



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 № 1 0

Название: Архитектура микросервисов на Golang

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-31Б

(Группа)

(Подпись, дата)

О.И.Ельничных

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2024

Цель работы

Получение первичных навыков организации кодовой базы проекта на Golang

Ход работы

Делаем fork репозитория (Рисунок 1).

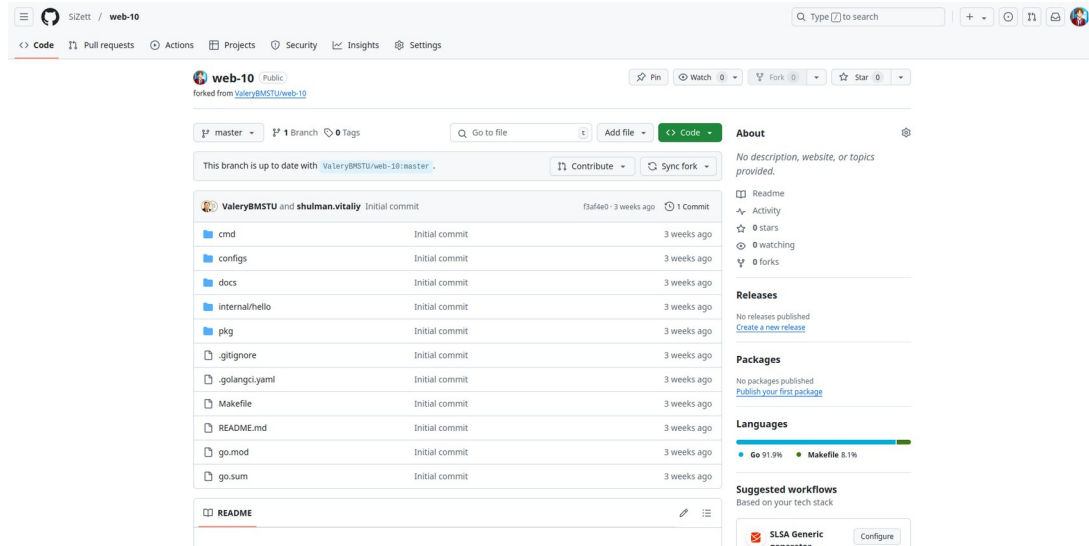


Рисунок 1

Код задания hello:

```
package main
```

```
import (  
    "flag"  
    "log"
```

```
"github.com/ValeryBMSTU/web-10/internal/hello/api"  
"github.com/ValeryBMSTU/web-10/internal/hello/config"  
"github.com/ValeryBMSTU/web-10/internal/hello/provider"  
"github.com/ValeryBMSTU/web-10/internal/hello/usecase"  
_ "github.com/lib/pq"  
)
```

```
func main() {  
    // Считываем аргументы командной строки  
    configPath := flag.String("config-path", "/home/sizet/web-10/configs/hello_example.yaml", "путь к  
    файлу конфигурации")  
    flag.Parse()
```

```
    cfg, err := config.LoadConfig(*configPath)  
    if err != nil {  
        log.Fatal(err)  
    }  
}
```

```

prv := provider.NewProvider(cfg.DB.Host, cfg.DB.Port, cfg.DB.User, cfg.DB.Password,
cfg.DB.DBname)
use := usecase.NewUsecase(cfg.Usecase.DefaultMessage, prv)
srv := api.NewServer(cfg.IP, cfg.Port, cfg.API.MaxMessageSize, use)

srv.Run()
}

```

Код задания query:

```

package main

import (
    "flag"
    "log"

    "github.com/ValeryBMSTU/web-10/internal/query/usecase"

    "github.com/ValeryBMSTU/web-10/internal/query/api"
    "github.com/ValeryBMSTU/web-10/internal/query/config"
    "github.com/ValeryBMSTU/web-10/internal/query/provider"
    _ "github.com/lib/pq"
)

func main() {
    // Считываем аргументы командной строки
    configPath := flag.String("config-path", "/home/sizet/web-10/configs/hello_example.yaml", "путь к
    файлу конфигурации")
    flag.Parse()

    cfg, err := config.LoadConfig(*configPath)
    if err != nil {
        log.Fatal(err)
    }

    prv := provider.NewProvider(cfg.DB.Host, cfg.DB.Port, cfg.DB.User, cfg.DB.Password,
    cfg.DB.DBname)
    use := usecase.NewUsecase(cfg.Usecase.DefaultMessageQuery, prv)
    srv := api.NewServer(cfg.IP, cfg.Port, cfg.API.MaxMessageSize, use)

    srv.Run()
}

```

Код задания count:

```

package main

import (
    "flag"
    "log"

    "github.com/ValeryBMSTU/web-10/internal/count/api"
    "github.com/ValeryBMSTU/web-10/internal/count/config"
    "github.com/ValeryBMSTU/web-10/internal/count/provider"

```

```
"github.com/ValeryBMSTU/web-10/internal/count/usecase"
```

```
_ "github.com/lib/pq"  
)
```

```
func main() {  
    // Считываем аргументы командной строки  
    configPath := flag.String("config-path", "/home/sizet/web-10/configs/hello_example.yaml", "путь к  
    файлу конфигурации")  
    flag.Parse()  
  
    cfg, err := config.LoadConfig(*configPath)  
    if err != nil {  
        log.Fatal(err)  
    }  
  
    prv := provider.NewProvider(cfg.DB.Host, cfg.DB.Port, cfg.DB.User, cfg.DB.Password,  
    cfg.DB.DBname)  
    use := usecase.NewUsecase(cfg.Usecase.DefaultMessageCount, prv)  
    srv := api.NewServer(cfg.IP, cfg.Port, cfg.API.MaxMessageSize, use)  
  
    srv.Run()  
}
```

Заключение

При выполнении заданий лабораторной работы №10 мы получили первичные навыки организации кодовой базы проекта на Golang и выполнили задание основанное на результатах лабораторной работы №6.