



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

ИУ6-32Б

(Группа)

(Подпись, дата)

Д.М. Айдиев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

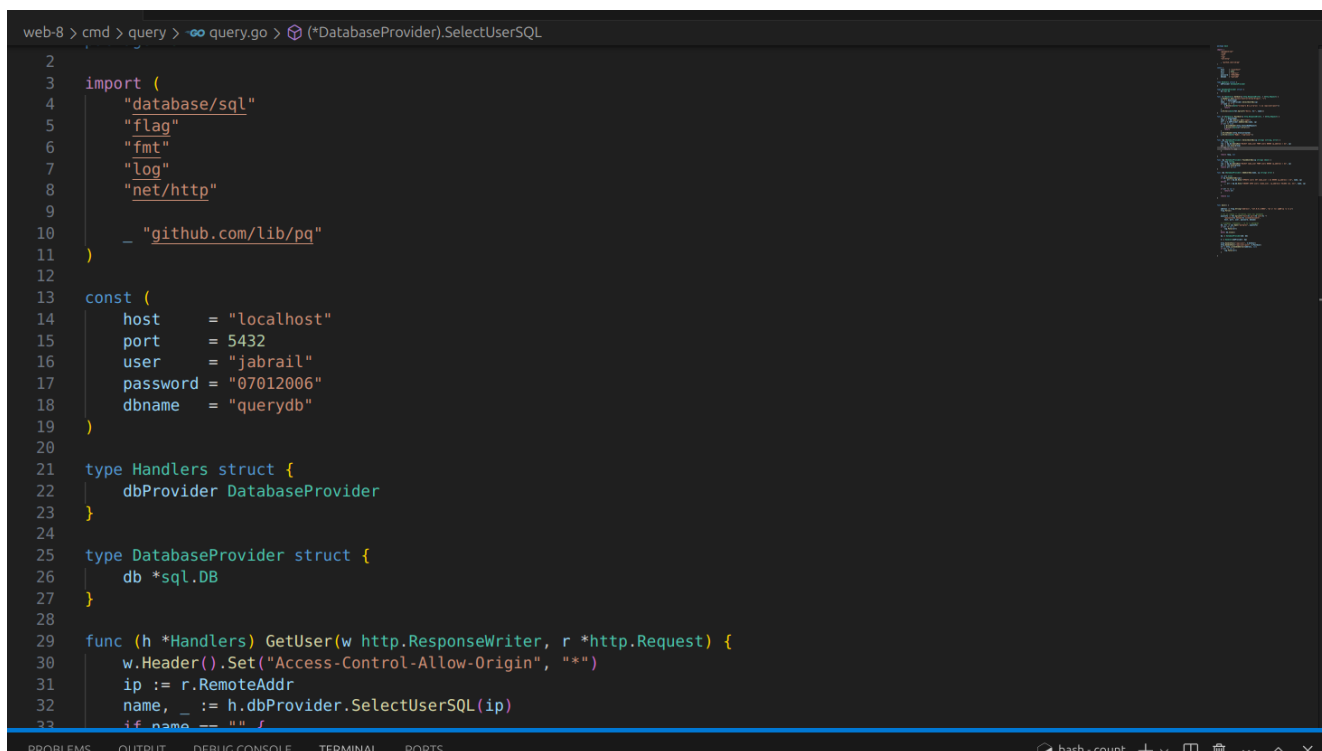
Москва, 2024

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

**Задание:** доработать сервисы с прошлой лабораторной работы таким образом, чтобы они использовали для хранения данных СУБД PostgreSQL.

## Ход работы

Доработали микросервисы (рис 1, 2)



```
web-8 > cmd > query > query.go > (*DatabaseProvider).SelectUserSQL

2
3 import (
4     "database/sql"
5     "flag"
6     "fmt"
7     "log"
8     "net/http"
9
10    _ "github.com/lib/pq"
11 )
12
13 const (
14     host      = "localhost"
15     port      = 5432
16     user      = "jabrail"
17     password  = "07012006"
18     dbname    = "querydb"
19 )
20
21 type Handlers struct {
22     dbProvider DatabaseProvider
23 }
24
25 type DatabaseProvider struct {
26     db *sql.DB
27 }
28
29 func (h *Handlers) GetUser(w http.ResponseWriter, r *http.Request) {
30     w.Header().Set("Access-Control-Allow-Origin", "")
31     ip := r.RemoteAddr
32     name, _ := h.dbProvider.SelectUserSQL(ip)
33     if name == "" {
```

рис 1

```
web-8 > cmd > count > count.go > (Handlers).handler
22
23 type Handlers struct {
24     dbProvider DatabaseProvider
25 }
26
27 type DatabaseProvider struct {
28     db *sql.DB
29 }
30
31 type myStruct struct {
32     Count int `json:"count"`
33 }
34
35
36 func (h Handlers) handler(w http.ResponseWriter, r *http.Request) {
37     w.Header().Set("Access-Control-Allow-Origin", "")
38     switch r.Method {
39     case http.MethodGet:
40         count, err := h.dbProvider.GetCount()
41         if err != nil {
42             w.WriteHeader(http.StatusBadRequest)
43             w.Write([]byte(err.Error()))
44             return
45         }
46         w.Write([]byte(strconv.Itoa(count)))
47     case http.MethodPost:
48
49         var tmp myStruct
50         r.ParseForm()
51         data, _ := io.ReadAll(r.Body)
52         if err := json.Unmarshal(data, &tmp); err != nil {
53             fmt.Println(err)
54         }
55     }
56 }
```

рис 2

Протестировали на Postman (рис 3, 4)

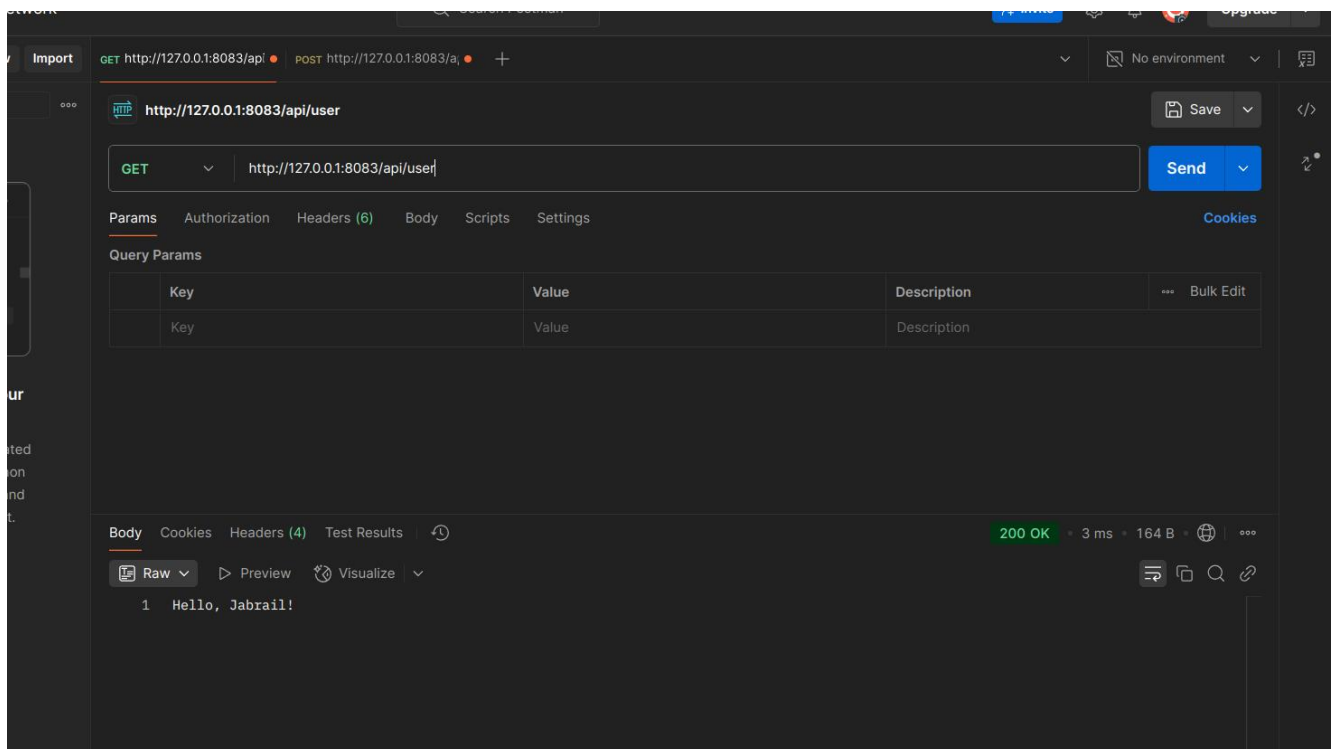


рис 3

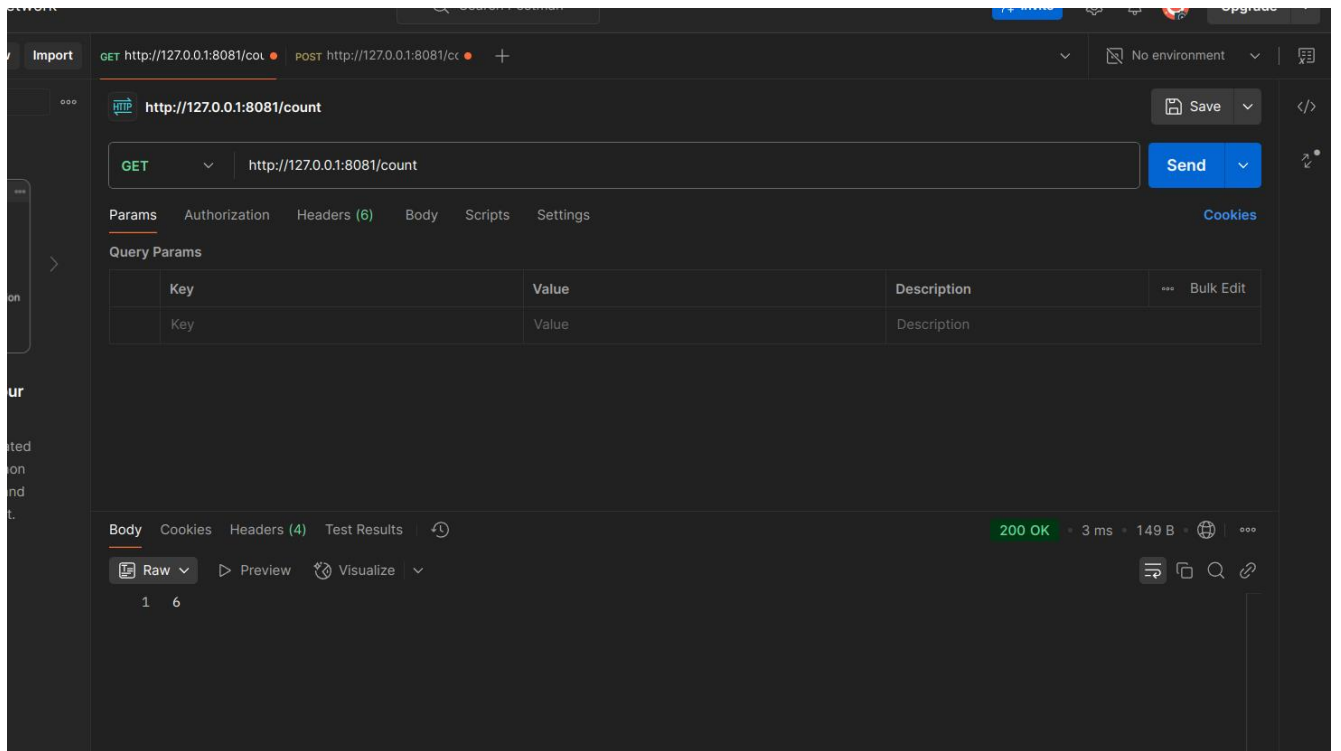


рис 4

**Заключение:** получили первичные навыки в организации долгосрочного хранения данных с использованием PostgreSQL и Golang

### Список использованных источников

<https://github.com/ValeryBMSTU/web-core/tree/master>