|  |  |
| --- | --- |
|  | **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 №7

Студент Компаниец Андрей Александрович

*фамилия, имя, отчество*

Группа ИУ5-51Б.

Студент 15.01.2022 **Компаниец А.А.**

*подпись, дата фамилия, и.о.*

Преподаватель   *подпись, дата фамилия, и.о.*

2022г.

**Задание:**

На основе методических указаний разработать пользовательский интерфейс для работы с REST API. Использовать REST API, разработанный в предыдущей лабораторной работе.

**Код:**

*index.js:*

'use strict';  
  
const ***express*** = require('express');  
const ***path*** = require('path');  
const app = ***express***();  
const htmlPath = ***path***.resolve(\_\_dirname, '..', 'src/index.html')  
const staticPath = ***path***.resolve(\_\_dirname, '..', 'src')  
const port = 3000;  
  
app.use('/', ***express***.static(staticPath));  
  
app.all('\*', (req, res) => {  
 res.sendFile(htmlPath);  
});  
  
app.listen(port, () => {  
 ***console***.log(`Server listening http://localhost:${port}`);  
});

*main.js*

import {MainPage} from "./pages/main/MainPage.js";  
  
const root = ***document***.getElementById('root');  
  
const mainPage = new MainPage(root)  
mainPage.render()

*MainPage.js:*

import {ButtonComponent} from "../../components/button/ButtonComponent.js";  
import {StockCardComponent} from "../../components/stock-card/StockCardComponents.js";  
import {StockPage} from "../stock/StockPage.js";  
import {***ajax***} from "../../modules/ajax.js";  
import {***urls***} from "../../modules/urls.js";  
  
export class MainPage {  
 constructor(parent) {  
 this.parent = parent;  
 }  
  
 get page() {  
 return ***document***.getElementById('main-page')  
 }  
  
 getHTML() {  
 return (  
 `  
 <div id="main-page" class="d-flex flex-wrap"><div/>  
 `  
 )  
 }  
  
  
 async getData() {  
 return ***ajax***.get(***urls***.stocks())  
 }  
  
 clickCard(e) {  
 const cardId = e.target.dataset.id  
  
 const stockPage = new StockPage(this.parent, cardId)  
 stockPage.render()  
 }  
  
 async render() {  
 this.parent.innerHTML = ''  
 const html = this.getHTML()  
 this.parent.insertAdjacentHTML('beforeend', html)  
  
 const data = await this.getData()  
 data.data.forEach((item) => {  
 const stockCard = new StockCardComponent(this.page)  
 stockCard.render(item, this.clickCard.bind(this))  
 })  
 }  
  
}

*StockPage.js:*

import {StockComponent} from "../../components/stock/StockComponent.js";  
import {BackButtonComponent} from "../../components/back-button/BackButtonComponent.js";  
import {MainPage} from "../main/MainPage.js";  
import {***ajax***} from "../../modules/ajax.js";  
import {***urls***} from "../../modules/urls.js";  
  
export class StockPage {  
 constructor(parent, id) {  
 this.parent = parent  
 this.id = id  
 }  
  
 async getData() {  
 return ***ajax***.get(***urls***.stock(this.pk))  
 }  
  
 get page() {  
 return ***document***.getElementById('stock-page')  
 }  
  
 getHTML() {  
 return (  
 `  
 <div id="stock-page">  
 </div>  
 `  
 )  
 }  
  
 clickBack() {  
 const mainPage = new MainPage(this.parent)  
 mainPage.render()  
 }  
  
  
 async render() {  
 this.parent.innerHTML = ''  
 const html = this.getHTML()  
 this.parent.insertAdjacentHTML('beforeend', html)  
  
 const backButton = new BackButtonComponent(this.page)  
 backButton.render(this.clickBack.bind(this))  
  
 const data = this.getData()  
 const stock = new StockComponent(this.page)  
 stock.render(data.data)  
 }  
   
}

*BackButtonComponent.js:*

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)  
 }  
}

*ButtonComponent.js:*

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

*StockComponent.js:*

export class StockComponent {  
 constructor(parent) {  
 this.parent = parent  
 }  
  
 getHTML(data) {  
 return (  
 `  
 <div class="card mb-3" style="width: 540px;">  
 <div class="row g-0">  
 <div class="col-md-4">  
 <img src="https://kuda-sochi.ru/uploads/7a3c046e1230dc19840256889c33987d.jpg" class="img-fluid" alt="картинка">  
 </div>  
 <div class="col-md-8">  
 <div class="card-body">  
 <h5 class="card-title">${data.company\_name}</h5>  
 <p class="card-text">Рейтинг: ${data.price}</p>  
 </div>  
 </div>  
 </div>  
 </div>  
 `  
 )  
 }  
  
 render(data) {  
 ***console***.log(data)  
 const html = this.getHTML(data)  
 this.parent.insertAdjacentHTML('beforeend', html)  
 }  
}

*StockCardComponent.js:*

export class StockCardComponent {  
 constructor(parent) {  
 this.parent = parent;  
 }  
 getHTML(data) {  
 return (  
 `  
 <div class="card" style="width: 300px;">  
 <img class="card-img-top" src="https://kuda-sochi.ru/uploads/7a3c046e1230dc19840256889c33987d.jpg" alt="картинка">  
 <div class="card-body">  
 <h5 class="card-title">${data.company\_name}</h5>  
 <p class="card-text">Рейтинг: ${data.price}</p>  
 <button class="btn btn-primary" id="click-card-${data.pk}" data-id="${data.pk}">Нажми на меня</button>  
 </div>  
 </div>  
 `  
 )  
 }  
 addListeners(data, listener) {  
 ***document*** .getElementById(`click-card-${data.pk}`)  
 .addEventListener("click", listener)  
 }  
  
 render(data, listener) {  
 const html = this.getHTML(data)  
 this.parent.insertAdjacentHTML('beforeend', html)  
 this.addListeners(data, listener)  
 }  
}

*ajax.js:*

class Ajax {  
 async get(url) {  
 const response = await fetch(url, {  
 method: "GET"  
 });  
 const responseData = await response.json();  
  
 return {  
 status: response.status,  
 data: responseData  
 };  
 }  
}  
  
export const ***ajax*** = new Ajax();

*urls.js:*

class Urls {  
 constructor() {  
 this.url = 'http://localhost:8000/';  
 }  
  
 stocks() {  
 return `${this.url}stocks/`  
 }  
  
 stock(id) {  
 return `${this.url}stocks/${id}/`  
 }  
}  
  
export const ***urls*** = new Urls()

**Демонстрация:**





