|  |  |
| --- | --- |
| **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**

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

**Дисциплина:** Языки интернет программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-32Б |  |  | А.Н. Абрамов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | И.О. Фамилия |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2024

# **Основы Front-End разработки на JavaScript**

**Цель работы:** 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.

**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.

**Решение:**

// App.js

import Hello from "./Hello.jsx"

import Query from "./Query.jsx"

import Count from "./Count.jsx"

export default function App() {

return (

<>

<Hello />

<Query />

<Count />

</>

);

}

// Count.js

import { useState } from "react"

export default function Count() {

const [reply, setReply] = useState(null)

const [value, setValue] = useState(0)

function onClickLoad() {

setReply("")

fetch(`http://127.0.0.1:8080/count`).then(

response => response.json().then(json => setReply(json.count)),

error => setReply(null)

)

}

function onClickChange() {

setReply("")

fetch(`http://127.0.0.1:8080/count`, {

method: 'POST',

headers: {'Content-Type': 'application/json'},

body: JSON.stringify({ "count": value }),

}).then(

response => response.status === 200 ? setReply("OK") : response.text().then(text => setReply(text))

)

setValue(0)

}

let content = <h1>No data, please load</h1>

let style = { borderColor: "red" }

if (reply !== null) {

content = <h1>{reply === '' ? 'Loading...' : reply}</h1>

style.borderColor = reply === '' ? 'black' : "green"

}

return (

<section style={style}>

{content}

<button onClick={onClickLoad}>Загрузить</button>

<input type="number" value={value} onChange={e => setValue(parseInt(e.target.value))}></input>

<button onClick={onClickChange}>Изменить</button>

</section>

)

}

// main.go

package main

import (

"fmt"

"net/http"

)

func main() {

http.HandleFunc("/get", HelloHandler) // hello.go

http.HandleFunc("/api/user", QueryHandler) // query.go

http.HandleFunc("/count", CountHandler) // count.go

fmt.Println("Server is listening on 127.0.0.1:8080")

error := http.ListenAndServe("127.0.0.1:8080", nil)

if error != nil {

fmt.Println(error)

}

}

**Заключение:** Были изучены Основы Front-End разработки на JavaScript

**Использованные источники**

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Promise/then

https://www.w3schools.com/react/default.asp