



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 № 7

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

**Дисциплина: Языки Интернет-программирования**

Студент

ИУ6-31Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

О.И.Ельничных

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

Москва, 2024

## 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 фронтонвой части веб-приложения.

## Ход работы

Делаем fork репозитория (Рисунок 1).

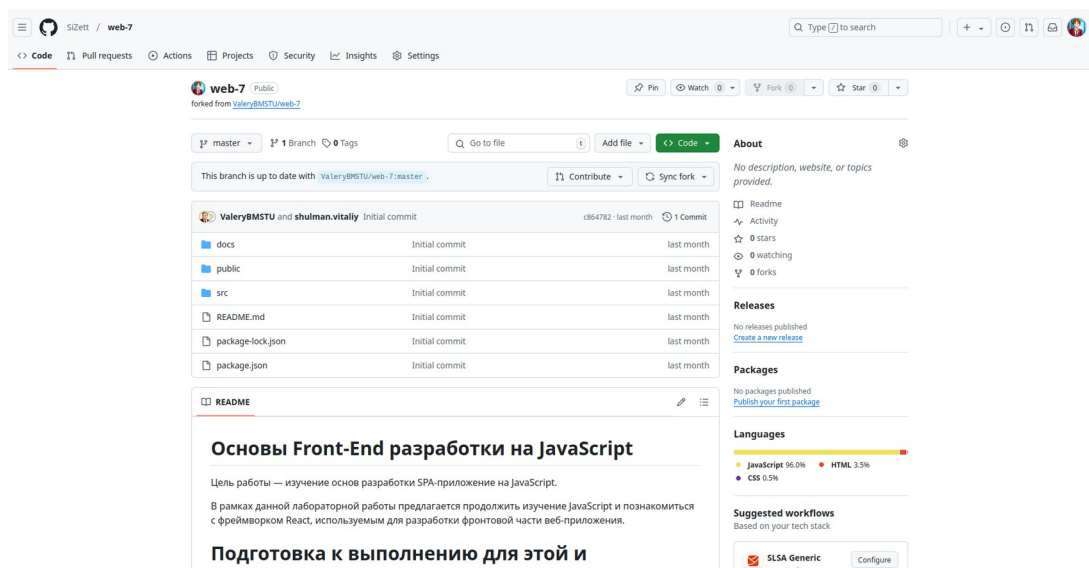


Рисунок 1

Код задания 1\_hello:

```
package main
// некоторые импорты нужны для проверки
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("/", handler)

    // Запускаем веб-сервер на порту 8080
    err := http.ListenAndServe(":8082", nil)
    if err != nil {
```

```

    fmt.Println("Ошибка запуска сервера:", err)
}
}

```

### Код задания 2\_query:

```

package main

// некоторые импорты нужны для проверки
import (
    "fmt"
    "net/http" // пакет для поддержки HTTP протокола
)

func handler(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Access-Control-Allow-Origin", "*")
    s := r.URL.Query().Get("name")
    w.Write([]byte("Hello," + s + "!"))
}

func main() {
    http.HandleFunc("/api/user", handler)

    err := http.ListenAndServe(":8083", nil)
    if err != nil {
        fmt.Println("Ошибка запуска сервера:", err)
    }
}

```

### Код задания 3\_count:

```

package main

// некоторые импорты нужны для проверки
import (
    "fmt"
    "net/http"
    "strconv" // вдруг понадобится вам ;)
)

var count1 int = 0

func handler(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Access-Control-Allow-Origin", "*")
    if r.Method == "GET" {
        w.WriteHeader(http.StatusOK)
        w.Write([]byte(strconv.Itoa(count1)))
        return
    } else if r.Method == "POST" {
        r.ParseForm()
        s := r.FormValue("count")
        if s == "" {
            w.WriteHeader(http.StatusBadRequest)
            w.Write([]byte("это не число"))
        }
    }
}

```

```

        return
    }
    number, err := strconv.Atoi(s)
    if err != nil {
        w.WriteHeader(http.StatusBadRequest)
        w.Write([]byte("это не число"))

        return
    }
    count1 += number
    return
} else {
    w.WriteHeader(http.StatusMethodNotAllowed)
    w.Write([]byte("Метод не поддерживается"))
    return
}
}

func main() {
    http.HandleFunc("/", handler)

    err := http.ListenAndServe(":8081", nil)
    if err != nil {
        fmt.Println("Ошибка запуска сервера!")
    }
}

```

## **Код App.js для пользовательского интерфейса:**

```

import React, { useState } from 'react';

const RequestForm = () => {
    const [requestType, setRequestType] = useState('hello');
    const [countMethod, setCountMethod] = useState('GET');
    const [inputValue, setInputValue] = useState('');
    const [response, setResponse] = useState('');

    const handleSubmit = async (e) => {
        e.preventDefault();
        let url;
        let method = 'GET';
        let body = null;

        if (requestType === 'hello') {
            url = 'http://localhost:8082';
        } else if (requestType === 'query') {
            url = `http://localhost:8083/api/user?name=${inputValue}`;
        } else if (requestType === 'count') {
            url = 'http://localhost:8081';
            method = countMethod;
            if (countMethod === 'POST' && inputValue) {

```

```

    body = new URLSearchParams({ count: inputValue });
  }
}

try {
  const res = await fetch(url, {
    method,
    headers: {
      'Content-Type': 'application/x-www-form-urlencoded',
    },
    body,
  });
  const data = await res.text();
  setResponse(data);
} catch (error) {
  setResponse('Ошибка при выполнении запроса');
}
};

return (
  <div>
    <h1>Отправка запросов</h1>
    <form onSubmit={handleSubmit}>
      <div>
        <label>
          <input
            type="radio"
            value="hello"
            checked={requestType === 'hello'}
            onChange={(e) => setRequestType(e.target.value)}
          />
          Hello
        </label>
        <label>
          <input
            type="radio"
            value="query"
            checked={requestType === 'query'}
            onChange={(e) => setRequestType(e.target.value)}
          />
          Query
        </label>
        <label>
          <input
            type="radio"
            value="count"
            checked={requestType === 'count'}
            onChange={(e) => setRequestType(e.target.value)}
          />
          Count
        </label>
      </div>
    </div>
  )
)

```

```

{requestType === 'count' && (
  <div>
    <label>
      <input
        type="radio"
        value="GET"
        checked={countMethod === 'GET'}
        onChange={(e) => setCountMethod(e.target.value)}
      />
      GET
    </label>
    <label>
      <input
        type="radio"
        value="POST"
        checked={countMethod === 'POST'}
        onChange={(e) => setCountMethod(e.target.value)}
      />
      POST
    </label>
  </div>
)}
{(requestType === 'query' || (requestType === 'count' && countMethod === 'POST')) && (
  <div>
    <input
      type="text"
      value={inputValue}
      onChange={(e) => setInputValue(e.target.value)}
      placeholder={requestType === 'query' ? 'Введите имя' : 'Введите число'}
    />
  </div>
)}
<button type="submit">Отправить</button>
</form>
{response && <div>Ответ: {response}</div>}
</div>
);
};

export default RequestForm;

```

## Заключение

При выполнении заданий лабораторной работы №7 мы изучили основы разработки SPA-приложения на JavaScript и выполнили задание основанное на результатах лабораторной работы №6.