**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа № 6

по дисциплине: Технологии Web-программирования

тема: «Работы с HTTP запросами»

Выполнил: ст. группы ПВ-42

Котухов Д. А.

Проверил: Картамышев С.В.

Белгород 2020

**Цель**: изучить принципы работы отправки Ajax запросов. Изучить работу отправки синхронных и асинхронных запросов. Получить навыки работы с промисами.

**Задание к лабораторной работе:**

1. Изучить возможности для отправки http запросов.
2. Выбрать подходящую библиотеку для работы с запросами.
3. Реализовать взаимодействие фронтенда с REST API, спроектированном в прошлой лабораторной работе.
4. Продемонстрировать работу взаимодействия фронтенд приложения с REST API.

**Ход работы**

**Переход на бэк (login)**

import React from 'react';  
import ***history*** from "../../history";  
  
import './style.css';  
  
const LoginPage = () => {  
 function handleClick() {  
 ***console***.log("Нажал");  
 ***window***.location.href = "http://localhost:8090/oauth2/authorize/vk";  
 }  
  
 return (  
 <div className="wrapper">  
 <div className="col-md-6">  
 <form className="form-horizontal">  
 <span className="heading">АВТОРИЗАЦИЯ ВК</span>  
 <div className="main-btn form-group">  
 <button onClick={handleClick} type="button" className="btn btn-default">ВХОД</button>  
 </div>  
 </form>  
 </div>  
 </div>)  
}  
  
export default LoginPage;

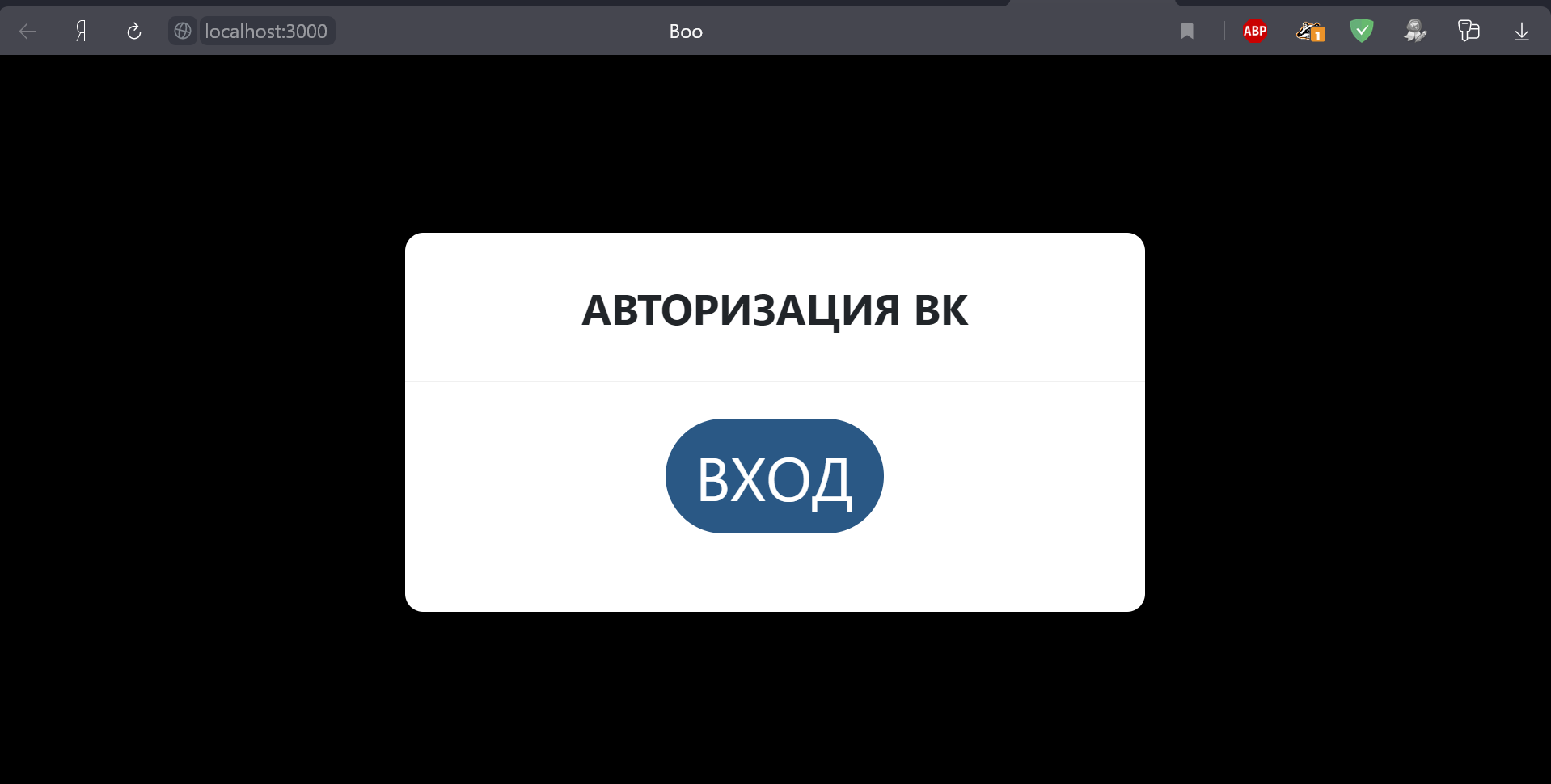
**Запрос авторизации**

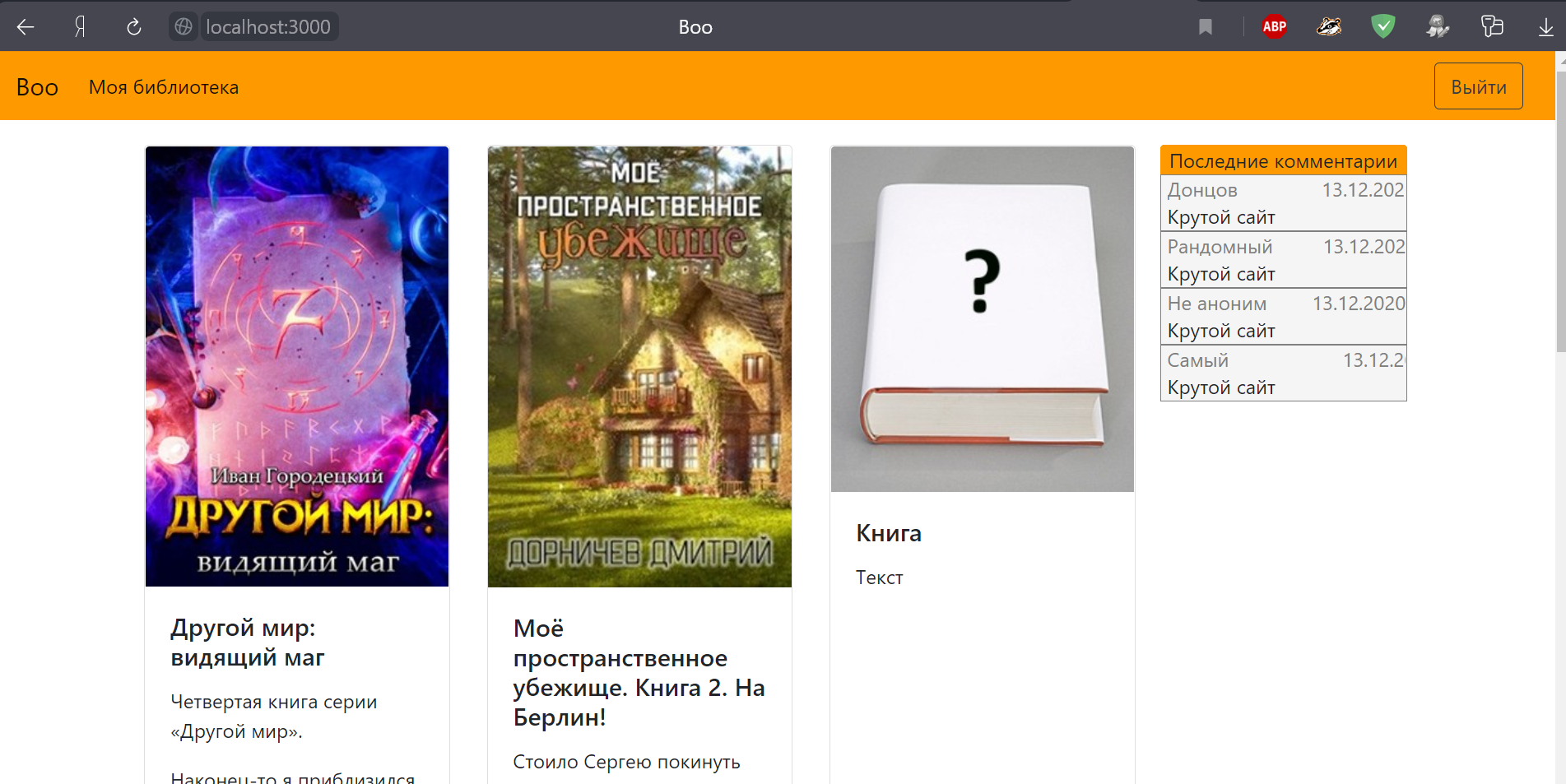
import ***http*** from '../../http'  
  
const ***AuthApi*** = {  
 checkToken: async (token: string) => {  
 await ***http***.get('checkToken', {  
 headers: {  
 'Authorization' : `Bearer ${token}`,  
 }  
 })  
 }  
}  
  
export default ***AuthApi***;

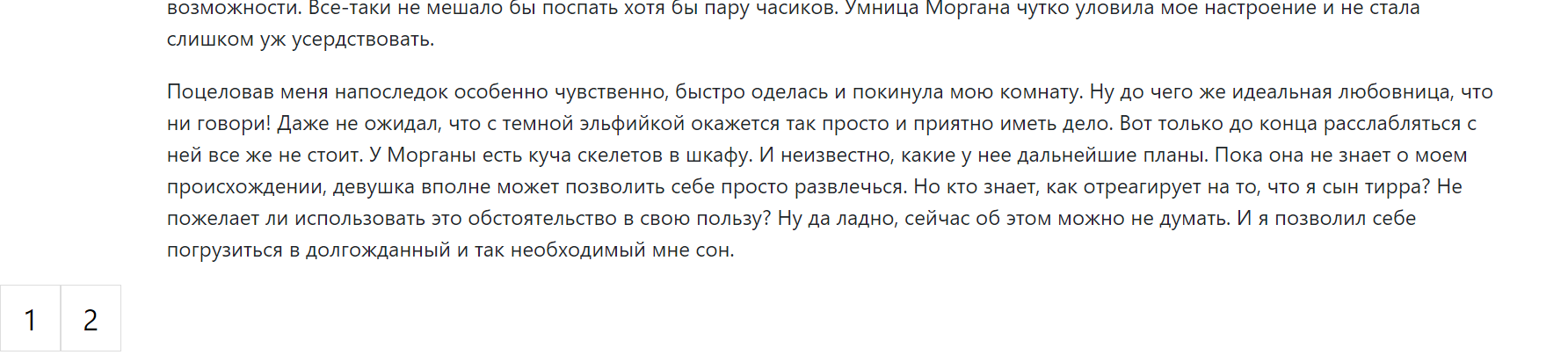
**Запрос книг**import ***http*** from '../../http'  
import {IBooks} from "../../services/book-service/type";  
  
const ***BookApi*** = {  
 getAllBook: async (token: String) => {  
 const books = await ***http***.get<IBooks[]>('/book/all', {  
 headers: {  
 'Authorization' : `Bearer ${token}`,  
 }  
 })  
 return books.data  
 }  
}  
  
export default ***BookApi***;

**Запрос главы**import ***http*** from "../../http";  
import {Read} from "./type";  
import {IRead} from "../../services/read-service/type";  
  
const ***ReadApi*** = {  
 getBookChapter: async (data: Read) => {  
 const books = await ***http***.get<IRead>(`/book/chapter/${data.bookId}/${data.chapter}`, {  
 headers: {  
 'Authorization' : `Bearer ${data.token}`,  
 }  
 })  
 return books.data  
 }  
}  
  
export default ***ReadApi***;

**Сервис для авторизации**import {createDomain, sample} from 'effector'  
import {***createReEffect***} from 'effector-reeffect'  
import ***AuthApi*** from "../../api/auth-api/auth-api";  
import ***history*** from "../../history";  
  
const checkTokenFx = ***createReEffect***({handler: ***AuthApi***.checkToken})  
  
const authDomain = createDomain('auth domain');  
  
const checkTokenEvent = authDomain.createEvent()  
  
checkTokenEvent.watch(() => checkTokenFx(***localStorage***.getItem('token') ?? ''))  
  
checkTokenFx.fail.watch(() => {  
 ***history***.push('/')  
})  
  
export const ***authService*** = {  
 checkTokenEvent  
}







Вывод: научился писать запросы на JS, понял принципы взаимодействия фронт-бэк.