МИНИСТЕРСТВО ТРАНСПОРТА РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (МИИТ)»

Кафедра ЦТУТП

**Отчёт**

По лабораторной работе №8  
по дисциплине «Корпоративные информационные системы»

Тема: «Написание приложение Vue»

Вариант №19

Выполнил: Любченкова О. А.

Группа: УИС-411

Преподаватель: доц. Кафедры ЦТУТП

Козьяков П. О.

­

Москва 2024 г.

**СОДЕРЖАНИЕ**

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc180325218)

[ХОД ВЫПОЛНЕНИЯ РАБОТЫ 4](#_Toc180325219)

[РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ 5](#_Toc180325220)

[КОД ПРОГРАММЫ 6](#_Toc180325221)

[ВЫВОД 9](#_Toc180325222)

# **ПОСТАНОВКА ЗАДАЧИ**

Для контроллера, реализованного в лабораторной работе №7, создать приложение на Vue, выполняющее запросы к контроллеру и обрабатывающее ответы.

# **ХОД ВЫПОЛНЕНИЯ РАБОТЫ**

В рамках лабораторной работы была разработана клиентская часть приложения для управления железнодорожными путями, реализованная на Vue.js. Основной функционал включает добавление, обновление, удаление и просмотр информации о путях. Приложение взаимодействует с серверной частью через REST API, предоставляемый на сервере, работающем по адресу http://localhost:8081.

Первым шагом было создание компонентов для работы с железнодорожными путями: RailwayForm и RailwayList. Компонент RailwayForm отвечает за добавление новых путей или обновление существующих. Он содержит форму с полями для ввода названия пути, длины, страны и статуса оперативности. Логика отправки данных реализована с использованием методов, предоставляемых сервисом railwayService.

Компонент RailwayList предназначен для отображения списка путей. Он запрашивает данные с сервера, выводит их в виде списка и предоставляет возможность удаления записи через соответствующую кнопку. После удаления список автоматически обновляется благодаря вызову метода fetchRailways.

Для взаимодействия с сервером был создан сервис railwayService, который реализует методы для отправки запросов на получение, добавление, обновление и удаление данных. Все запросы используют библиотеку Axios, а базовый URL настроен в файле конфигурации.

Основной компонент приложения App объединяет оба вышеописанных компонента, предоставляя удобный интерфейс для управления данными. В корневом компоненте прописана базовая структура, связывающая RailwayForm и RailwayList в одном интерфейсе.

В ходе работы было настроено взаимодействие с сервером, обеспечена правильная обработка данных, а также добавлены проверки корректности заполнения формы в компоненте RailwayForm.

# **РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

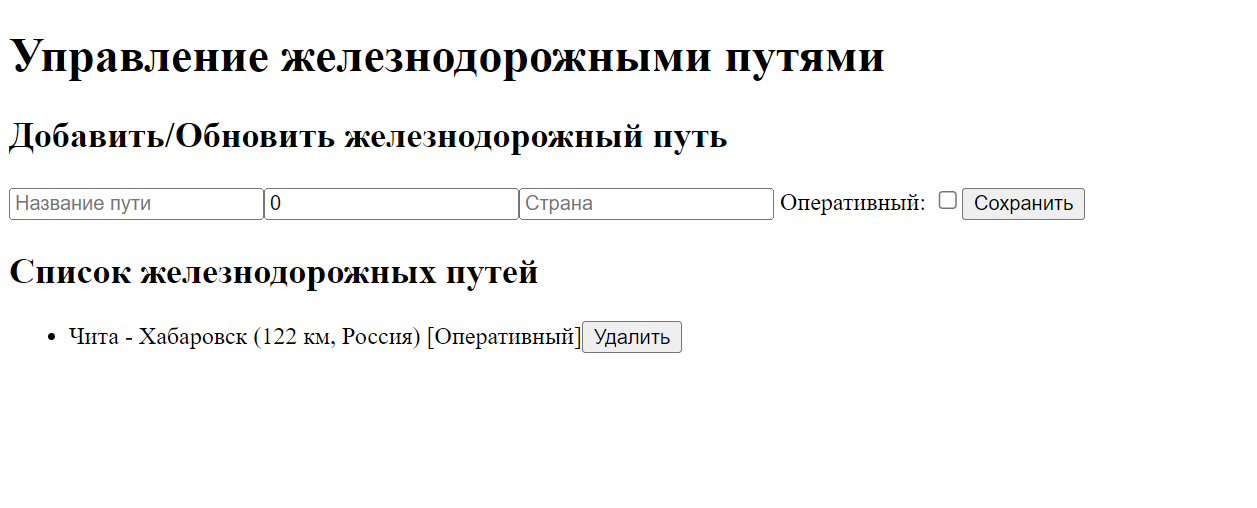


Рисунок 1 – Результат работы программы

# **КОД ПРОГРАММЫ**

<template>  
 <div>  
 <h2>Добавить/Обновить железнодорожный путь</h2>  
 <form @submit.prevent="submitRailway">  
 <input v-model="railway.name" placeholder="Название пути" required />  
 <input v-model.number="railway.length\_km" placeholder="Длина пути (км)" type="number" step="0.1" required />  
 <input v-model="railway.country" placeholder="Страна" required />  
 <label>  
 Оперативный:  
 <input type="checkbox" v-model="railway.is\_operational" />  
 </label>  
 <button type="submit">Сохранить</button>  
 </form>  
 </div>  
</template>  
  
<script>  
import railwayService from '@/services/railwayService';  
  
export default {  
 data() {  
 return {  
 railway: {  
 id: null,  
 name: '',  
 length\_km: 0,  
 country: '',  
 is\_operational: false,  
 },  
 };  
 },  
 methods: {  
 async submitRailway() {  
 if (this.railway.id) {  
 await railwayService.updateRailway(this.railway);  
 } else {  
 await railwayService.addRailway(this.railway);  
 }  
 this.clearForm();  
 },  
 clearForm() {  
 this.railway = {  
 id: null,  
 name: '',  
 length\_km: 0,  
 country: '',  
 is\_operational: false,  
 };  
 },  
 },  
};  
</script>

<template>  
 <div>  
 <h2>Список железнодорожных путей</h2>  
 <ul>  
 <li v-for="railway in railways" :key="railway.id">  
 {{ railway.name }} ({{ railway.length\_km }} км, {{ railway.country }})  
 <span v-if="railway.is\_operational">[Оперативный]</span>  
 <button @click="deleteRailway(railway.id)">Удалить</button>  
 </li>  
 </ul>  
 </div>  
</template>  
  
<script>  
import railwayService from '@/services/railwayService';  
  
export default {  
 data() {  
 return {  
 railways: [],  
 };  
 },  
 methods: {  
 async fetchRailways() {  
 const response = await railwayService.getRailways();  
 this.railways = response.data;  
 },  
 async deleteRailway(id) {  
 await railwayService.deleteRailway(id);  
 this.fetchRailways();  
 },  
 },  
 created() {  
 this.fetchRailways();  
 },  
};  
</script>

import axios from 'axios';  
  
const API\_URL = 'http://localhost:8081/railways';  
  
export default {  
 getRailways() {  
 return axios.get(API\_URL);  
 },  
 addRailway(railway) {  
 console.log('Перед отправкой:', railway); // Для проверки  
 return axios.post(API\_URL, railway);  
 },  
 updateRailway(railway) {  
 return axios.put(API\_URL, railway);  
 },  
 deleteRailway(id) {  
 return axios.delete(`${API\_URL}/${id}`);  
 },  
};

<template>  
 <div id="app">  
 <h1>Управление железнодорожными путями</h1>  
 <RailwayForm />  
 <RailwayList />  
 </div>  
</template>  
  
<script>  
import RailwayList from '@/components/RailwayList.vue';  
import RailwayForm from '@/components/RailwayForm.vue';  
  
export default {  
 components: {  
 RailwayList,  
 RailwayForm,  
 },  
};  
</script>

import { createApp } from 'vue';  
import App from './App.vue';  
import axios from 'axios';  
  
axios.defaults.baseURL = 'http://localhost:8081'; // Адрес бэкенда  
createApp(App).mount('#app');

# **ВЫВОД**

В результате выполнения лабораторной работы было создано удобное клиентское приложение для управления железнодорожными путями. Пользователь может добавлять новые пути, обновлять существующие, удалять ненужные записи, а также просматривать список доступных данных. Приложение демонстрирует корректное взаимодействие с серверной частью и отвечает требованиям наглядности и функциональности. Полученный опыт работы с Vue.js и Axios позволяет закрепить знания о создании клиент-серверных приложений и взаимодействии с REST API.