|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)**

**Отчет**

**по лабораторной работе № 7**

**Дисциплина:** Языки интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-32Б |  |  | А. В. Тимохин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В. Д. Шульман |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2024

Цель: изучение основ разработки SPA-приложение на JavaScript.

В рамках данной лабораторной работы предлагается продолжить изучение JavaScript и познакомиться с фреймворком React, используемым для разработки фронтовой части веб-приложения.

Задача:

Реализовать интерфейс для работы с микросервисами из 6 лабораторной работы с использованием react.

Запустим 3 сервера:



Рисунок 1 – запуск сервера

Остальные сервера по аналогии. Также к коду, который был в 6 лабораторной добавим CORS, чтобы было возможно получать данный с хостов через браузер:

w.Header().Set("Access-Control-Allow-Origin", "\*")

w.Header().Set("Access-Control-Allow-Methods", "GET, OPTIONS")

w.Header().Set("Access-Control-Allow-Headers", "Content-Type")

Это пример из файла hello.

Код на react:

// src/App.js

import React, { useState } from "react";

function App() {

  const [webResponse, setWebResponse] = useState("");

  const [userResponse, setUserResponse] = useState("");

  const [counterResponse, setCounterResponse] = useState("");

  const [counterInput, setCounterInput] = useState(0);

  // Функции для получения данных от каждого микросервиса

  const getWebResponse = async () => {

    try {

      const response = await fetch("http://127.0.0.1:8081/get");

      const text = await response.text();

      setWebResponse(text);

    } catch (err) {

      setWebResponse("Ошибка соединения");

    }

  };

  const getUserResponse = async () => {

    try {

      const name = prompt("Введите имя:");

      const response = await fetch(`http://127.0.0.1:9000/api/user?name=${name}`);

      const text = await response.text();

      setUserResponse(text);

    } catch (err) {

      setUserResponse("Ошибка соединения");

    }

  };

  const getCounter = async () => {

    try {

      const response = await fetch("http://127.0.0.1:3333/count");

      const text = await response.text();

      setCounterResponse(text);

    } catch (err) {

      setCounterResponse("Ошибка соединения");

    }

  };

  const incrementCounter = async () => {

    try {

      const response = await fetch("http://127.0.0.1:3333/count", {

        method: "POST",

        headers: { "Content-Type": "application/x-www-form-urlencoded" },

        body: `count=${counterInput}`,

      });

      const text = await response.text();

      setCounterResponse(text);

    } catch (err) {

      setCounterResponse("Ошибка соединения");

    }

  };

  return (

    <div style={{ padding: "20px" }}>

      <h1>Микросервисы</h1>

      <section>

        <h2>Сервис 1: Hello Web</h2>

        <button onClick={getWebResponse}>Получить сообщение</button>

        <p>Ответ: {webResponse}</p>

      </section>

      <section>

        <h2>Сервис 2: Hello, User</h2>

        <button onClick={getUserResponse}>Введите имя и получите ответ</button>

        <p>Ответ: {userResponse}</p>

      </section>

      <section>

        <h2>Сервис 3: Счетчик</h2>

        <button onClick={getCounter}>Получить текущий счетчик</button>

        <br />

        <input

          type="number"

          value={counterInput}

          onChange={(e) => setCounterInput(Number(e.target.value))}

          placeholder="Введите число"

        />

        <button onClick={incrementCounter}>Увеличить счетчик</button>

        <p>Ответ: {counterResponse}</p>

      </section>

    </div>

  );

}

export default App;

Результат работы:

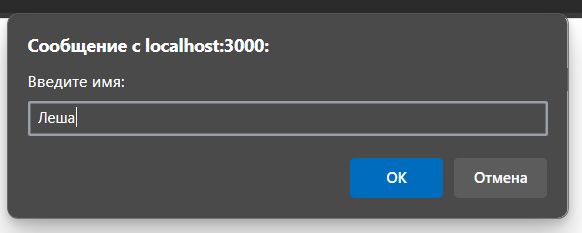


Рисунок 2 – ввод имени для 2 микросервиса

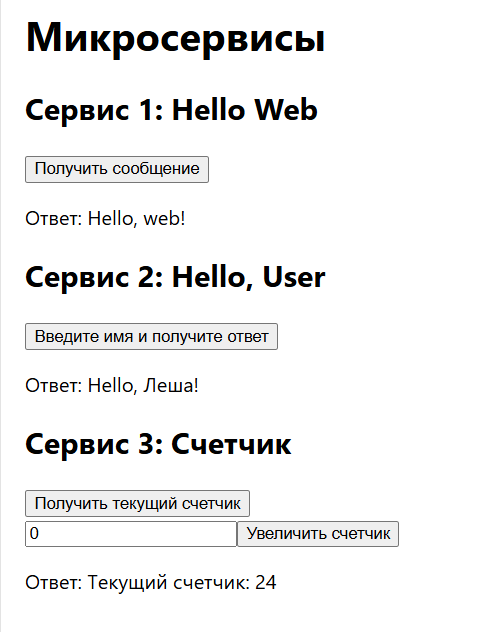


Рисунок 3 – результат работы программы

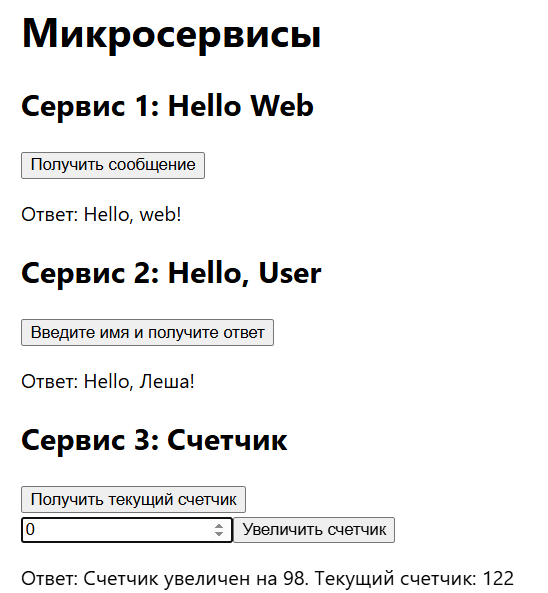


Рисунок 4 – результат работы при добавлении count

Вывод: я ознакомился с react’ом и написал на нем приложение на локальном хосте.