

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №7
«Разработка пользовательского интерфейса для работы с REST API»

Выполнил:
студент группы РТ5-51Б
Борисочкин М.И.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Балашов А. М.
Подпись и дата:

Москва, 2021 г.

Описание задания

1. На основе методических указаний разработайте пользовательский интерфейс для работы с REST API. Используйте REST API, разработанный Вами в предыдущей лабораторной работе.

Текст программы

src\index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Lab 7</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.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.jsdelivrivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.mi
n.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\pages\main\MainPage.js

```
import {PizzaCardComponent} from "../../components/pizza-card/PizzaCardComponent.js";
import {PizzaPage} from "../../pizza/PizzaPage.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.pizzas());
  }

  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 pizzaPage = new PizzaPage(this.parent, cardId);
    pizzaPage.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 pizzaCard = new PizzaCardComponent(this.page);
      pizzaCard.render(item, this.clickCard.bind(this));
    });
  }
}
```

src\components\pizza-card\PizzaCardComponent.js

```
export class PizzaCardComponent {
  constructor(parent) {
    this.parent = parent;
  }

  getHTML(data) {
    return (
      `<div class="card" style="width: 300px;">
        
        <div class="card-body">
          <h5 class="card-title">${data.name}</h5>
          <p class="card-text">${data.topping}</p>
          <button class="btn btn-primary" id="click-card-
${data.pk}" data-id="${data.pk}">
            Подробнее
          </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);
  }
}
```

src\pages\pizza\PizzaPage.js

```
import {PizzaComponent} from "../../components/pizza/PizzaComponent.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 PizzaPage {
  constructor(parent, id) {
    this.parent = parent;
    this.id = id;
  }

  async getData() {
    return ajax.get(urls.pizza(this.id));
  }

  get page() {
    return document.getElementById('pizza-page');
  }

  getHTML() {
    return (`<div id="pizza-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 = await this.getData();
    const pizza = new PizzaComponent(this.page);
    pizza.render(data.data);
  }
}
```

src/components/pizza/PizzaComponent.js

```
export class PizzaComponent {
  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">
            
          </div>
          <div class="col-md-8">
            <div class="card-body">
              <h5 class="card-title">${data.name}</h5>
              <p class="card-subtitle">Состав:</p>
              <p class="card-text">${data.topping}</p>
              <p class="card-subtitle">Диаметр:</p>
              <p class="card-text">${data.diameter}</p>
            </div>
          </div>
        </div>
      </div>
    `;
  }

  render(data) {
    const html = this.getHTML(data);
    this.parent.insertAdjacentHTML('beforeend', html);
  }
}
```

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">На главную</button>`;
  }

  render(listener) {
    const html = this.getHTML();
    this.parent.insertAdjacentHTML('beforeend', html);
    this.addListeners(listener);
  }
}
```

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/';
  }

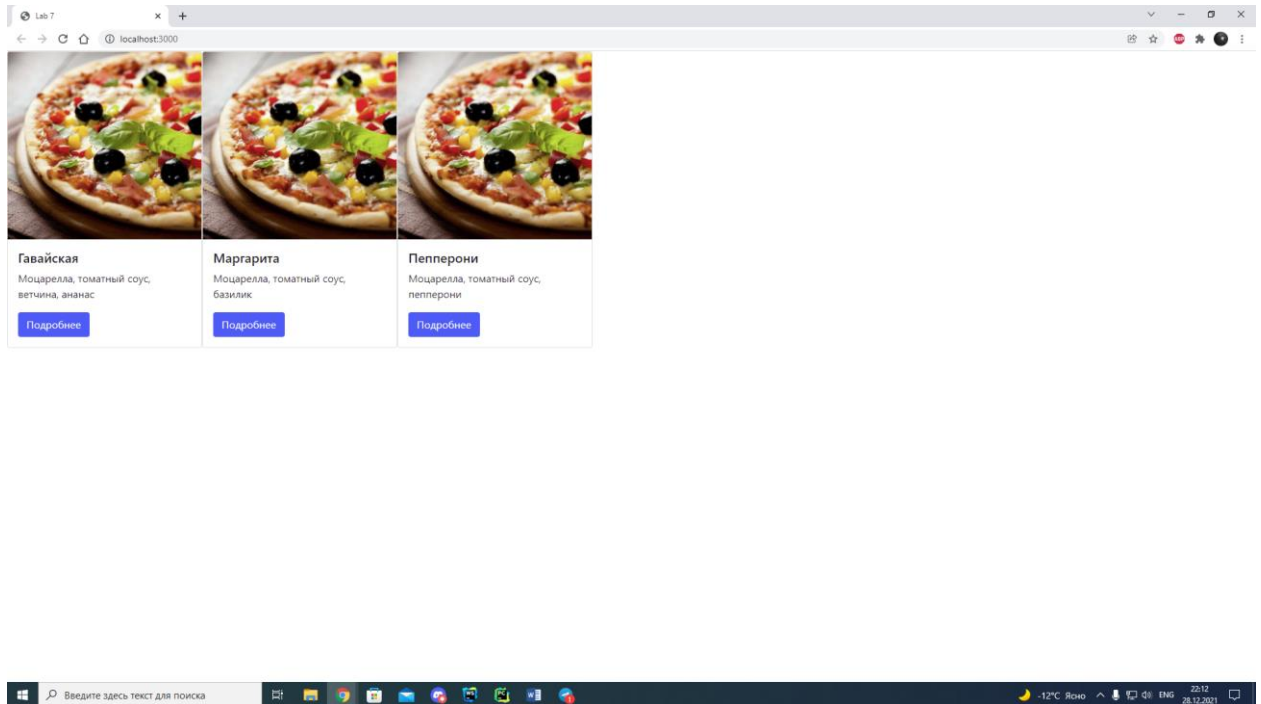
  pizzas() {
    return `${this.url}pizza/`;
  }

  pizza(id) {
    return `${this.url}pizza/${id}/`;
  }
}

export const urls = new Urls();
```

Примеры выполнения программы

Скриншот № 1 (Главная страница)



Скриншот № 2 (Страница detail)

