



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 рубежному контролю № 1

Название: Разработка WEB-сервера на Golang

Дисциплина: Языки интернет-программирования

Студент

ИУ6-31Б

(Группа)

(Подпись, дата)

Д.Ю. Воронин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

Билет №29. Возведение в куб

Задание:

Необходимо написать веб-сервер на GO, который возводит в куб каждый символ строки. Сервер должен запускаться по адресу `127.0.0.1:8081`.

У сервера должна быть ручка (handler) `GET /kub`. Эта ручка ожидает, что через query-параметр `?string=<передаваемая_строка>` будет передана строка вида `1234`.

При обработке http-запроса каждый символ строки будет возведён в куб.

В качестве ответа сервер должен возвращать JSON с единственным полем `result`.

Пример запроса (curl):

...

```
curl --request GET http://127.0.0.1:8081/kub?string=1234
```

...

Пример ответа:

...

```
{"result":182764}
```

...

Ход работы

Решение

```
package main

import (
    "encoding/json"
    "fmt"
    "net/http"
    "strconv"
)

type Output struct {
    Result string `json:"result"`
}

// Обработчик HTTP-запроса
func CalculateHandler(w http.ResponseWriter, r *http.Request) {
    if r.Method != "GET" {
        w.WriteHeader(http.StatusMethodNotAllowed)
        w.Write([]byte("method not allowed"))
        return
    }

    if !r.URL.Query().Has("string") {
        w.WriteHeader(http.StatusBadRequest)
        w.Write([]byte("Query params don't have property \"string\"!"))
        return
    }

    str := r.URL.Query().Get("string")

    if str == "" {
```

```
    w.WriteHeader(http.StatusBadRequest)
    w.Write([]byte("string is empty!"))
    return
}
```

```
var output Output
for _, symbol := range str {
    number, err := strconv.Atoi(string(symbol))
    if err != nil {
        w.WriteHeader(http.StatusBadRequest)
        w.Write([]byte("wrong format of string!"))
        return
    }
    output.Result += strconv.Itoa(number * number * number)
}
```

```
w.Header().Set("Content-Type", "application/json")
w.WriteHeader(http.StatusOK)
respBytes, _ := json.Marshal(output)
w.Write(respBytes)
}
```

```
func main() {
    // Регистрируем обработчик для пути "/calculate"
    http.HandleFunc("/kub", CalculateHandler)

    // Запускаем веб-сервер на порту 8081
    fmt.Println("starting server...")
    err := http.ListenAndServe("127.0.0.1:8081", nil)
    if err != nil {
        fmt.Println("Ошибка запуска сервера:", err)
    }
}
```

}
}

Тестирование

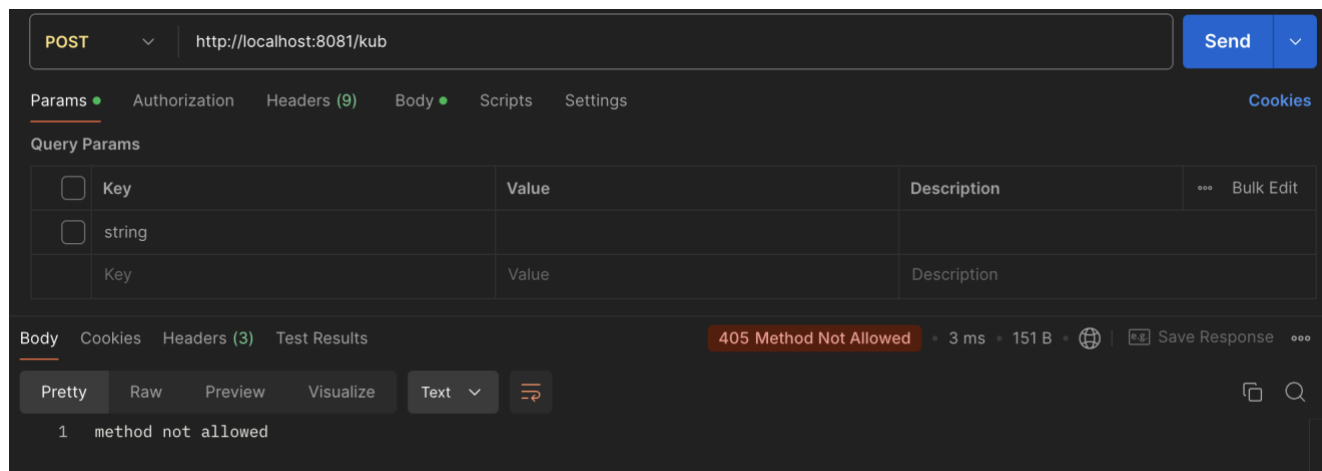


Рисунок 1 – Тестирование 1

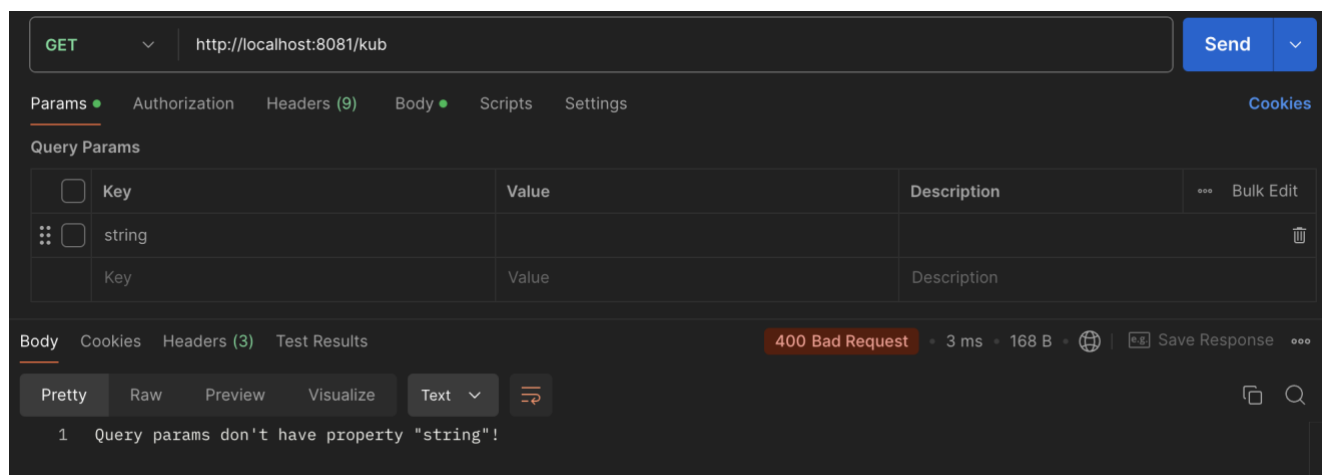


Рисунок 2 – Тестирование 2

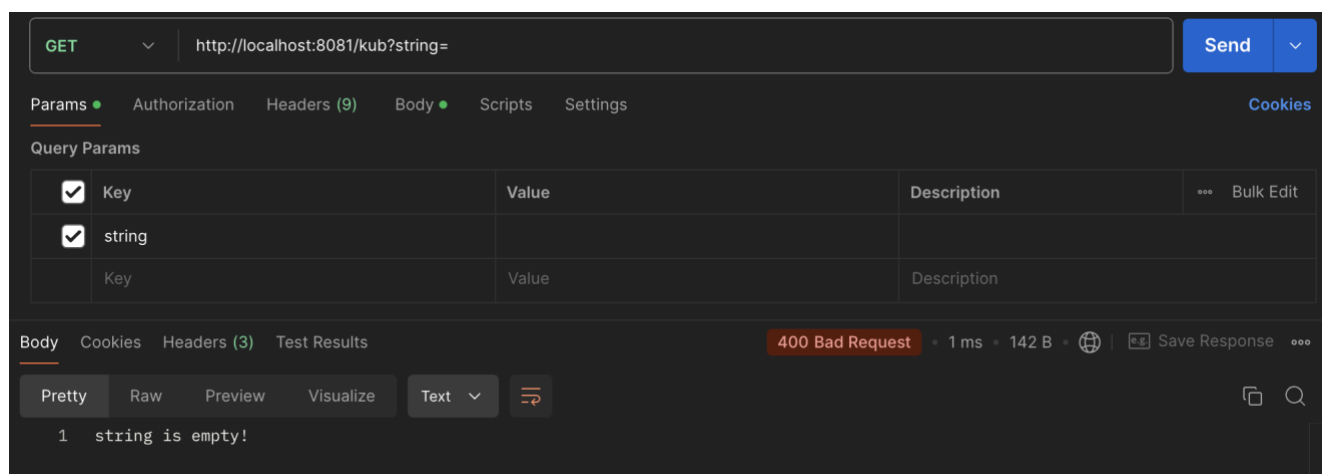


Рисунок 3 – Тестирование 3

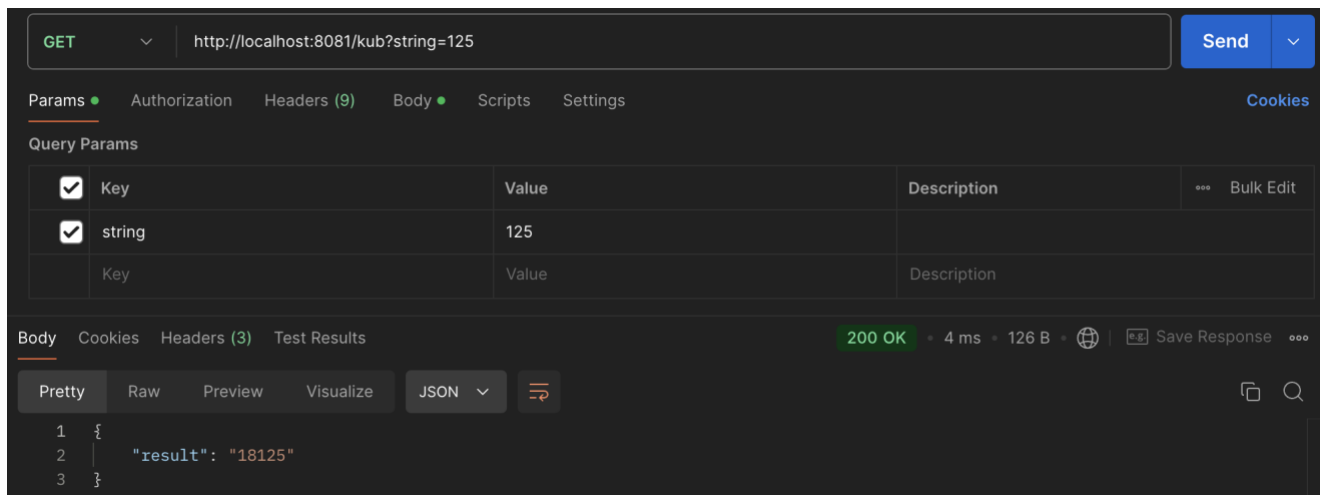


Рисунок 4 – Тестирование 4

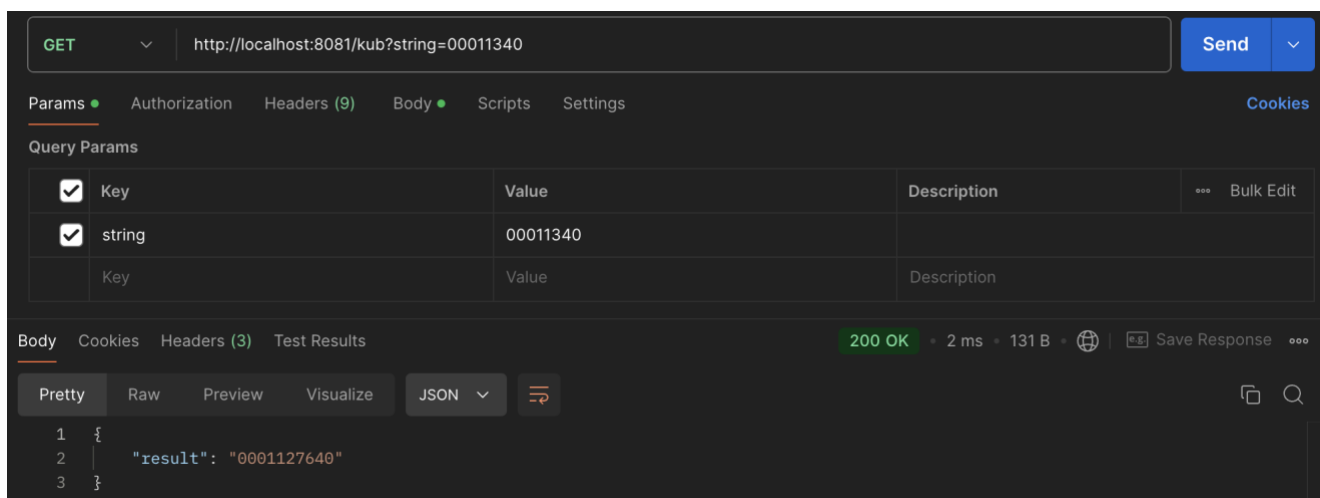


Рисунок 5 – Тестирование 5

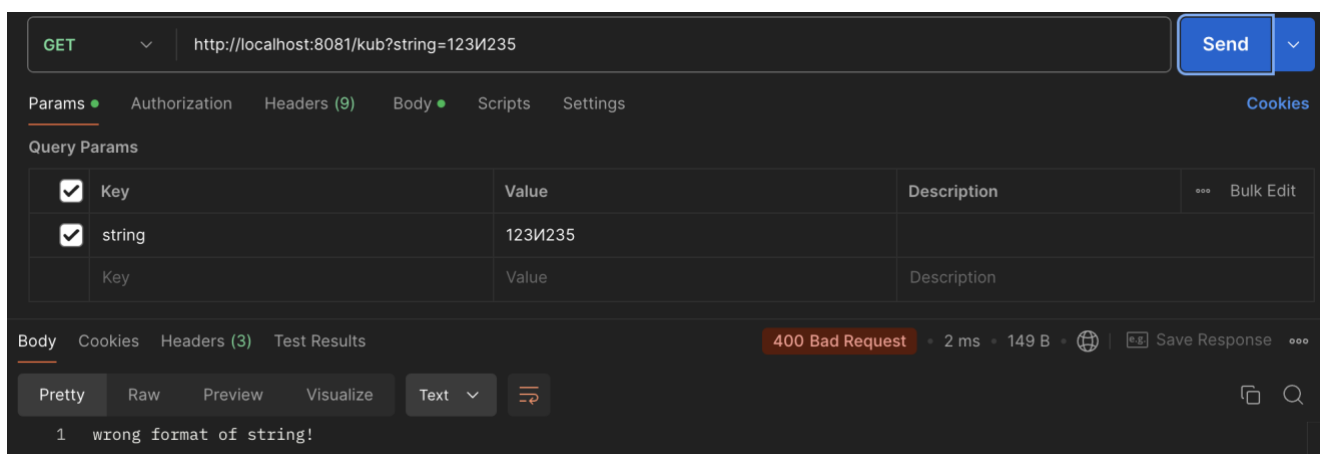


Рисунок 6 – Тестирование 6

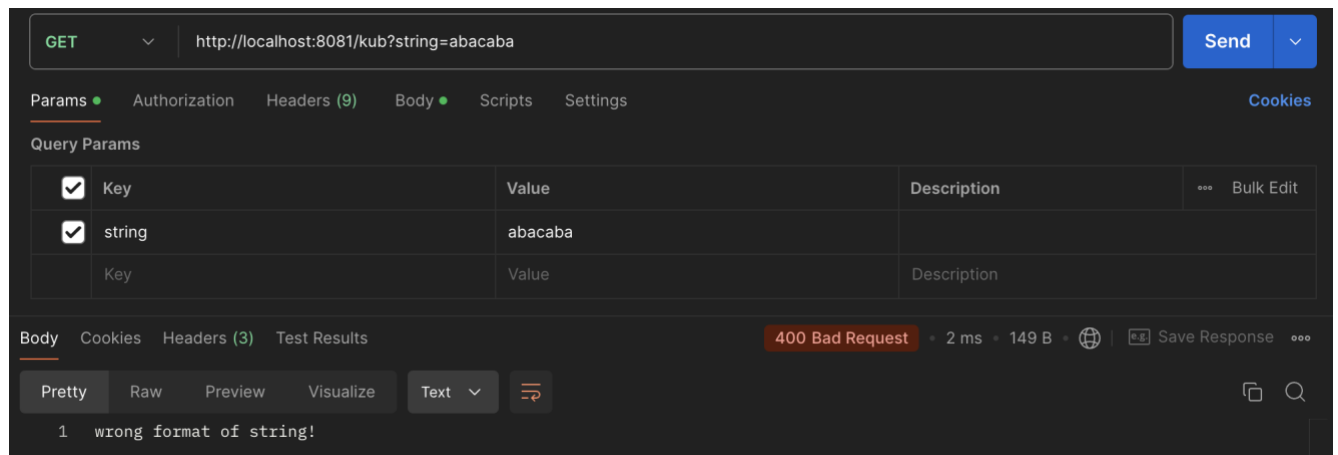


Рисунок 7 – Тестирование 7

Заключение:

Язык программирования Golang позволяет полноценно работать с сетью. Например, есть возможность создать веб-сервер без подключения дополнительных сторонних библиотек с возможностью обработки HTTP запросов.

Список использованных источников:

<https://github.com/ValeryBMSTU/web-rk1>