МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа №6**

дисциплина «Технологии web-программирования»

тема: «Работа с HTTP-запросами»

Выполнил: студент группы ВТ-41

Ковалёв И. Д.

Проверил:

Картамышев С. В.

Белгород 2020

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

**Задание к лабораторной работе:**

1. Изучить возможности Javascript 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**

Для обращения к API будем использовать axios.

Был создан класс ApiService AuthService для работы с API.

Листинг ApiService.js:

import axios from "axios"

export default class ApiService {

    \_apiBase = 'http://localhost:8000/api'

    //\_apiBase = 'http://localhost:8080'

    async getResource(url) {

        try{

            const response = await axios({

                method: 'get',

                url: `${this.\_apiBase}${url}`,

            });

            return response.data;

        }

        catch (e){

            return null

        }

    }

    async getAllAlbums() {

        const res = await this.getResource('/albums/')

        return res

    }

    async getBestAlbums() {

        const res = await this.getResource('/top\_albums/')

        return res

    }

    async getAlbum(id) {

        const res = await this.getResource(`/albums/${id}/`)

        return res

    }

    async getAllbands() {

        const res = await this.getResource('/bands/')

        return res

    }

    async getBand(id) {

        const res = await this.getResource(`/bands/${id}/`)

        return res

    }

    async searchBand(searchRequest) {

        const res = await this.getResource(`/bands/?search=${searchRequest}`)

        return res

    }

    async searchAlbum(searchRequest) {

        const res = await this.getResource(`/albums/?search=${searchRequest}`)

        return res

    }

    async getPreferences(){

        const accessToken = localStorage.getItem('access')

        try{

            const response = await axios({

                method: 'get',

                url: `${this.\_apiBase}/user\_preferences/`,

                headers: {

                    "Authorization" : `Bearer ${accessToken}`

                }

            });

            return response.data;

        }

        catch (e){

            return null

        }

    }

}

Методы этого класса используются большинством компонентов, например, AlbumCard:

import { Component } from 'react'

import { Card, Col, Button } from 'react-bootstrap'

import ApiService from '../services/ApiService';

export default class AlbumCard extends Component {

    apiservice = new ApiService();

    state = {

        title: null,

        cover: null,

        rating: null,

        release\_date: null,

        band: null,

        genres\_id: null,

        genres: null,

        band\_id: null

    }

    componentDidMount() {

        this.apiservice

            .getAlbum(this.props.id)

            .then((data) => {

                this.setState({

                    title: data.title,

                    cover: data.cover,

                    release\_date: data.release\_date,

                    rating: data.rating,

                    band: data.band.name,

                    band\_id: data.band.id,

                    genres: data.genres,

                })

            })

    }

    renderItems(items) {

        return items.map((item) => {

            return (

                <span key={item.id}>{item.name} </span>

            )

        })

    }

    renderButton() {

        if (!this.props.isInBand) {

            return (

            <Button variant="link" type="submit" block href={"/albums/" + this.props.id}>

                Посмотреть поближе

            </Button>

            )}

    }

    render() {

        const { title, cover, rating, release\_date, band, band\_id, genres } = this.state

        if (!genres) {

            return <div></div>

        }

        const genreItems = this.renderItems(genres)

        return (

            <Col className="mt-4 align-items-stretch" xs={12} md={4} lg={3} xl={3} >

                <Card>

                    <Card.Img variant="top" src={cover} />

                    <Card.Body>

                        <Card.Title>

                            {title}

                        </Card.Title>

                        <Card.Text>

                            Исполнитель: <a href = {"bands/" + band\_id}>{" " + band}</a>

                        </Card.Text>

                        <Card.Text>

                            Дата выхода: {release\_date}

                        </Card.Text>

                        <Card.Text>

                            Рейтинг: {rating}

                        </Card.Text>

                        <Card.Text>

                            Жанры: {genreItems}

                        </Card.Text>

                    </Card.Body>

                    {this.renderButton()}

                </Card>

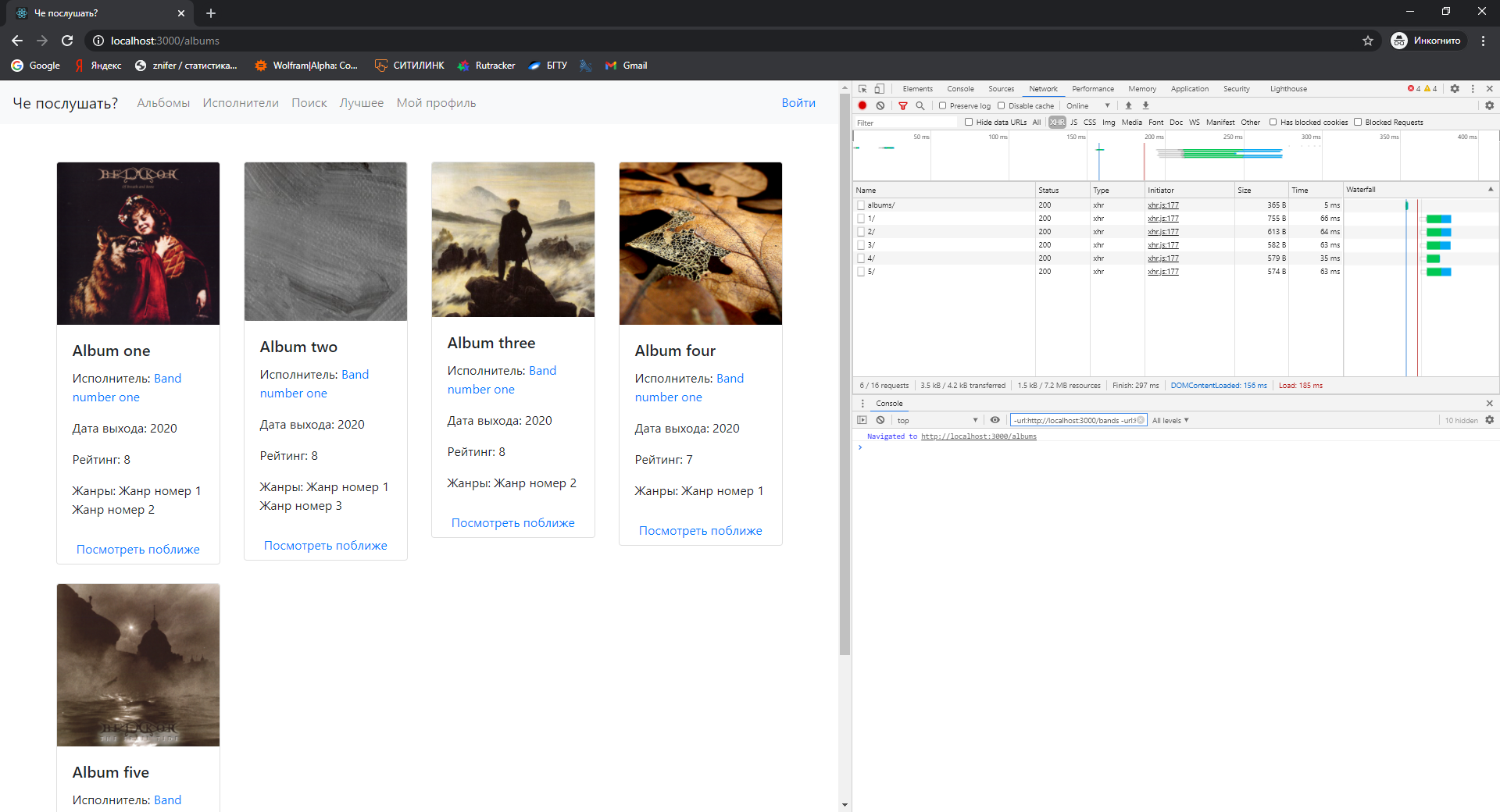
            </Col>

        )

    }

}

Проверим данный компонент на работоспособность:



Как видно из скриншота, данные подгружаются корректно.