



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 № 6**

**Название: Основы Front-End разработки на JavaScript**

**Дисциплина: Языки Интернет Программирования**

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

А.С.Авдеева

(И.О. Фамилия)

Преподаватель

В.Д.Шульман

(Подпись, дата)

(И.О. Фамилия)

Москва, 2024

**Цель работы** — изучение основ Front-End разработки на JavaScript с помощью react и Back-End разработки на GoLang с помощью AJAX-запросов.

**Задание** - Реализовать пользовательский веб-интерфейс для взаимодействия с микросервисами, которые были получены в ходе выполнения предыдущей лабораторной работы. Взаимодействие с Back-End частью веб-приложения должно осуществляться с помощью AJAX-запросов.

Я реализовала React – приложение на JavaScript на порту 8000(рис.1)

← → ↻ ⓘ localhost:8000

## Приложение на React Авдеевой Алёны ИУ6-32Б

### Сервер1 - Hello:

Получить ответ

### Query Service

Enter name Query Name

### Сервер3 - Count подсчёт:

Нажми

Число:0

Введите число Отправить

(рис.1)

**Код к приложению в файле App.js:**

```
import './App.css';
import React, {useState} from 'react';
import { funcHello, funcQuery, funcCount, postCount } from './api';
```

```

function App() {
  const [hello, setHello] = useState('');
  const [query, setQuery] = useState('');
  const [queryResponse, setQueryResponse] = useState('');
  const [count, setCount] = useState(0);
  const [increment, setIncrement] = useState('');
  const handleFuncHello = async () => {
    const data = await funcHello();
    setHello(data);
  };
  const handleFetchQuery = async () => {
    const data = await funcQuery(query);
    setQueryResponse(data);
  };
  const handleFuncCount = async () => {
    const data = await funcCount();
    setCount(data);
  };
  const handleFuncIncrement = async () => {
    if (!increment || isNaN(increment)) {
      alert("Введите корректное число!");
      return;
    }
    try {
      await postCount(increment);
      handleFuncCount();
    } catch (error) {
      console.error("Ошибка при отправке", error);
    }
  };
  return (
    <div>
      <h1>Приложение на React Авдеевой Алёны ИУ6-32Б</h1>
      <div>
        <h2>Сервер1 - Hello:</h2>
        <button onClick={handleFuncHello}>Получить ответ</button>
        <p>{hello}</p>
      </div>
      <div>
        <h2>Query Service</h2>
        <input
          type="text"
          placeholder="Enter name"
          value={query}
          onChange={(e) => setQuery(e.target.value)}
        />
        <button onClick={handleFetchQuery}>Query Name</button>
        <p>{queryResponse}</p>
      </div>
      <div>
        <h2>Сервер3 - Count подсчёт:</h2>
        <button onClick={handleFuncCount}>Нажми</button>
        <p>Число: {count}</p>
        <input
          type="number"

```

```

        placeholder = "Введите число"
        value = {increment}
        onChange={(e) => setIncrement(e.target.value)}
      />
      <button onClick = {handleFuncIncrement}>Отправить</button>
    </div>
  </div>

);
}

export default App;

```

**А также код для работы с API:**

```

import axios from 'axios';

const URL = {
  hello: 'http://localhost:8082/get',
  queryName: 'http://localhost:8083/api/user',
  count: 'http://localhost:8081/count',
};

export const funcHello = async () => {
  const response = await axios.get(URL.hello);
  return response.data;
};

export const funcQuery = async (name) => {
  const response = await axios.get(`${URL.queryName}?name=${name}`);
  return response.data;
};

export const funcCount = async () => {
  const response = await axios.get(URL.count);
  return response.data;
};

export const postCount = async (num) => {
  const response = await axios.post(URL.count, new URLSearchParams({ count: num }));
  return response.data;
};

```

Далее я столкнулась с проблемой, что мой код, изначально написанный для микросервисов в 6 лабораторной работе не подходит, поэтому использовала настройку ответа, которая обрабатывается внутри обработчика запроса.

**Пример для Hello запроса:**

```

package main

import (
  "fmt"
  "net/http"
)

```

```
func handler(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Access-Control-Allow-Origin", "*")
    w.Write([]byte("Hello, web!"))
}

func main() {
    http.HandleFunc("/get", handler)
    err := http.ListenAndServe(":8082", nil) // Указываем порт 8082
    if err != nil {
        fmt.Println("Error starting server:", err)
    }
}
```

Теперь все корректно и успешно выводится на странице(рис.2)

## Приложение на React Авдеевой Алёны ИУ6-32Б

### Сервер1 - Hello:

Hello, web!

### Query Service

Hello, Алена!

### Сервер3 - Count подсчёт:

Число:1

(рис.2) – Добавляем 1

**Заключение** – проделана успешная работа в понимании работы с созданием приложение на React и работой с Вас-End на Golang!