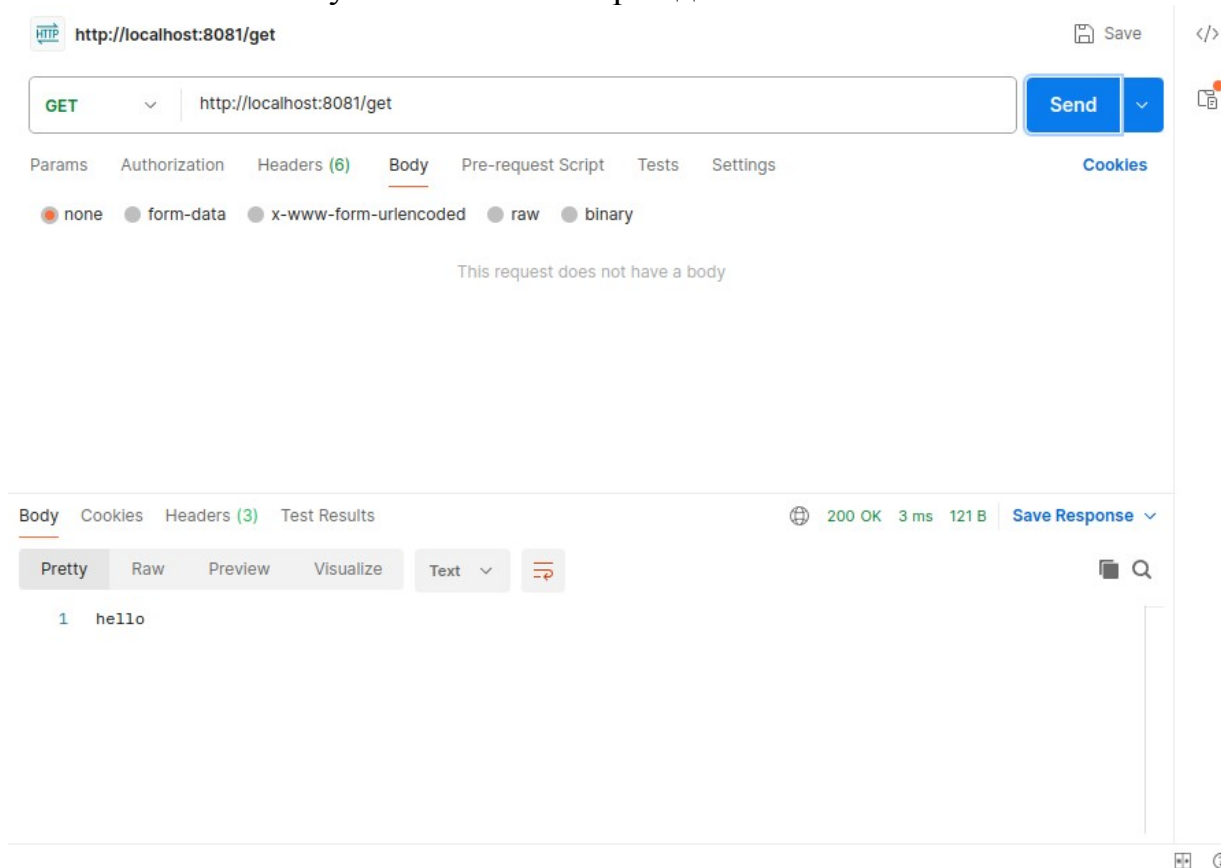


Рисунок 15 – Get запрос

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