



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 Компьютерные системы и сети

О т ч е т

по лабораторной работе № 6

Название: Основы Back-End разработки на Golang

Дисциплина: Язык интернета-программирования

Студент

ИУ6И-31Б

(Группа)

(Подпись, дата)

Х. Чэнь

(И.О. Фамилия)

Преподаватель

В.Д. Шульман

(Подпись, дата)

(И.О. Фамилия)

Москва, 2024

## Цель работы

изучение основ сетевого взаимодействия и серверной разработки с использованием языка Golang. <https://stepik.org/course/54403/info>

## Ход работы

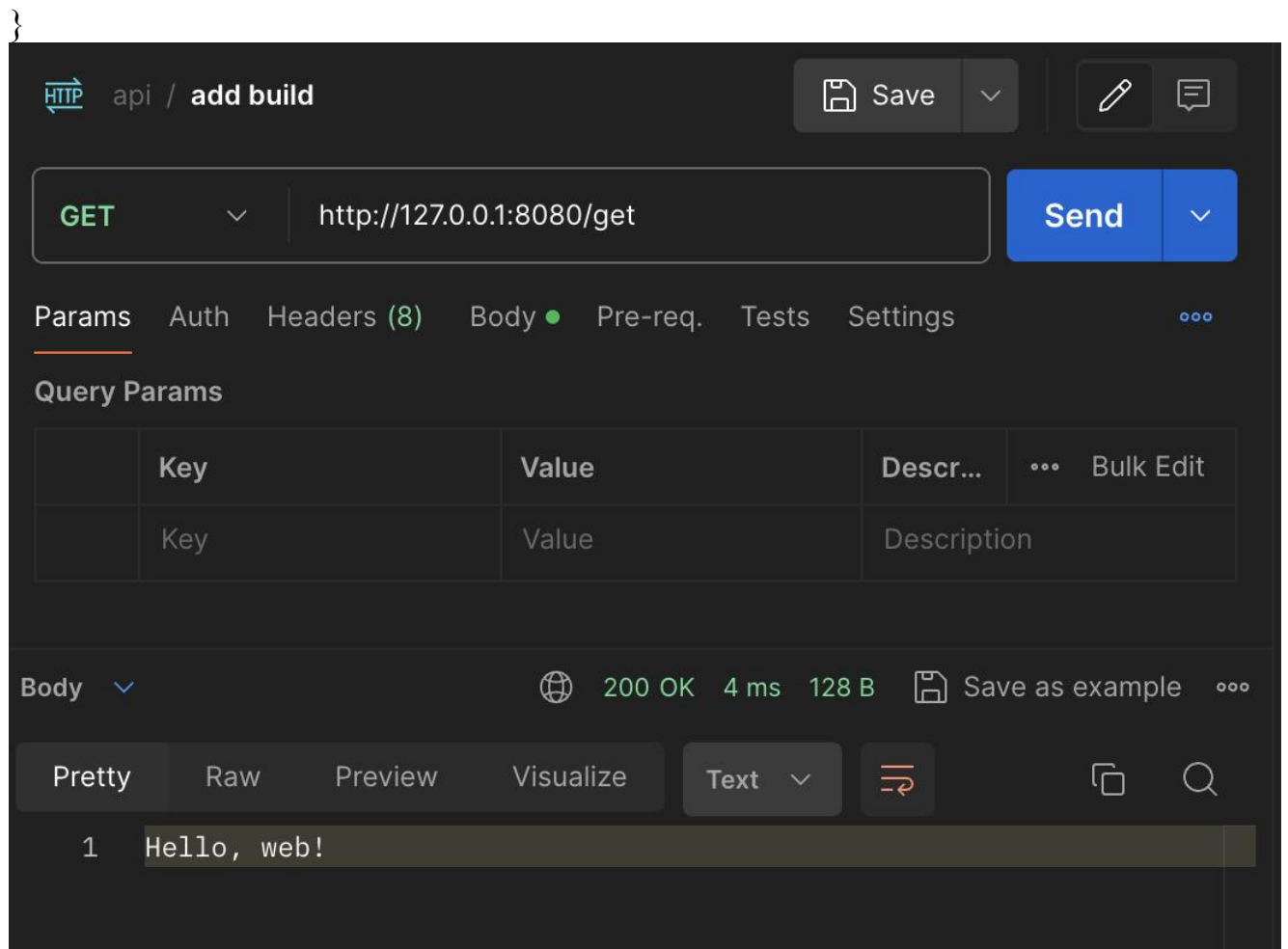
### Задание 1

#### Условие

Напишите веб сервер, который по пути /get отдает текст "Hello, web!". Порт должен быть :8080.

#### Решение

```
package main
import (
    "fmt"
    "net/http"
)
func handler(w http.ResponseWriter, r *http.Request) {
    w.Write([]byte("Hello, web!"))
}
func main() {
    http.HandleFunc("/get", handler)
    // Запускаем веб-сервер на порту 8080
    err := http.ListenAndServe(":8080", nil)
    if err != nil {
        fmt.Println("Ошибка запуска сервера:", err)
    }
}
```



(Рис. 1. result)

## Задание 2

### Условие

Напишите веб-сервер который по пути /api/user приветствует пользователя:  
 Принимает и парсит параметр name и делает ответ "Hello,<name>!"  
 Пример: /api/user?name=Golang  
 Ответ: Hello,Golang!  
 порт :9000

### Решение

```
package main
```

```
import (
    "log"
    "net/http"
)
```

```
func handler(w http.ResponseWriter, r *http.Request) {
```

```

s := r.URL.Query().Get("name")
if s == "" {
    s = "World"
}
response := "Hello, " + s + "!"
log.Println("Received request:", r.RemoteAddr, r.Method, r.URL.Path, "name=", s)
w.Write([]byte(response))
}

func main() {
    http.HandleFunc("/api/user", handler)

    err := http.ListenAndServe("127.0.0.1:9000", nil)
    if err != nil {
        log.Fatal("server wrong:", err)
    }
}

```



(Рис. 2 result)

### Задание 3

#### Условие

Напиши веб сервер (**порт :3333**) - счетчик который будет обрабатывать GET (`/count`) и POST (`/count`) запросы:

#### Решение

```
package main
```

```
import (
    "log"

```

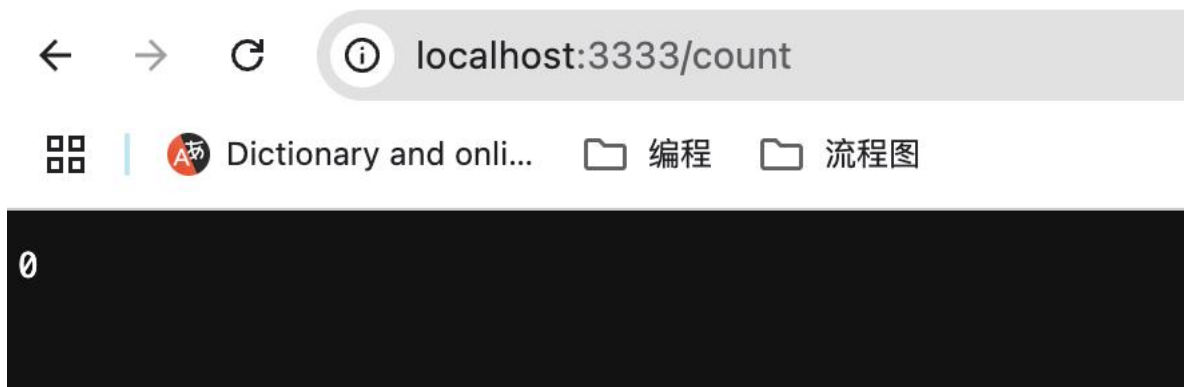
```

    "net/http"
)

func handler(w http.ResponseWriter, r *http.Request) {
    s := r.URL.Query().Get("name")
    if s == "" {
        s = "World"
    }
    response := "Hello, " + s + "!"
    log.Println("Received request:", r.RemoteAddr, r.Method, r.URL.Path, "name=", s)
    w.Write([]byte(response))
}

func main() {
    http.HandleFunc("/api/user", handler)
    err := http.ListenAndServe("127.0.0.1:9000", nil)
    if err != nil {
        log.Fatal("server wrong:", err)
    }
}

```



(Рис. 3 result)

### Вывод:

В этом упражнении вы изучите и попрактикуетесь в использовании Back-End.