



**«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)

**О т ч е т**

**по лабораторной работе №7**

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

**Дисциплина:** Языки интернет-программирования

Студент гр. ИУ6-33Б

\_\_\_\_\_  
(Подпись, дата)

Цыганчук П. В.  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

В.Д. Шульман  
(И.О. Фамилия)

Москва, 2024

## Введение

### Цель

Изучение основ разработки SPA-приложение на JavaScript.

### 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.

### Ход работы

Отдельно созданы файлы в директории *pages* для описания контента каждой из трех страниц. Рассмотрим каждую из них.

#### 1) Hello

Здесь необходимо отправить лишь GET-запрос и отобразить ответ на странице (рис. 1)

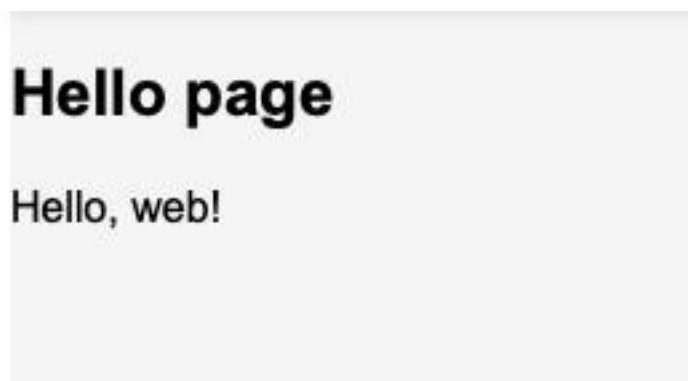


Рисунок 1. Страница приветствия

Далее приведен листинг отправки запроса и получения ответа.

```
componentDidMount() {  
  fetch('http://127.0.0.1:8080/get')  
    .then((response) => {  
if (!response.ok) {  
      throw new Error(`Ошибка HTTP: ${response.status}`);  
    }  
    return response.text(); // Получаем текст HTML, а не промис  
  })  
    .then((data) => {  
this.setState({  
  response: data,  
  isLoading: false  
})  
})  
}
```

```

    })
    .catch((error) => {
        console.log('Произошла ошибка')
    console.log(error.message)
    });
}

```

## 2) UserGreeting

На данной странице необходимо реализовать ввод имени, которое будет отправляться в URL-запросе (рис. 2)

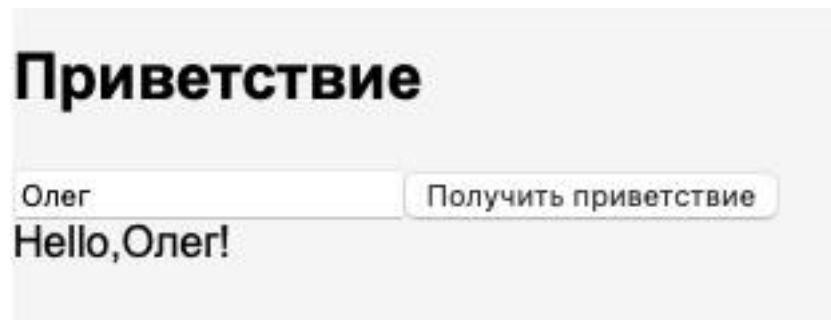


Рисунок 2. Отправка имени

Ниже представлен листинг отправки запроса и получения ответа

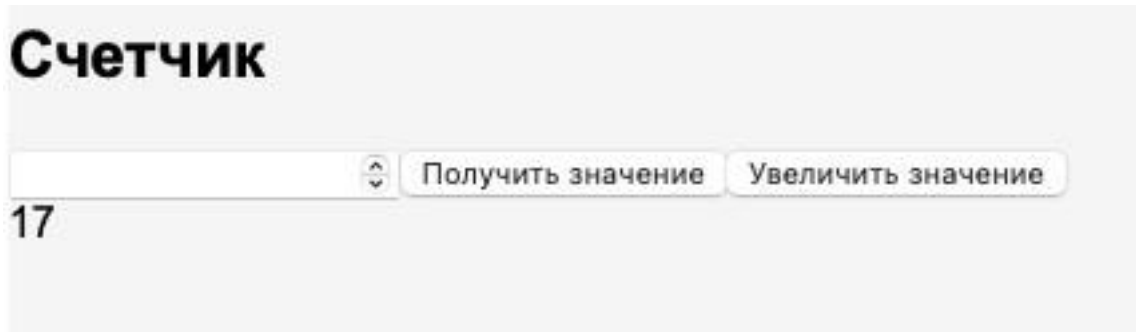
```

handleSubmit = () => {
    const { name } = this.state;
    if (!name) return; // Если имя не указано, не отправляем запрос
    this.setState({ isLoading: true, error: null });
    fetch(`http://localhost:9000/api/user?name=${name}`)
        .then((response) => {
            if (!response.ok) {
                throw new Error(`Ошибка HTTP: ${response.status}`);
            }
            return response.text(); // Ожидаем текстовый ответ
        })
        .then((data) => {
            this.setState({ greeting: data, isLoading: false });
        })
        .catch((error) => {
            this.setState({ error: error.message, isLoading: false });
        });
};

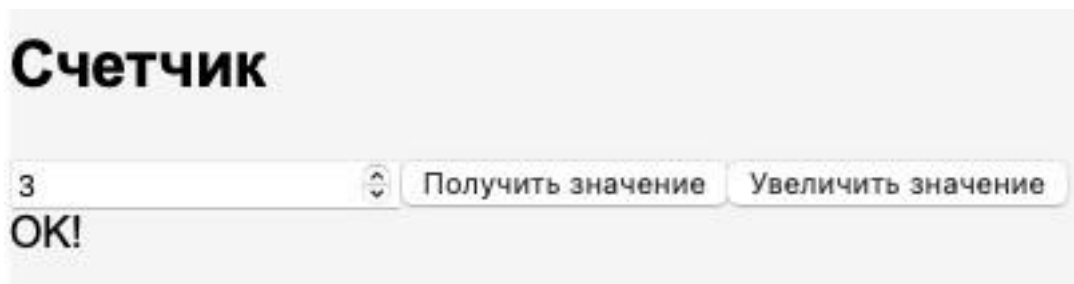
```

## 3) Counter

На этой странице пользователь либо получает значение счетчика, либо увеличивает его на определенное значение (рис. 3, 4)



*Рисунок 3. Получение значения*



*Рисунок 4. Увеличение значения*

### **Вывод**

Таким образом, изучены основы фреймворка React, используемого для фронтальной части веб-приложения. Добавили маршрутизацию, которая позволяет переключаться между страницами приложения без перезагрузки страницы целиком.