|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю.Е.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |  | Демонстрация:  Гапанюк Ю.Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

**Отчет по лабораторной работе № 7 по курсу**

**Разработка интернет-приложений**

**ГУИМЦ**

#### Тема работы: " Разработка пользовательского интерфейса для работы с REST API "

11

(количество листов)

Вариант № **4**

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-73Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Фоминский Г. В. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г. |

Москва, МГТУ - 2021

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

**СОДЕРЖАНИЕ**

[1. Описание задания 3](#_Toc89607885)

[2. Текст программы 3](#_Toc89607886)

[3. Результаты выполнения программы 10](#_Toc89607887)

# Описание задания

На основе [методических указаний](https://github.com/iu5team/iu5web-fall-2021/blob/main/tutorials/lab7/lab7_tutorial.md) разработайте пользовательский интерфейс для работы с REST API. Используйте REST API, разработанный Вами в предыдущей лабораторной работе.

# 2. Текст программы

**package.json**

{  
 "name": "lab\_7",  
 "version": "1.0.0",  
 "description": "",  
 "main": "index.js",  
 "scripts": {  
 "test": "echo \"Error: no test specified\" && exit 1"  
 },  
 "keywords": [],  
 "author": "",  
 "license": "ISC",  
 "devDependencies": {  
 "express": "^4.17.1"  
 }  
}

**server/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}`);  
});

**src/index.html**

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <title>ЛР №7</title>  
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css">  
</head>  
<body>  
<div id="root"></div>  
<script src="main.js" type="module"></script>  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>  
</body>  
</html>

**src/main.js**

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

**src/components/animal/AnimalComponent.js**

export class AnimalComponent {  
 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="${data.animal\_photo}" class="img-fluid" alt="фото">  
 </div>  
 <div class="col-md-8">  
 <div class="card-body">  
 <h5 class="card-title">${data.animal\_name}</h5>  
 <p class="card-text">${data.animal\_type}</p>  
 </div>  
 </div>  
 </div>  
 </div>  
 `  
 )  
 }  
  
 render(data) {  
 const html = this.getHTML(data)  
 this.parent.insertAdjacentHTML('beforeend', html)  
 }  
}

**src/components/animal-card/AnimalCardComponent.js**

export class AnimalCardComponent {  
 constructor(parent) {  
 this.parent = parent;  
 }  
  
 getHTML(data) {  
 return (  
 `  
 <div class="card" style="width: 300px;">  
 <img class="card-img-top" src="${data.animal\_photo}" alt="картинка">  
 <div class="card-body">  
 <h5 class="card-title">${data.animal\_name}</h5>  
 <p class="card-text">${data.animal\_type}</p>  
 <button class="btn btn-primary" id="click-card-${data.id}" data-id="${data.id}">Подробнее</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)  
 }  
}

**src/components/back-button/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)  
 }  
}

**src/components/button/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>')  
 }  
}

**src/modules/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();

**src/modules/urls.js**

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

**src/pages/animal/AnimalPage.js**

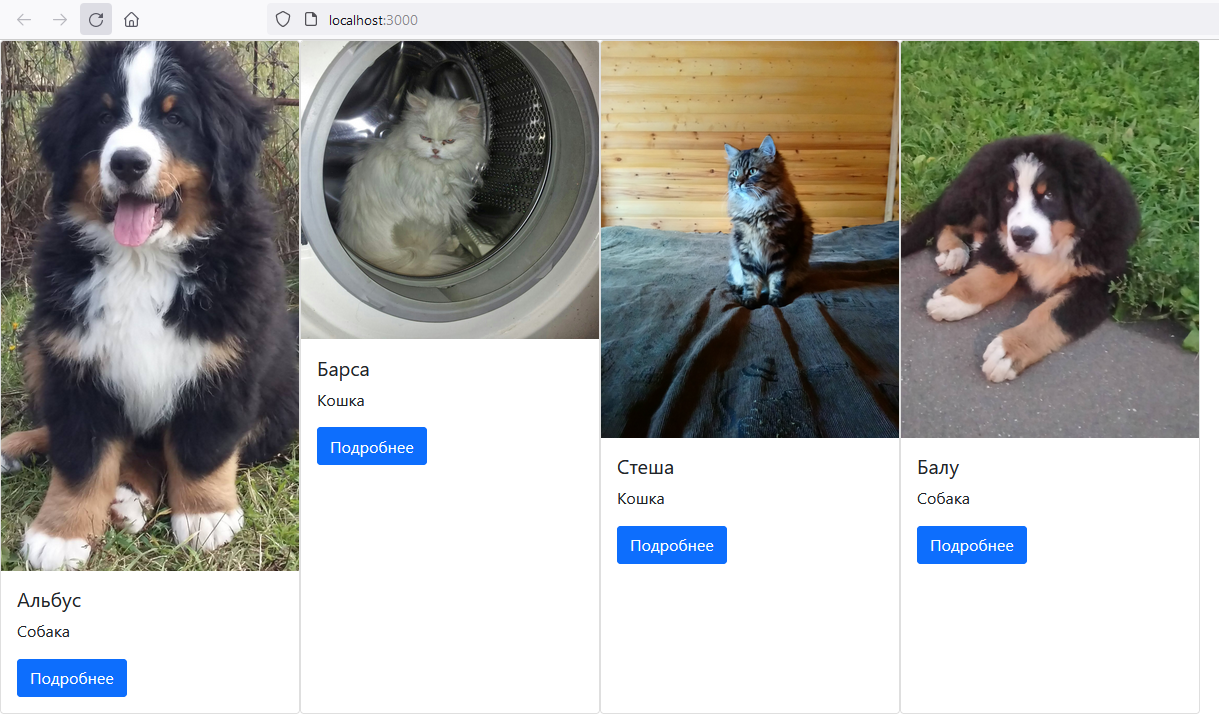
import {AnimalComponent} from "../../components/animal/AnimalComponent.js";  
import {BackButtonComponent} from "../../components/back-button/BackButtonComponent.js";  
import {MainPage} from "../main/MainPage.js";  
import {ajax} from "../../modules/ajax.js";  
import {urls} from "../../modules/urls.js";  
  
export class AnimalPage {  
 constructor(parent, id) {  
 this.parent = parent  
 this.id = id  
 }  
  
 async getData() {  
 return ajax.get(urls.animal(this.id))  
 }  
  
 get page() {  
 return document.getElementById('animal-page')  
 }  
  
 getHTML() {  
 return (  
 `  
 <div id="animal-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 animal = new AnimalComponent(this.page)  
 animal.render((await data).data)  
 }  
}

**src/pages/main/MainPage.js**

import {AnimalCardComponent} from "../../components/animal-card/AnimalCardComponents.js";  
import {AnimalPage} from "../animal/AnimalPage.js";  
import {ajax} from "../../modules/ajax.js";  
import {urls} from "../../modules/urls.js";  
  
export class MainPage {  
 constructor(parent) {  
 this.parent = parent;  
 }  
  
 async getData() {  
 return ajax.get(urls.animals())  
 }  
  
 get page() {  
 return document.getElementById('main-page')  
 }  
  
 getHTML() {  
 return (  
 `  
 <div id="main-page" class="d-flex flex-wrap"><div/>  
 `  
 )  
 }  
  
 clickCard(e) {  
 const cardId = e.target.dataset.id  
   
 const animalPage = new AnimalPage(this.parent, cardId)  
 animalPage.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 AnimalCard = new AnimalCardComponent(this.page)  
 AnimalCard.render(item, this.clickCard.bind(this))  
 })  
 }  
}

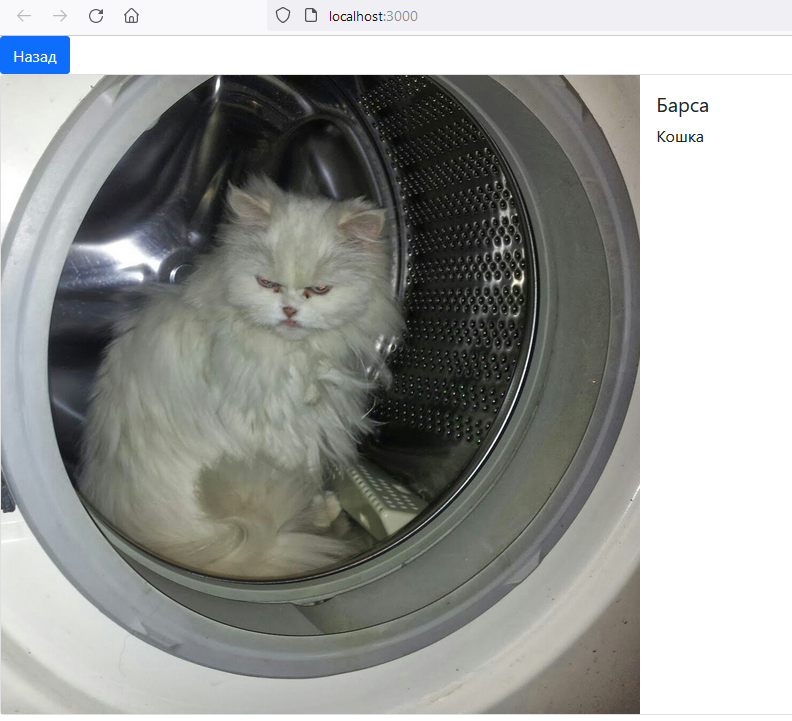
# Результаты выполнения программы

**Главная страница**



Содержит список животных, полученных из API.

**Страница животного**



Содержит данные о выбранном животном. Фото приведено в большем размере (по сравнению с карточкой в списке).