|  |  |
| --- | --- |
| ГУАП  КАФЕДРА № 43  ОТЧЕТ  ЗАЩИЩЕН С ОЦЕНКОЙ  ПРЕПОДАВАТЕЛЬ |  |
|  |  |
| ассистент | Кочин Д.А. |

подпись, дата

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Разработка простого AJAX приложения Spring

по дисциплине: ТЕХНОЛОГИИ РАЗРАБОТКИ СЕРВЕРНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4936 Петровнина Д.В.

подпись, дата фамилия, инициалы

Санкт-Петербург 2021

**Лабораторная работа №3**

Разработка простого AJAX приложения Spring

**Цель работы:**   
Разработать AJAX приложение на основе Angular, Vue или React, взаимодействующее с раннее написанным REST сервисом посредством соответствующих HTTP запросов. Реализовать задачи из лабораторной 1 (чтение, добавление и удаление данных). Обмен данными (объектами) должен происходить в текстовом формате JSON.  
  
**Задание на лабораторную работу:**

1. Создайте один или несколько форм и контроллеров для страниц приложения. Подключите библиотеку angularJS
2. Реализуйте задачи из лабораторной работы 1 средствами AJAX и REST-сервисов, разработанных в предыдущей лабораторной работе.

16. Система безопасности предприятия (помещения, люди, права на вход)  
**Ход работы:**

1. Создадим клиентский javascript проект на основе фреймворка Vue.js. REST сервис, написанный раннее, работает на Spring, Java
2. Определим основные формы(компоненты) клиентского приложения для выполнения необходимых функций (чтение, добавление и удаление данных):  
   - Форма добавления нового элемента

- Форма вывода одного элемента списка  
- Форма вывода списка данных

1. Форма добавления просит пользователя ввести необходимые данные и передает их в главную родительскую форму, которая с помощью POST запроса передает их на сервер.   
     
    this.$emit('create', this.estate)

this.estate = {

room: '',

people: '',

right: “

}

async postEstate(estate) {

try {

const response = await axios.post('http://localhost:8080/estates”, estate);

} catch (e) {

alert('Ошибка')

}

}

Форма вывода единичного элемента содержит в себе текстовые поля для вывода основных данных объекта и кнопку удаления.

<div class="estate" >

<div>

<div><strong>Квартира: </strong>{{ estate.room }}</div>

<div><strong>Люди: </strong>{{ estate.people }}</div>

<div><strong>Права: </strong>{{ estate.right }}</div>

</div>

<div class="estate\_btns">

<my-button @click="$emit('remove', estate)"

type="submit" class="btn btn-sm red">

Удалить

</my-button>

</div>

</div>

@remove="$emit('remove', estate)"

/>

Полный код программы находится в Приложении 1.

1. Каждая форма содержит в себе поля и необходимые скрипты для их обработки
2. Данные с сервера постоянно обновляются (без перезагрузки страницы)

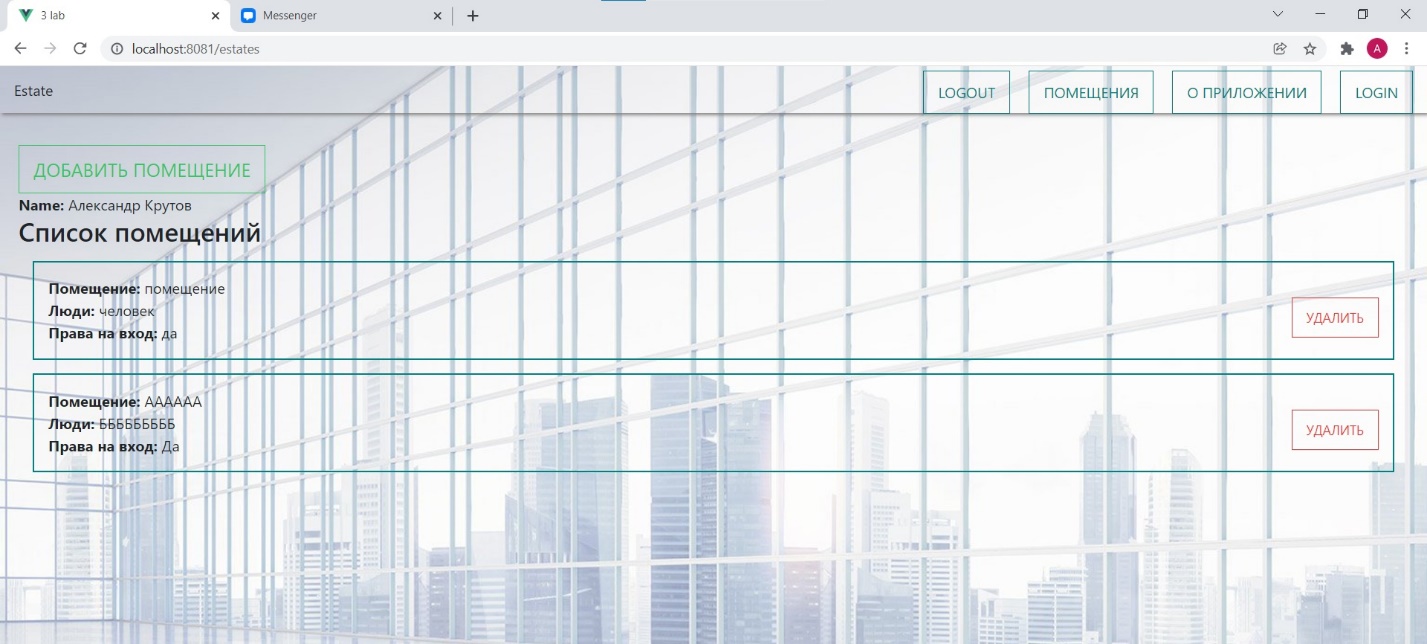
mounted() {  
 this.fetchEstates()  
}

1. Данные на сервер(POST) и с него (GET) передаются в текстовом формате JSON, который далее преобразуется в соответствующее объекты классов приложения.

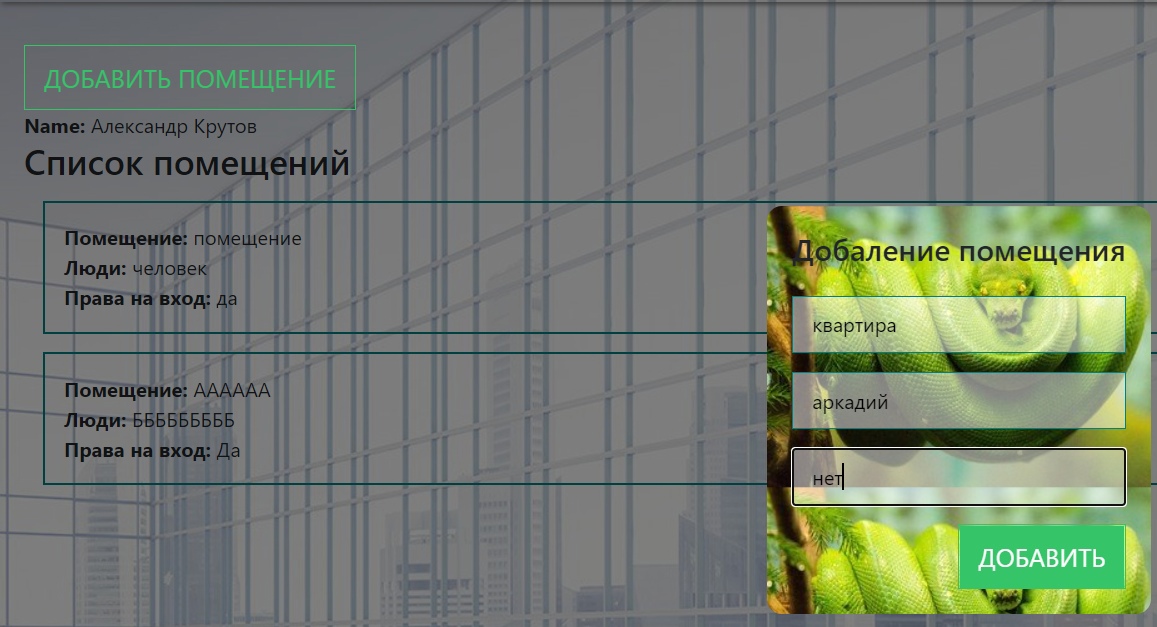
Клиент отправляет все запросы с помощью axios.

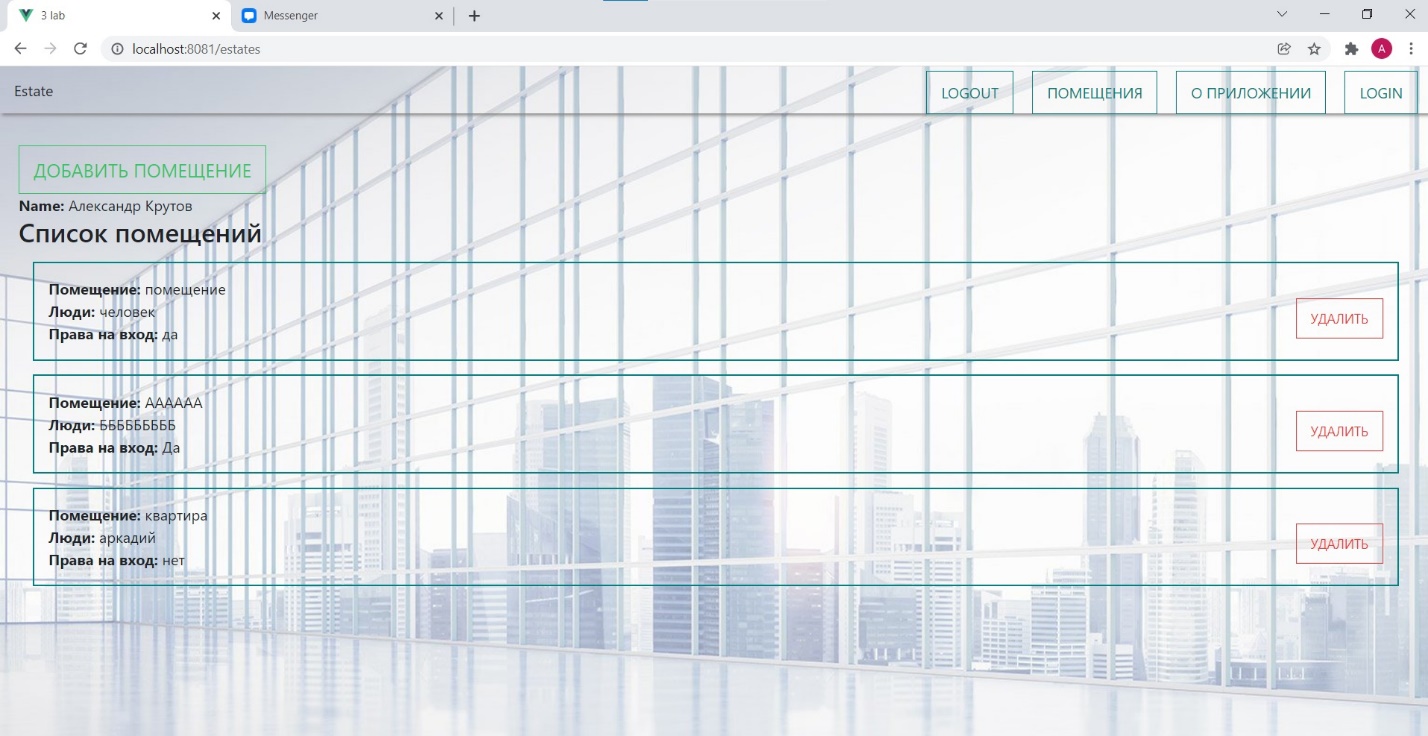
1. Проверим работы программы:

Получение данные(GET):

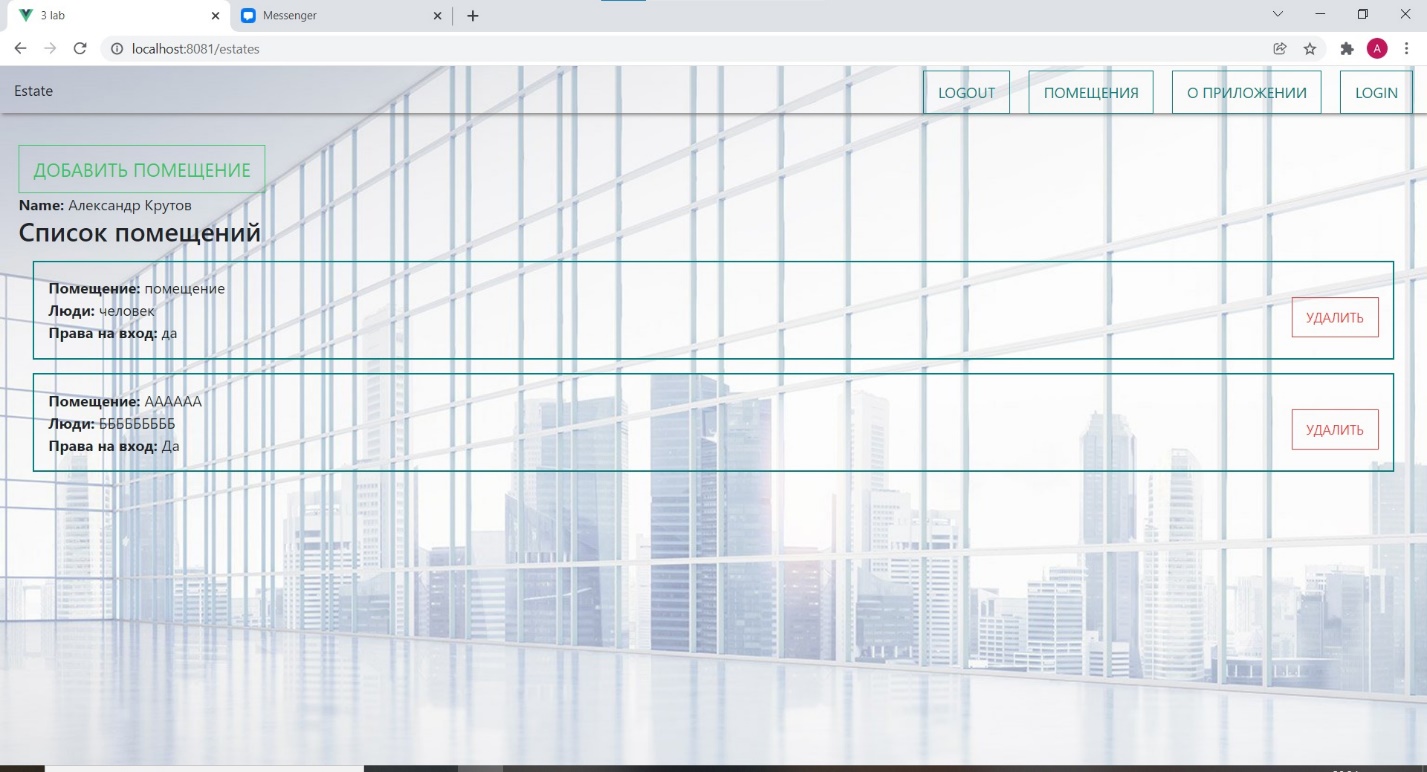


Создание нового элемента(POST):





Удаление элемента(DELETE):



**Вывод:** в ходе лабораторной работы были получены основные навыки разработки front-end части AJAX веб приложения с помощью javascript. Изучены основы взаимодействия клиента и REST сервиса с использованием JSON.

**Приложение 1.**