|  |  |
| --- | --- |
| 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 **СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**Отчет по лабораторной работе № 5**

**по дисциплине Программирование сетевых приложений**

#### Тема работы: "Простое веб-приложение. Работа VK Api (Продолжение)"

**Выполнил:**

Студент группы ИУ5Ц-54Б

Перевощиков Н.Д.

(дата, подпись)

**Проверила:**

Преподаватель

Аксёнова М.В.

(дата, подпись)

Москва, 2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ ОТЧЕТА**

[1. Цель лабораторной работы 3](#_Toc115088672)

[2. Описание задания 3](#_Toc115088673)

[3. Текст программы 4](#_Toc115088674)

[4. Результат 12](#_Toc115088675)

## Цель лабораторной работы

Целью данной лабораторной работы также является взаимодействие с внешним Api. Но если в прошлой лабораторной работе мы это делали с помощью XMLHttpRequest, то в этот раз мы будем пользоваться более современным "инструментом" - fetch.

## Описание задания

1. Введение в Promise
2. Использование Promise
3. Что такое async await в JS
4. Пояснение про fetch
5. Пример использования fetch

## Текст программы

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Simple App</title>

    <link rel="stylesheet" href="node\_modules/bootstrap/dist/css/bootstrap.min.css">

</head>

<body>

<div id="root"></div>

<script src="main.js" type="module"></script>

<script src="node\_modules/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**Main.js**

//Добавляем вызов файла

import {MainPage} from "./pages/main/index.js";

const root = document.getElementById('root');

const mainPage = new MainPage(root);

mainPage.render();

**Папка components**

**Папка back-button**

**Index.js**

//Создаем компонент

export class BackButtonComponent {

    constructor(parent) {

        this.parent = parent;

    }

    addListeners(listener) {

        document

            .getElementById("back-button")

            .addEventListener("click", listener)

    }

    getHTML() {

        return (

            `

                <button id="back-button" class="btn btn-primary" type="button">Назад</button>

            `

        )

    }

    render(listener) {

        const html = this.getHTML()

        this.parent.insertAdjacentHTML('beforeend', html)

        this.addListeners(listener)

    }

}

**Папка button**

**Index.js**

//Создаем наш компонент

export class ButtonComponent {

    constructor(parent) {

        this.parent = parent;

    }

    render() {

        this.parent.insertAdjacentHTML('beforeend', '<button type="button" class="btn btn-primary">Hello world 4!</button>');

    }

}

**Папка product**

**Index.js**

//Создаем компонент продукта

export class ProductComponent {

    constructor(parent) {

        this.parent = parent

        console.log('pr', parent)

    }

    //Изменяем функцию получения данных

    getData() {

        ajax.post(urls.getUserInfo(this.id), (data) => {

            this.renderData(data.response)

        })

    }

    //Добавляем функцию отрисовки карточек по данным

    renderData(item) {

        const product = new ProductComponent(this.pageRoot)

        product.render(item[0])

    }

    //Нам нужно поменять структуру компонента ProductComponent, так же как мы сделали с компонентом ProductCardComponent

    getHTML(data) {

        return (

            `

                <div class="card mb-3" style="width: 540px;">

                    <div class="row g-0">

                        <div class="col-md-4">

                            <img src="${data.src}" class="img-fluid" alt="картинка">

                        </div>

                        <div class="col-md-8">

                            <div class="card-body">

                                <h5 class="card-title">${data.first\_name} ${data.last\_name}</h5>

                                <h5 class="card-title">${data.country}</h5>

                                <h5 class="card-title">${data.status}</h5>

                            </div>

                        </div>

                    </div>

                </div>

            `

        )

    }

    //Модифицируем функцию отрисовки страницы

    render(data) {

        debugger;

        console.log('this.data\_info.id:', data)

        const html = this.getHTML(data)

        this.parent.insertAdjacentHTML('beforeend', html)

        //const backButton = new BackButtonComponent(this.pageRoot)

        //backButton.render(this.clickBack.bind(this))

        //this.getData()

    }

}

**Папка product-card**

**Index.js**

//Создаем компонент карточки

export class ProductCardComponent {

    constructor(parent) {

        this.parent = parent;

    }

    getHTML(data) {

        return (

            `

                <div class="card-${data.id}" style="width: 300px;">

                    <img class="card-img-top" src="${data.photo\_400\_orig}" alt="картинка">

                    <div class="card-body">

                        <h5 class="card-title">${data.first\_name} ${data.last\_name}</h5>

                        <button class="btn btn-primary" id="click-card-${data.id}" data-id="${data.id}">Нажми на меня</button>

                    </div>

                </div>

            `

        )

    }

    addListeners(data, listener) {

        document

            .getElementById(`click-card-${data.id}`)

            .addEventListener("click", listener)

    }

    render(data, listener) {

        const html = this.getHTML(data)

        this.parent.insertAdjacentHTML('beforeend', html)

        this.addListeners(data, listener)

    }

}

**Папка modules**

**Ajax.js**

class Ajax {

    debugger;

    post(url, callback) {

        // console.log('url:', url);

        fetch(url, {method: 'POST'})

            .then(response => response.json())

            .then(data => {

                callback(data);

                // console.log(data);

            });

    }

}

export const ajax = new Ajax();

**Consts.js**

export const groupId = GROUP\_ID

export const accessToken = 'ACCESS\_TOKEN'

export const version = '5.131'

groupId – id группы, которую мы создали

accessToken – секретный ключ нашей группы

version – версия VK API

**Urls.js**

import {accessToken, version} from "./consts.js";

class Urls {

    constructor() {

        this.url = 'https://api.vk.com/method'

        this.commonInfo = `access\_token=${accessToken}&v=${version}`

    }

    getUserInfo(userId) {

        return `${this.url}/users.get?user\_ids=${userId}&fields=photo\_400\_orig&${this.commonInfo}`

    }

    getGroupMembers(groupId) {

        return `${this.url}/groups.getMembers?group\_id=${groupId}&fields=photo\_400\_orig, country, status&${this.commonInfo}`

    }

    getGroupID(groupId) {

        return `${this.url}/groups.getMembers?group\_id=${groupId}&${this.commonInfo}`

    }

}

export const urls = new Urls()

**Папка pages**

**Папка main**

**Index.js**

//import {ButtonComponent} from "../../components/button/index.js";

import {ProductCardComponent} from "../../components/product-card/index.js";

//import {ProductComponent} from "../../components/product/index.js";

import {ProductPage} from "../product/index.js";

//import {BackButtonComponent} from "../../components/back-button/index.js";

import {ajax} from "../../modules/ajax.js";

import {urls} from "../../modules/urls.js";

import {groupId} from "../../modules/consts.js";

// Создаем нашу страницу

export class MainPage {

    constructor(parent, data\_info) {

        this.parent = parent;

        this.data\_info = data\_info

    }

    getData() {

        debugger

        ajax.post(urls.getGroupMembers(groupId), (data) => {

            this.renderData(data.response.items)

            this.data\_info = data.response.items

            console.log(this.data\_info)

        })

    }

    renderData(items) {

        items.forEach((item) => {

            const productCard = new ProductCardComponent(this.pageRoot)

            productCard.render(item, this.clickCard.bind(this))

        })

    }

    get pageRoot() {

        return document.getElementById('main-page')

    }

    getHTML() {

        return (

            `

                <div id="main-page" class="d-flex flex-wrap"><div/>

            `

        )

    }

    clickCard(e) {

        debugger;

        const cardId = e.target.dataset.id

        const press = undefined;

        Object.entries(this.data\_info).forEach(([key, value]) => {

            console.log(key, value.id);

            if(value.id == cardId){

                // press = this.data\_info.id;

                this.data\_info = value;

                // console.log('press:', press)

            }

        });

        console.log('this.data\_info:', this.data\_info)

        const productPage = new ProductPage(this.parent, cardId, this.data\_info)

        productPage.render()

        console.log('cardId:', cardId)

        console.log('this.data\_info.id:', this.data\_info)

    }

    render() {

        debugger

        this.parent.innerHTML = ''

        const html = this.getHTML()

        this.parent.insertAdjacentHTML('beforeend', html)

        this.getData()

    }

}

**Папка product**

**Index.js**

import {ProductComponent} from "../../components/product/index.js";

import {BackButtonComponent} from "../../components/back-button/index.js";

import {MainPage} from "../main/index.js";

//Создаем страницу продукта

export class ProductPage {

    constructor(parent, id, data) {

        this.parent = parent

        this.id = id

        this.data = data

        console.log('ProductPage data:', data)

    }

    getData() {

        return {

            id: 1,

            src: `${this.data.photo\_400\_orig}`,

            title: `Акция ${this.data.id}`,

            first\_name: this.data.first\_name,

            last\_name: this.data.last\_name,

            country: this.data.country.title,

            status: this.data.status,

            text: `Такой акции вы еще не видели`

        }

    }

    get pageRoot() {

        return document.getElementById('product-page')

    }

    getHTML() {

        return (

            `

                <div id="product-page"></div>

            `

        )

    }

    clickBack() {

        const mainPage = new MainPage(this.parent)

        mainPage.render()

    }

    render() {

        this.parent.innerHTML = ''

        const html = this.getHTML()

        this.parent.insertAdjacentHTML('beforeend', html)

        const backButton = new BackButtonComponent(this.pageRoot)

        backButton.render(this.clickBack.bind(this))

        const data = this.getData()

        const stock = new ProductComponent(this.pageRoot)

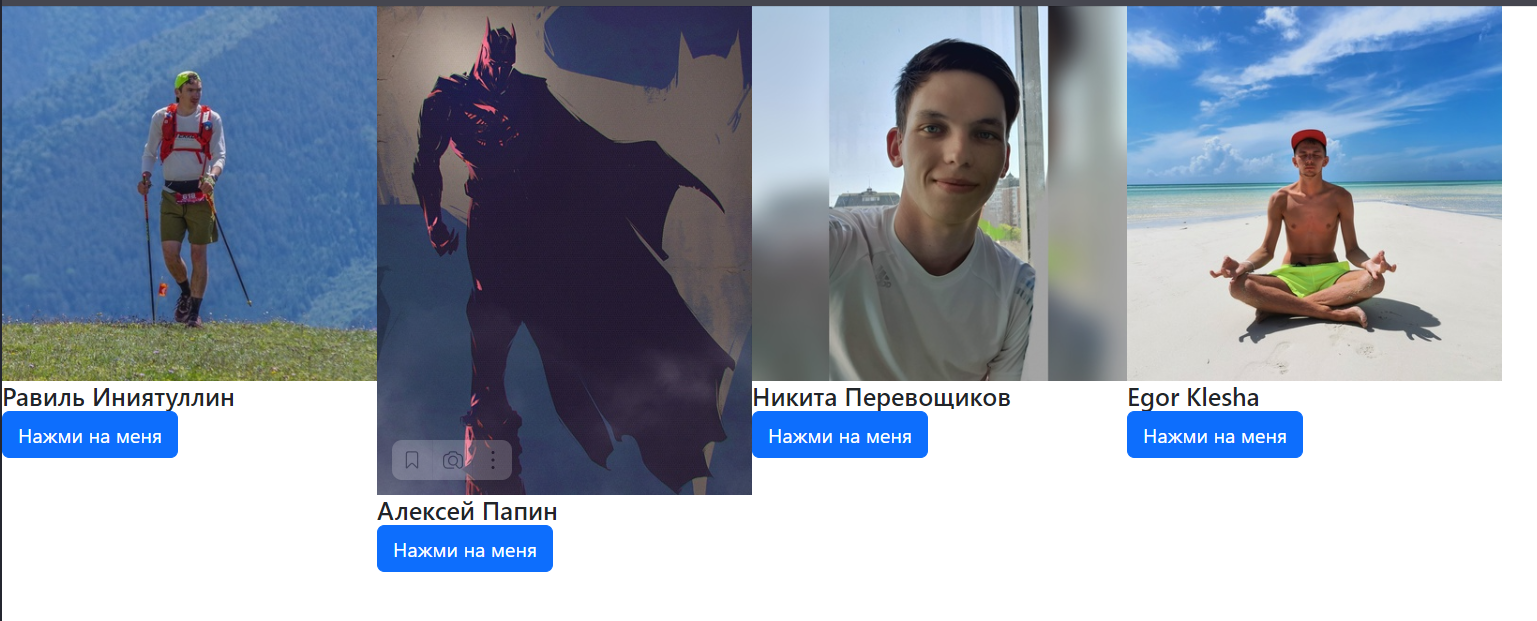
        stock.render(data)

    }

}

## Результат

Выводит на экран карточки подписчика из сообшества Вконтакте.



При нажатии на кнопку «Нажмите на меня» на Равиля или меня, выводит карточка с информацией о ФИО подписчика, страны и статуса.

