



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)

С.В. Сонин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Цель работы:

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

Задание:

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

Результаты работы.

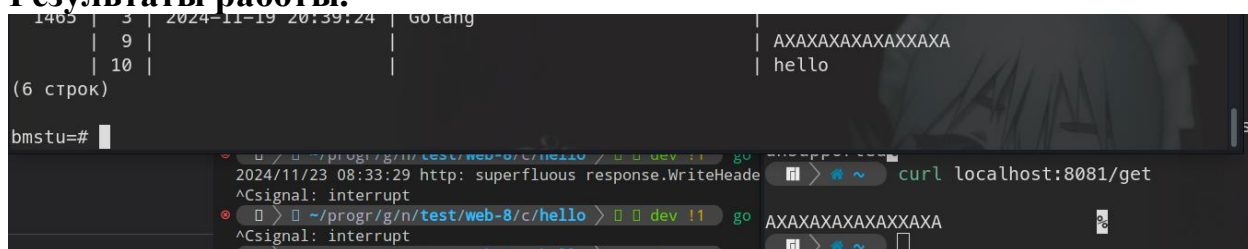


Рисунок 1 – Результат работы «hello-get».

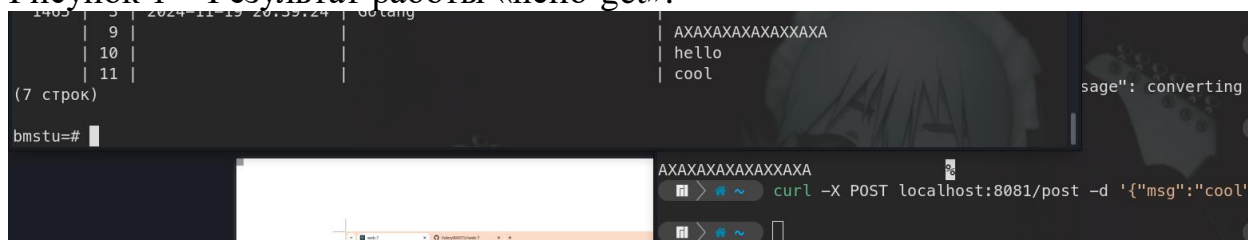


Рисунок 2 – Результат работы «hello-post».

```
zsh: no matches found: localhost:8081/api/use
hal: ~ curl "localhost:8081/api/use
Hello, Golang%
hal: ~ curl "localhost:8081/api/use
Hello, durak%
hal: interrupt
~/progr/g/n/test/web-8/cmd/query > dev ?1
hal: interrupt
```

count	id	time	name
1488	1		
0	2	2024-11-19 19:12:48	
0	3	2024-11-19 19:15:16	Golang
0	4	2024-11-19 19:15:36	durak

(4 строки)

```
bmstu=#
```

Рисунок 3 — Результат работы программы «query» 1

```
hal: ~ curl "localhost:8081/api/user?name=durak"
Hello, durak . your last visit was in 2024-11-19T19:15:36Z%
hal: ~ curl "localhost:8081/api/user?name=Golang"
Hello, Golang . your last visit was in 2024-11-19T19:15:16Z%
hal: ~ cd /cmd/query
такого файла или каталога: /cmd/query
~/programming/go/new/test/web-8 > dev ?1 cd cmd/query
~/programming/go/new/test/web-8/cmd/query > dev ?1 go run main.go -address "127.123.23.1:8082"
```

count	id	time	name
1488	1		
0	2	2024-11-19 19:12:48	
0	4	2024-11-19 19:20:28	durak
0	3	2024-11-19 19:20:41	Golang

(4 строки)

Рисунок 4 — Результат работы программы «query» 2

```
hal: ~ curl localhost:8081
1465%
hal: ~ curl -X POST localhost:8081 -d '{"count":"1489"}'
hal: ~ curl localhost:8081
2954%
hal: ~
```

Рисунок 5 — Результат работы программы «count»

Вывод:

Получил первичные навыки в организации долгосрочного хранения данных с использованием PostgreSQL и Golang

Список использованных источников:

<https://tproger.ru/articles/osnovy-postgresql-dlya-nachinayushhih--ot-ustanovki-do-pervyh-zaprosov-250851>

<https://golangdocs.com/golang-postgresql-example>