

Выполнил: Студент группы
ИУ5-53Б
Борисов Андрей
Михайлович

Задание

Разработка пользовательского интерфейса для работы с REST API.

Цель лабораторной работы: изучение возможностей создания пользовательского интерфейса в веб-приложениях.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

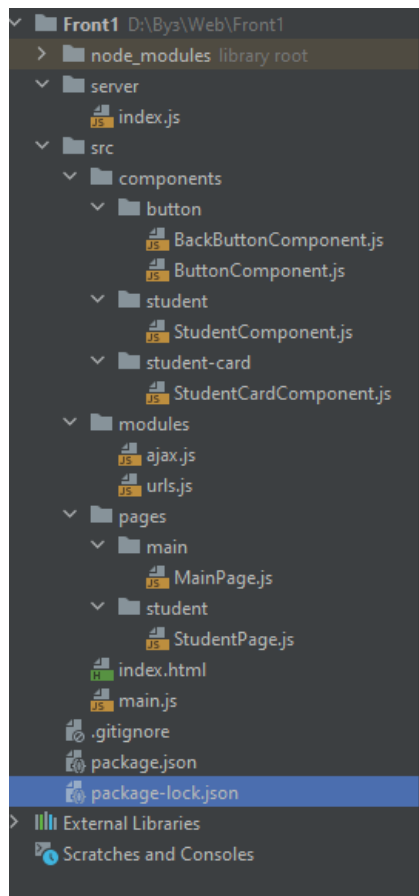
1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

Задание:

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

Текст программы

Структура проекта:



Main.js

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

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

const mainPage = new MainPage(root)
mainPage.render()
```

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>
<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>
```

BackButtonComponent.js

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

  addListeners(listener) {
    document
      .getElementById( 'back-button')
      .addEventListener( type: "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( position: 'beforeend', html)
    this.addListeners(listener)
  }
}
```

ButtonComponent.js

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

  render() {
    this.parent.insertAdjacentHTML( position: 'beforeend', text: '<button type="button" class="btn btn-primary">Hello world 4!</button>' )
  }
}
```

StudentComponent.js

```

export class StudentComponent {
  constructor(parent) {
    this.parent = parent
  }

  getHTML(data) {
    return (
      <div className="card" style="width: 700px;">
        
        <div className="card-body">
          <h5 className="card-title">Студент №${data.id}</h5>
        </div>
        <ul className="list-group list-group-flush">
          <li className="list-group-item"><strong>Имя:</strong> ${data.name}</li>
          <li className="list-group-item"><strong>Фамилия:</strong> ${data.surname}</li>
          <li className="list-group-item"><strong>Возраст:</strong> ${data.age}</li>
          <li className="list-group-item"><strong>Университет:</strong> ${data.university}</li>
          <li className="list-group-item"><strong>Курс обучения:</strong> ${data.course_number}</li>
        </ul>
      </div>
    )
  }

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

```

StudentCardComponent.js

```

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

  getHTML(data) {
    return (
      <div class="card" style="width: 300px;">
        
        <div class="card-body">
          <h5 class="card-title">Студент №{data.id}</h5>
          <button class="btn btn-primary" id="click-card-{data.id}">Перейти к анкете</button>
        </div>
        <ul class="list-group list-group-flush">
          <li class="list-group-item"><strong>Имя:</strong> {data.name}</li>
          <li class="list-group-item"><strong>Фамилия:</strong> {data.surname}</li>
          <li class="list-group-item"><strong>Возраст:</strong> {data.age}</li>
        </ul>
      </div>
    )
  }

  addListeners(data, listener) {
    document
      .getElementById( `click-card-{data.id}` )
      .addEventListener( type: "click", listener )
  }

  render(data, listener) {
    const html = this.getHTML(data)
    this.parent.insertAdjacentHTML( position: 'beforeend', html )
    this.addListeners( data, listener )
  }
}

```

Ajax.js

```

class Ajax {
  async get(url) {
    fetch(url) Promise<Response>
      .then((response : Response ) => {
        return response.json();
      }) Promise<any>
      .then((data) => {
        console.log(data);
      });

    const response = await fetch(url, {
      method: "GET"
    });

    const responseData = await response.json();

    return responseData;
  }
}

export const ajax = new Ajax();

```

Url.js

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

MainPage.js

```
export class MainPage {  
  constructor(parent) {  
    this.parent = parent;  
  }  
  
  get page() {  
    return document.getElementById( 'elementId: 'main-page')  
  }  
  
  getHTML() {  
    return (  
      <div id="main-page" class="d-flex flex-wrap"><div/>  
    )  
  }  
  
  getData() {  
    return ajax.get(urls.students())  
  }  
  
  clickCard(e) {  
    const studentPage = new StudentPage(this.parent, id: 2)  
    studentPage.render()  
  }  
  
  func(data){  
    data.forEach((item) => {  
      const studentCardComponent = new StudentCardComponent(this.page)  
      studentCardComponent.render(item, this.clickCard.bind(this))  
    })  
  }  
  
  render() {  
    this.parent.innerHTML = ''  
    const html = this.getHTML()  
    this.parent.insertAdjacentHTML( positions: 'beforeend', html)  
  
    const data = this.getData()  
  
    data.forEach((item) => {  
      const studentCardComponent = new StudentCardComponent(this.page)  
      studentCardComponent.render(item, this.clickCard.bind(this))  
    })  
  }  
}
```

StudentPge.js

```
export class StudentPage {
  constructor(parent, id) {
    this.parent = parent
    this.id = id
  }

  getData() {
    return ajax.get(urls.student(this.id))
  }

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

  getHTML() {
    return (
      <div id="stock-page">
      </div>
    )
  }



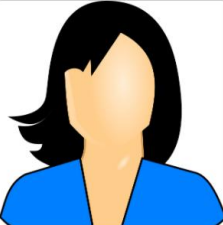



  clickBack() {
    const mainPage = new MainPage(this.parent)
    mainPage.render()
  }

  render() {
    this.parent.innerHTML = ''
    const html = this.getHTML()
    this.parent.insertAdjacentHTML( position: 'beforeend', html)

    const backButton = new BackButtonComponent(this.page)
    backButton.render(this.clickBack.bind(this))


    const student = new StudentComponent(this.page)
    const data = this.getData()
    data.then(resp => {student.render(resp)})
  }
}
```


Результаты выполнения программы

					
Студент №1	Студент №2	Студент №3	Студент №4	Студент №5	Студент №6
Перейти к анкете	Перейти к анкете	Перейти к анкете	Перейти к анкете	Перейти к анкете	Перейти к анкете
Имя: Данил	Имя: Николай	Имя: Полина	Имя: Анна	Имя: Семен	Имя: Екатерина
Фамилия: Петров	Фамилия: Иванов	Фамилия: Данилова	Фамилия: Петрова	Фамилия: Сидоров	Фамилия: Иванова
Возраст: 20	Возраст: 19	Возраст: 20	Возраст: 21	Возраст: 20	Возраст: 20

← WebSt... localhost:3000

Назад

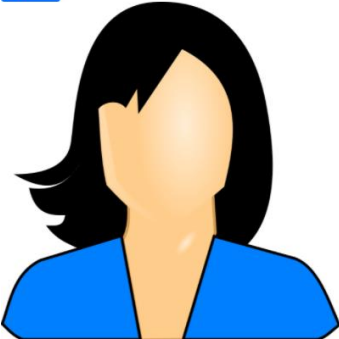


Студент №2

- Имя: Николай
- Фамилия: Иванов
- Возраст: 19
- Университет: МГТУ им. Н.Э.Баумана
- Курс обучения: 1

← WebSt... localhost:3000

Назад



Студент №4

- Имя: Анна
- Фамилия: Петрова
- Возраст: 21
- Университет: МЭИ
- Курс обучения: 4

← WebSt... localhost:3000

Назад



Студент №1

- Имя: Данил
- Фамилия: Петров
- Возраст: 20
- Университет: МГТУ им. Н.Э.Баумана
- Курс обучения: 3