



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 № 9**

**Название:** Back-End разработка с использованием  
фреймворка Echo

**Дисциплина:** Языки-интернет программирования

Студент

ИУ6-33Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Д.Е. Горячев

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Москва, 2024

### 1) Цель

Получение первичных навыков использования веб-фреймворков в Backend разработке на Golang.

### 2) Задание

Доработать сервисы таким образом, чтобы роутинг, обработка запросов, парсинг json, обработка ошибок и логирование осуществлялись на базе фреймворка Echo.

### 3) Ход работы

Фреймворк Echo упрощает разработку серверной части, так как автоматически обрабатывает разные типы HTTP-запросов. Вместо того, чтобы вручную проверять метод запроса (например, POST или GET), как это нужно было делать в стандартной библиотеке Go, с Echo достаточно просто указать, какие запросы сервер должен обрабатывать. Например, для сервиса count не нужно добавлять условные проверки типа:

```
go
Копировать код
if r.Method == "POST" {
    // обработка POST запроса
} else if r.Method == "GET" {
    // обработка GET запроса
} else {
    w.Write([]byte("Разрешены только POST и GET методы"))
}
```

С использованием Echo это можно сделать так:

```
go
Копировать код
e := echo.New()
e.POST("/count", h.PostCounter)
e.GET("/count", h.GetCounter)
```

Таким образом, фреймворк берет на себя часть работы, и разработчик просто определяет, какие обработчики должны срабатывать для конкретных запросов.

### 4) Вывод

В ходе выполнения лабораторной работы был изучен веб-фреймворк Echo, который используется для обработки запросов от клиентов. Echo значительно упрощает разработку, выполняя функции, которые в стандартной библиотеке Go обычно требуют использования нескольких отдельных пакетов, таких как net/http для работы с HTTP-запросами и encoding/json для обработки JSON-данных. Вместо того чтобы использовать эти пакеты вручную, Echo предоставляет удобные и высокоуровневые абстракции для работы с запросами, маршрутизацией и сериализацией данных. Это помогает ускорить процесс разработки и повысить читаемость кода.