



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 Название направления

**О Т Ч Е Т**

по рубежному контролю №1

Название: Программирование на GoLang

Дисциплина: Языки интернет программирования

Студент

ИУ6-32Б

(Группа)

22.10.2024

(Подпись, дата)

Т.А. Гаджиев

(И.О. Фамилия)

Преподаватель

22.10.2024

(Подпись, дата)

Т.В. Пелевина

(И.О. Фамилия)

Москва, 2024

## Задание

## Билет №33. Лишние символы

Необходимо написать веб-сервер на GO, удаляющий заданный символ из строки. Сервер должен запускаться по адресу `127.0.0.1:8081`.

У сервера должна быть ручка (handler) `POST /clean`. Эта ручка ожидает, что через JSON будет передано 2 параметра типа string: `array` и `sign`.

При обработке http-запроса из строки удаляется переданный символ. В качестве ответа сервер должен возвращать JSON с единственным полем `result`. Примерм запроса (curl):

```
curl --header "Content-Type: application/json" --request POST --data
'{"array":"1*2п*ке*65*","sign":"*"}' http://127.0.0.1:8081/clean
```

Пример ответа:

```
{"result":"12пке65"}
```

## Цель работы

Разработка веб-сервера на языке Go, который удаляет заданный символ из строки, переданной через HTTP-запрос.

## Ход работы

1) Реализовали сервер с ручкой /clean, продумав обработку исключений

```
package main
```

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

```
type Input struct {
    Array *string `json:"array"`
    Sign *string `json:"sign"`
}
```

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

```
func CleanHandler(w http.ResponseWriter, r *http.Request) {
    if r.Method != "POST" {
        w.WriteHeader(http.StatusMethodNotAllowed)
        w.Write([]byte("method not allowed"))
    }
}
```

```

return
}

var input Input

decoder := json.NewDecoder(r.Body)
err := decoder.Decode(&input)
if err != nil {
w.WriteHeader(http.StatusBadRequest)
w.Write([]byte(err.Error()))
return
}

if input.Array == nil || input.Sign == nil {
w.WriteHeader(http.StatusBadRequest)
w.Write([]byte("param is missing :('))
return
}

sign := *input.Sign

if len([]rune(sign)) > 1 {
w.WriteHeader(http.StatusBadRequest)
w.Write([]byte("некоректно введённый параметр sign: должен быть строкой длиной 1 символ"))
return
}

cleanedString := RemoveCharacter(*input.Array, sign)

output := Output{Result: &cleanedString}

w.Header().Set("Content-Type", "application/json")
json.NewEncoder(w).Encode(output)
}

func RemoveCharacter(input string, charToRemove string) string {
if charToRemove == "" {
return input
}

result := []rune{}
for _, char := range input {
if string(char) != charToRemove {
result = append(result, char)
}
}

return string(result)
}

func main() {
http.HandleFunc("/clean", CleanHandler)
fmt.Println("starting server on 127.0.0.1:8082...")
}

```

```
err := http.ListenAndServe("127.0.0.1:8082", nil)
if err != nil {
panic(err)
}
}
```

2) Протестировали программу(рис. 1). Проверили работу исключений(рис. 2-4)

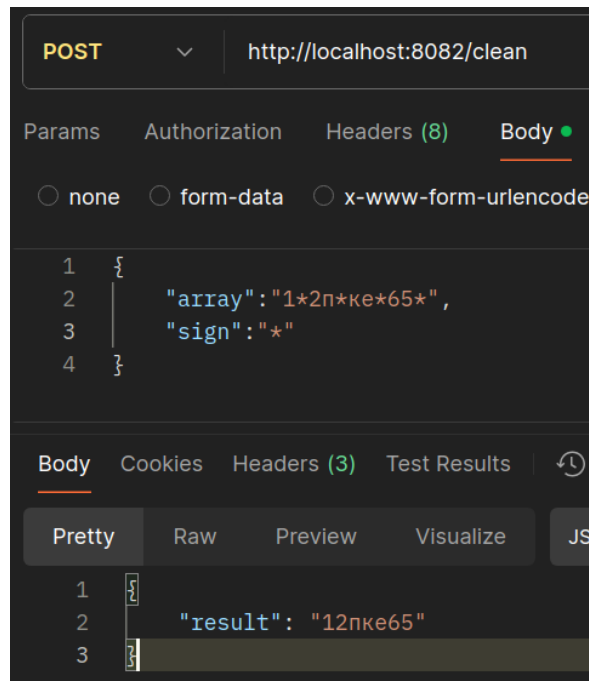


Рисунок 1 — Результат Post запроса

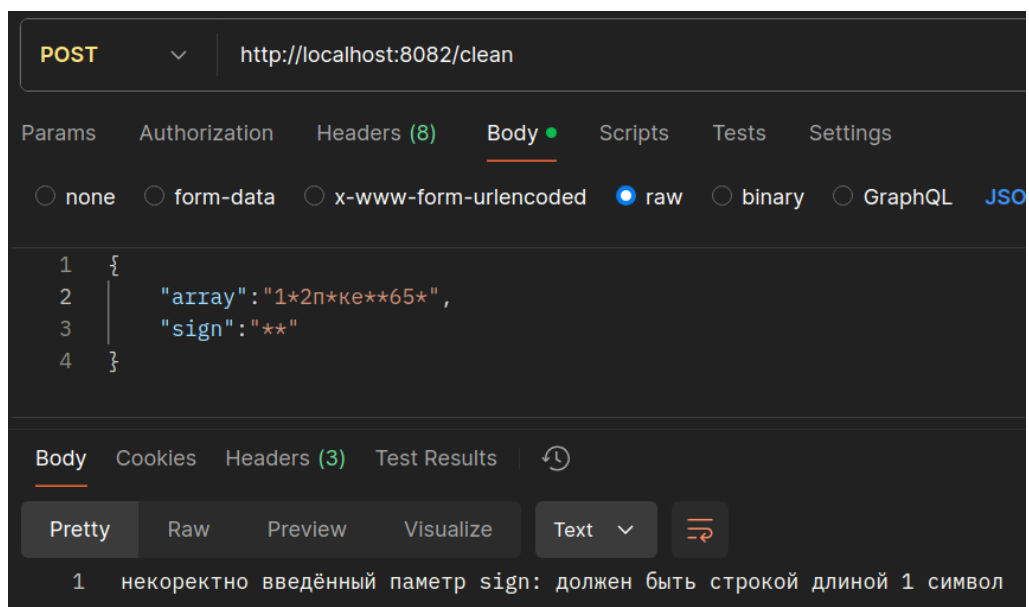


Рисунок 2 — Обработка исключения(длина символа для удаления - 1)

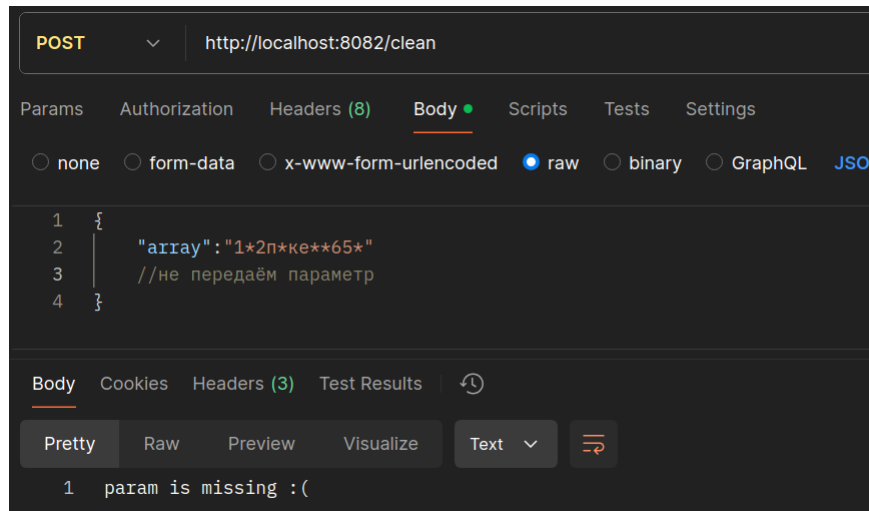


Рисунок 3 — Обработка исключения(отсутствие параметра)

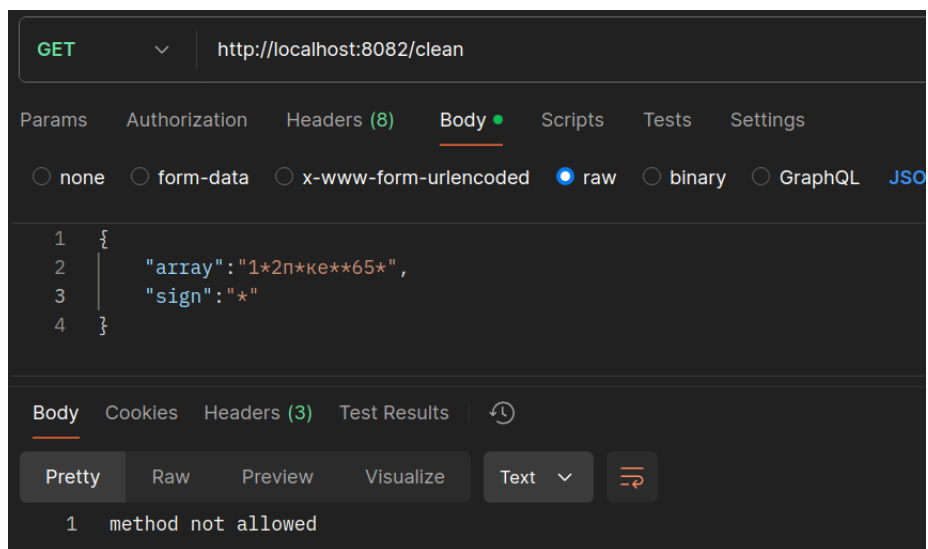


Рисунок 4 — Обработка исключения(проверка на валидность метода запроса)

Вывод: В ходе выполнения рубежного контроля написали веб-сервер на Golang. Закрепили знания о разработке веб-серверов на Go и обработке JSON.