



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: Языки-интернет программирования

Студент

ИУ6-33Б

(Группа)

(Подпись, дата)

Д.Е. Горячев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

1) Цель лабораторной работы

Получение первичных навыков в организации долгосрочного хранения данных с использованием PostgreSQL и Golang.

2) Задание

В рамках данной лабораторной работы предлагается продолжить изучение Golang и познакомиться с набором стандартных библиотек, используемых для организации клиент-серверного взаимодействия между Golang и PostgreSQL, где в роли клиента выступает сервис Golang, а в роли сервера СУБД PostgreSQL.

3) Ход работы

3.1) count

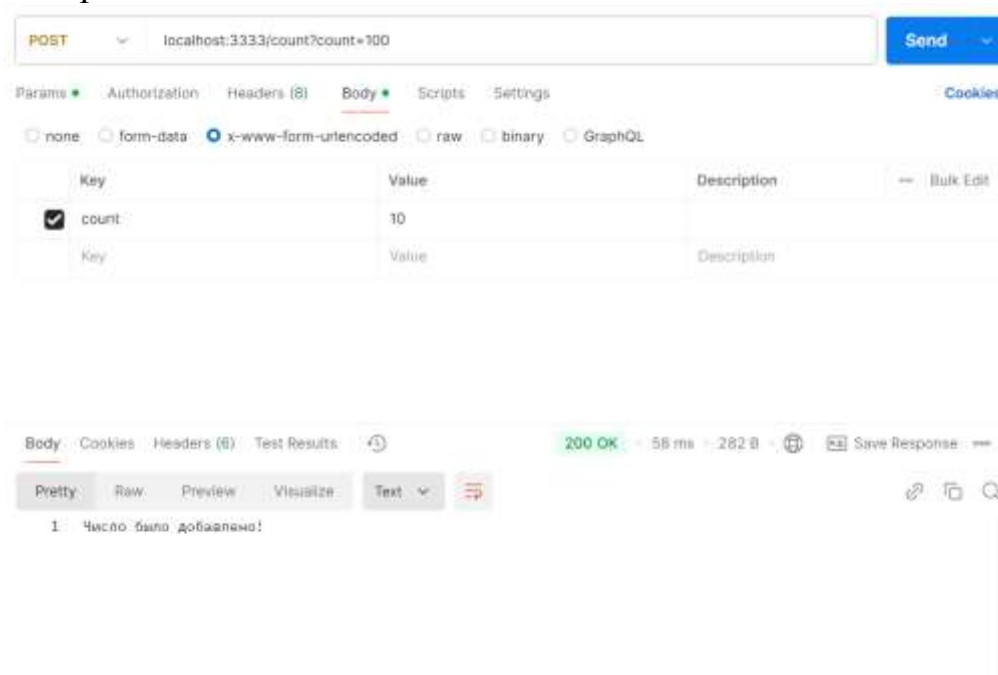
Эта программа при POST-запросе увеличивает число в таблице на параметр count, а при GET-запросе получает его и выводит на экран.



	name text	count integer
1	key1	0

Вид начальной таблицы.

Теперь добавим с помощью POST число 10.



POST localhost:3333/count?count=10

Send

Params Authorization Headers (8) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description
count	10	

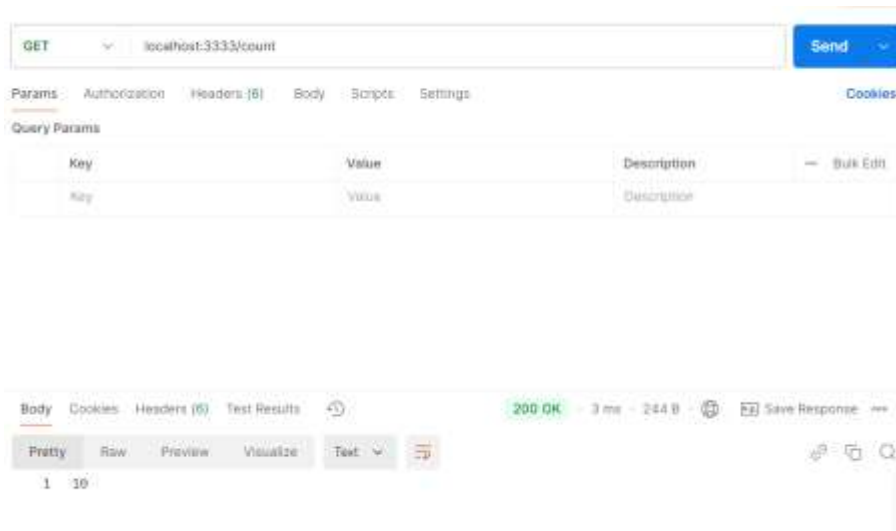
Body: Cookies Headers (8) Test Results

200 OK 58 ms 282 B Save Response

Pretty Raw Preview Visualize Text

1: Число было добавлено!

Добавили число. Теперь с помощью GET проверим добавилось ли число в таблицу.



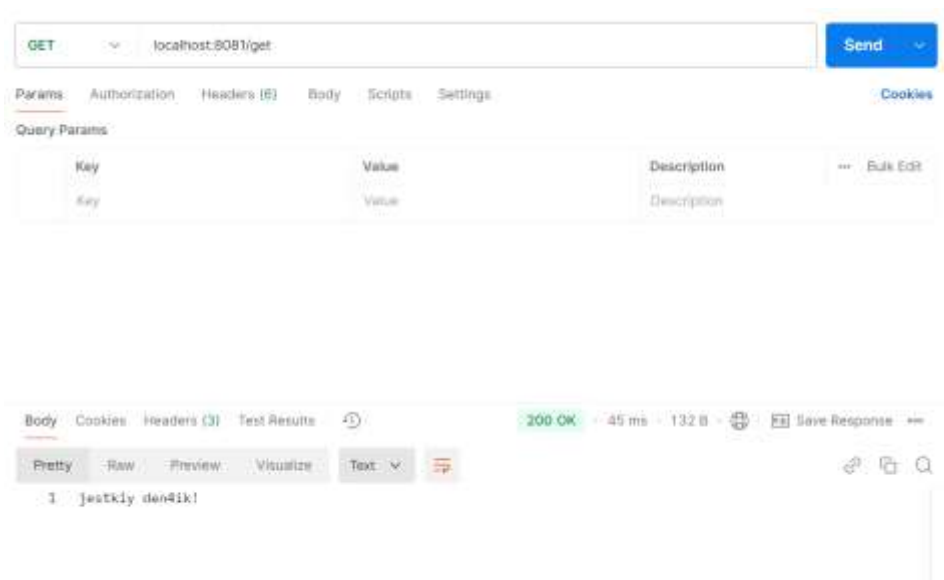
Как можем заметить число добавилось.

3.2)hello

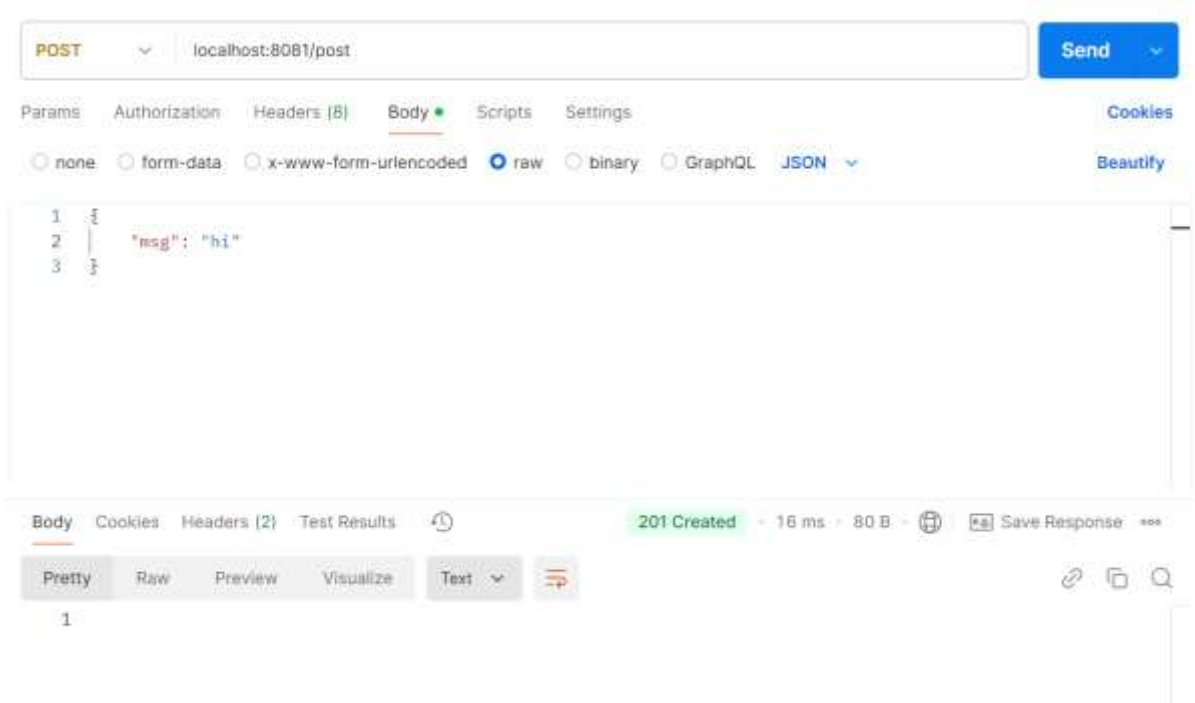
Эта программа теперь POST-запросом сохраняет поле в таблице базы данных, а GET-запросом получает случайное имя из таблицы.

	message text
1	hello!
2	oleg!
3	jestkiy den4ik!
4	dfg
5	куку
6	oleg
7	param


Вид изначальной таблицы.



Вызвали метод GET и нам вернулось слово “jestkiy den4ik!”, которое есть в изначальной таблице. Теперь с помощью метода POST добавим слово в таблицу.



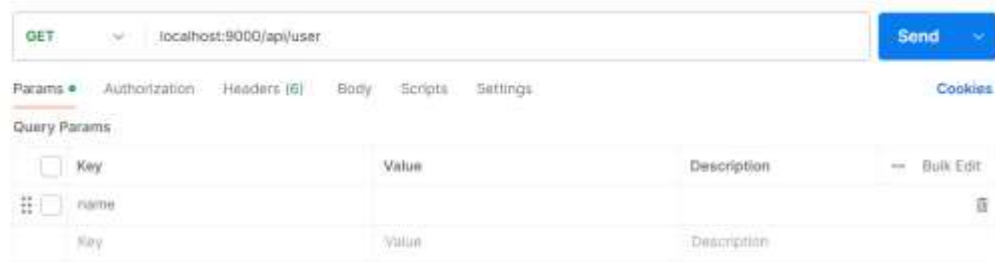
Добавили слово “hi”. Проверим появилось ли оно в изначальной таблице.

	message 
1	hello!
2	oleg!
3	jestkiy den4ik!
4	dfg
5	куку
6	oleg
7	param
8	hi

Как можем заметить слово появилось.

3.3)query

В этой программе реализована схожая с предыдущей функция, только теперь если аргумент name отсутствует, берется случайное имя из таблицы, а если указан, то выводится “Hello,{name}!” и name добавляется в таблицу, если оно там отсутствует.



Вывелось случайное имя из таблицы.



Теперь ввели имя “Dennis” и проверим добавилось ли оно в таблицу.

	message
	text
1	hello!
2	oleg!
3	jestkiy den4ik!
4	dfg
5	kyky
6	oleg
7	param
8	hi
9	Dennis

Как можем заметить – оно добавилось.

4) Вывод

Изучены стандартные библиотеки Go, используемые для организации взаимодействия между клиентом и сервером при работе с PostgreSQL в Golang.