



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 РАДИОТЕХНИЧЕСКИЙ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

Студент Столярова Ольга Денисовна  
*фамилия, имя, отчество*

Группа РТ5-51Б

Название предприятия МГТУ им. Н. Э. Баумана

Студент Столярова О.Д.

Преподаватель Гапанюк Ю.Е.

2021 г.

## Цель работы

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

## Задание

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

## Пример выполнения

<http://localhost:3000/>

<b>Ангел</b> Чистые, непорочные создания.Противоположность демонов. <a href="#">Нажми на меня</a>	<b>Демон</b> Эти личности, как вы прекрасно знаете, являются адскими созданиями.Однако, даже несмотря на то, что они являются таковыми, не у всех кровожадный характер.Все они разные. <a href="#">Нажми на меня</a>	<b>Оборотень</b> Мифические существа, люди, способные временно менять свой облик магическим путём, превращаясь из человека в животное и наоборот. <a href="#">Нажми на меня</a>	<b>Фея</b> Многие из нас знают, кто такие феи и с чем их едят. Фей много, на любой вкус и цвет. Одна фея владеет только одним видом магии. Фея Пламени никогда не сможет использовать магию Воды. <a href="#">Нажми на меня</a>
---	--	---	---

[Назад](#)

**Ангел**

Чистые, непорочные создания.Противоположность демонов.

## Код программы

MainPage.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;
  }

  async 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/>
  )
}

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))
  })
}
}

```

## MainPage.js

```

import { StockComponent } from "../../components/stock/StockComponent.js";
import { BackButtonComponent } from "../../components/back-button/BackButtonComponent.js";
import { MainPage } from "../../pages/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.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)

  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)
}
}

```

## Back-button.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)
  }
}

```

## StockCardComponent.js

```

export class StockCardComponent {
  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.description}</p>
        <button class="btn btn-primary" id="click-card-${data.id}" data-id="${data.id}">Нажми на
меня</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)
}
}

```

## 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-8">
                        <div class="card-body">
                            <h5 class="card-title">${data.name}</h5>
                            <p class="card-text">${data.description}</p>
                        </div>
                    </div>
                </div>
            </div>
        )
    }

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

```

## 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()
```

## Ссылка на GitHub

<https://github.com/OlyaSto/Olyabmstu>