



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: Основы web программирования

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

И.В.Порохницкий

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д.Шульман

(И.О. Фамилия)

Москва, 2024

Цель работы - изучение основ сетевого взаимодействия и серверной разработки с использованием языка Golang.

Задание

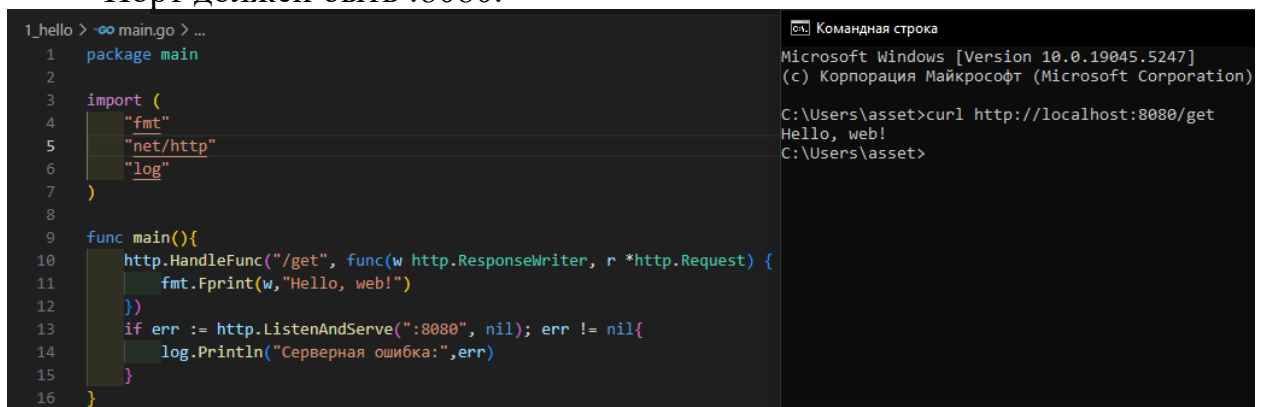
1. Ознакомиться с разделом "4. Списки, сеть и сервера" курса <https://stepik.org/course/54403/info>
2. Сделать форк данного репозитория в GitHub, клонировать получившуюся копию локально, создать от мастера ветку dev и переключиться на неё
3. Выполнить задания. Ссылки на задания можно найти в README-файлах в директории projects
4. (опционально) Проверить свой код линтерами с помощью команды `make lint`
5. Сделать отчёт и поместить его в директорию docs
6. Зафиксировать изменения, сделать коммит и отправить получившееся состояние ветки dev в личный форк данного репозитория в GitHub
7. Через интерфейс GitHub создать Pull Request dev --> master

Ход работы

1. Ознакомился с курсом
2. Сделал форк данного репозитория в GitHub, клонировал получившуюся копию локально, создал от мастера ветку dev и переключился на нее
3. Выполнил 3 задания:

Задание №1 “hello”(Рис. 1):

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



```
1_hello > main.go > ...
1 package main
2
3 import (
4     "fmt"
5     "net/http"
6     "log"
7 )
8
9 func main(){
10     http.HandleFunc("/get", func(w http.ResponseWriter, r *http.Request) {
11         fmt.Fprint(w, "Hello, web!")
12     })
13     if err := http.ListenAndServe(":8080", nil); err != nil{
14         log.Println("Серверная ошибка:", err)
15     }
16 }
```

Командная строка
Microsoft Windows [Version 10.0.19045.5247]
(c) Корпорация Майкрософт (Microsoft Corporation)
C:\Users\asset>curl http://localhost:8080/get
Hello, web!
C:\Users\asset>

Рис.1

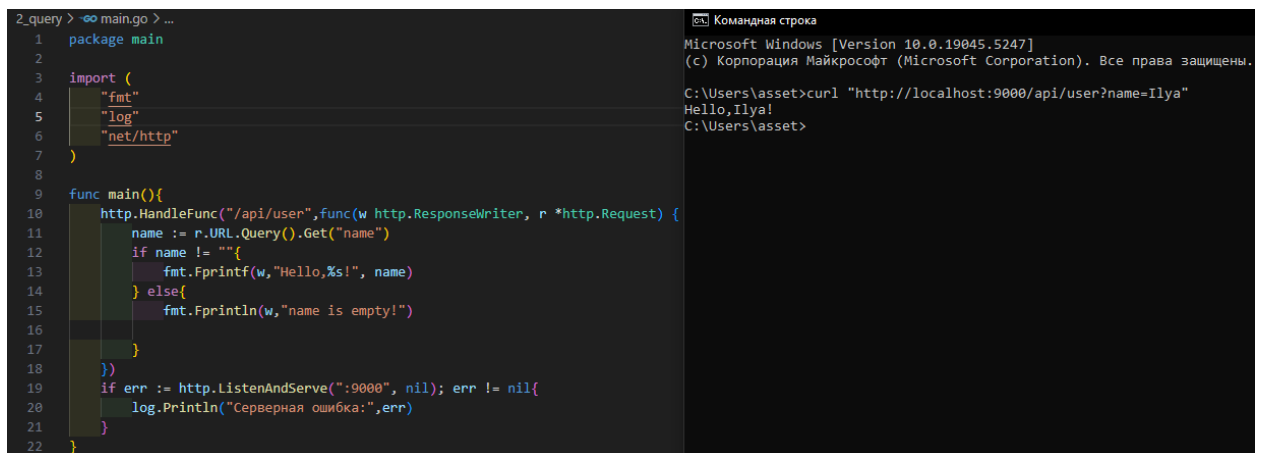
Задание №2 “query”(Рис. 2):

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!



```
2_query > -o main.go > ...
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "net/http"
7 )
8
9 func main(){
10     http.HandleFunc("/api/user",func(w http.ResponseWriter, r *http.Request) {
11         name := r.URL.Query().Get("name")
12         if name != ""{
13             fmt.Fprintf(w,"Hello,%s!", name)
14         } else{
15             fmt.Fprintln(w,"name is empty!")
16         }
17     })
18 }
19 if err := http.ListenAndServe(":9000", nil); err != nil{
20     log.Println("Серверная ошибка:",err)
21 }
22 }
```

Командная строка
Microsoft Windows [Version 10.0.19045.5247]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
C:\Users\asset>curl "http://localhost:9000/api/user?name=Ilya"
Hello,Ilya!
C:\Users\asset>

Рис.2

Задание №3 “count”(Рис. 3):

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

GET: возвращает счетчик

POST: увеличивает ваш счетчик на значение (с ключом "count")

которое вы получаете из формы, но если пришло НЕ число то нужно ответить клиенту: "это не число" со статусом `http.StatusBadRequest` (400).

```
3_count > - main.go > main
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "net/http"
7     "strconv"
8 )
9
10 func main(){
11     counter := 0
12     http.HandleFunc("/count",func(w http.ResponseWriter, r *http.Request) {
13         if r.Method == "GET"{
14             fmt.Fprintf(w,"полученное значение: %d",counter)
15         } else if r.Method == "POST"{
16             r.ParseForm()
17             count, err := strconv.Atoi(r.FormValue("count"))
18             if err != nil{
19                 http.Error(w,"IT'S NOT NUMBER: ",http.StatusBadRequest)
20             }else{
21                 counter += count
22                 fmt.Fprintf(w,"введенное значение: %d",count)
23             }
24         }
25     })
26 }
27 if err := http.ListenAndServe(":3333", nil); err != nil{
28     log.Println("Серверная ошибка:",err)
29 }
30 }
```

Командная строка

Microsoft Windows [Version 10.0.19045.5247]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\asset>curl "http://localhost:9000/api/user?name=Ilya"
Hello,Ilya!
C:\Users\asset>curl http://localhost:3333/count
полученное значение: 0
C:\Users\asset>curl -X POST -d "count=10" http://localhost:3333/count
введенное значение: 10
C:\Users\asset>curl http://localhost:3333/count
полученное значение: 10
C:\Users\asset>curl -X POST -d "count=12" http://localhost:3333/count
введенное значение: 12
C:\Users\asset>curl http://localhost:3333/count
полученное значение: 22
C:\Users\asset>

рис. 3

4. Сделал отчёт и поместил его в директорию docs
Зафиксировал изменения, сделал коммит и отправил полученное состояние ветки dev в удаленный репозиторий GitHub

Вывод

Я изучил основы сетевого взаимодействия и серверной разработки с использованием языка Golang.