**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка Интернет-Приложений»

Отчет по лабораторной работе №7

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-53Б |  | Юрий Евгеньевич Гапанюк |
| Ветошкин Артём |  |  |
| Подпись и дата:  20.12.21 |  | Подпись и дата: |

Москва, 2021 г.

Цель лабораторной работы: изучение возможностей создания пользовательского интерфейса в веб-приложениях. Задание: разработать пользовательский интерфейс для работы с REST API. Использовать REST API, разработанный в 6 лабораторной работе.

**Выполнение задания лабораторной работы**

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}`);  
});

button.js

export class ButtonComponent {  
 constructor(parent, text, id, click\_event) {  
 this.parent = parent;  
 this.text = text;  
 this.id = id  
 this.event = click\_event  
 }  
  
 getHTML(text) {  
 return (  
 `  
 <button type="button" id="click-card-${this.id}" data-id="${this.id}" class="btn btn-primary">${text}</button>  
 `  
 )  
 }  
  
 render() {  
 this.parent.insertAdjacentHTML('beforeend', this.getHTML(this.text))  
 ***document***.getElementById(`click-card-${this.id}`).addEventListener("click", this.event)  
 }  
}

stocks.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-8">  
 <div class="card-body">  
 <h5 class="card-title">${data.company\_name}</h5>  
 <p class="card-text">${data.price}</p>  
 <p class="card-text">${data.is\_growing}</p>  
 <p class="card-data">${data.date\_modified}</p>  
 </div>  
 </div>  
 </div>  
 </div>  
 `  
 )  
 }  
  
 render(data) {  
 const html = this.getHTML(data)  
 this.parent.insertAdjacentHTML('beforeend', html)  
 }  
}

stock\_card.js

import {ButtonComponent} from "./button.js";  
import {StockPage} from "../pages/stock\_page.js";  
  
export class StockCardComponent {  
 constructor(parent) {  
 this.parent = parent;  
 }  
  
 getHTML(data) {  
 return (  
 `  
 <div class="card" style="width: 300px;">  
 <div id = "card-${data.pk}" class="card-body">  
 <h5 class="card-title">${data.company\_name}</h5>  
 <p class="card-text">${data.price}</p>  
 </div>  
 </div>  
 `  
 )  
 }  
  
 clickCard(e) {  
 const cardId = e.target.dataset.id  
  
 const stockPage = new StockPage(this.parent, cardId)  
 stockPage.render()  
 }  
  
 render(data) {  
 const html = this.getHTML(data)  
 this.parent.insertAdjacentHTML('beforeend', html)  
 let before\_button = ***document***.getElementById(`card-${data.pk}`)  
 let button = new ButtonComponent(before\_button, "Подробнее", data.pk, this.clickCard.bind(this))  
 button.render()  
 }  
}

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()

main\_page.js

import {StockCardComponent} from "../components/stock\_card.js";  
import {ajax} from "../modules/ajax.js";  
import {urls} from "../modules/urls.js";  
  
export class MainPage {  
 constructor(parent) {  
 this.parent = parent;  
 }  
  
 getData() {  
 return ***ajax***.get(***urls***.stocks())  
 }  
  
 get page() {  
 return ***document***.getElementById('main-page')  
 }  
  
 getHTML() {  
 return (  
 `  
 <div id="main-page" class="d-flex flex-wrap"><div/>  
 `  
 )  
 }  
  
 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)  
 })  
 }  
}

stock\_page.js

import {StockComponent} from "../components/stock.js";  
import {ButtonComponent} from "../components/button.js";  
import {MainPage} from "./main\_page.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.id))  
 }  
  
 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)  
 let before\_button = ***document***.getElementById(`stock-page`)  
 let button = new ButtonComponent(before\_button, "Назад", '', this.clickBack.bind(this))  
 button.render()  
  
 const data = (await this.getData()).data  
 const stock = new StockComponent(this.page)  
 stock.render(data)  
 }  
}

index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Lab7</title>  
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css">  
</head>  
<div id="root">  
</div>  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>  
<script src="main.js" type="module"></script>  
</html>

main.js

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

**Пример работы программы:**



