

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Лабораторная работа по теме  
«Работа с HTTP запросами»  
дисциплина:  
Технологии web-программирования

Выполнил: ст. группы ВТ-41  
Фаракшин Н. Р.  
Проверил: ст. пр. Картамышев С.В.

## **Лабораторная работа № 6**

### **Работа с HTTP запросами**

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

#### **Задание к лабораторной работе:**

1. Изучить возможности React 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.

#### **Выполнение работы**

Для работы с запросами к REST API была выбрана библиотека axios. Для комфортного использования библиотеки написан класс ApiService, содержащий функции вызова конкретных конечных точек REST API. Авторизация выполняется с помощью JWT токенов. Полученный при аутентификации токен сохраняется в localStorage и используется для авторизации в дальнейшем, до истечения срока действия токена. Процесс получения токена и проверки статуса пользователя перенесен в класс AuthService, таким образом, работа с API аутентификации централизована.

## Демонстрация работы приложения

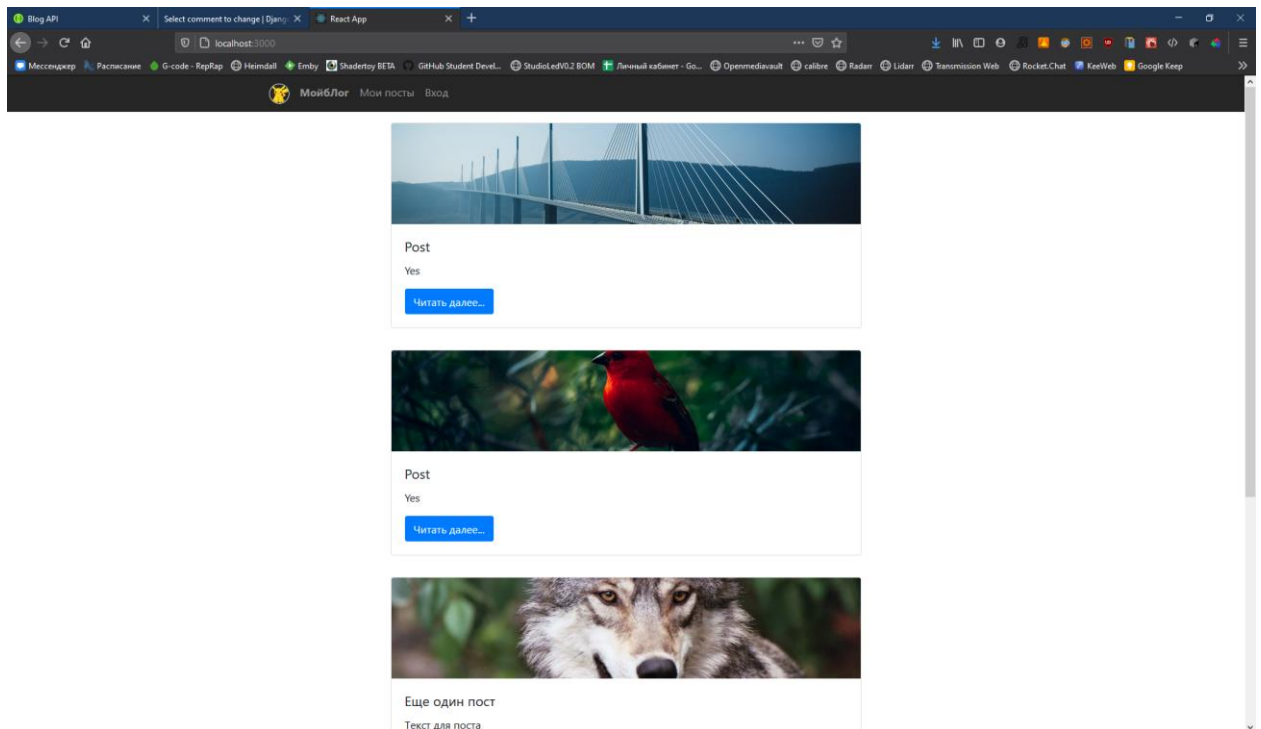


Рис. 1 Список постов всех пользователей

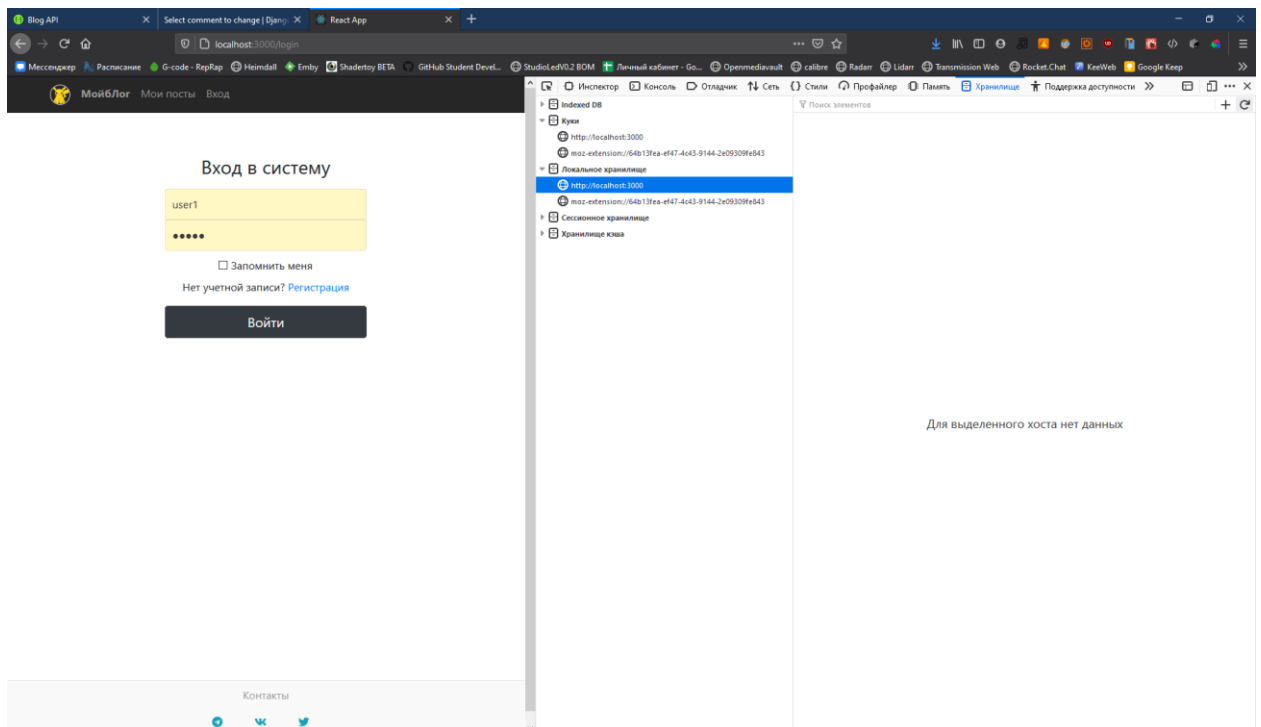


Рис. 2 Для просмотра страницы со своими постами необходимо войти

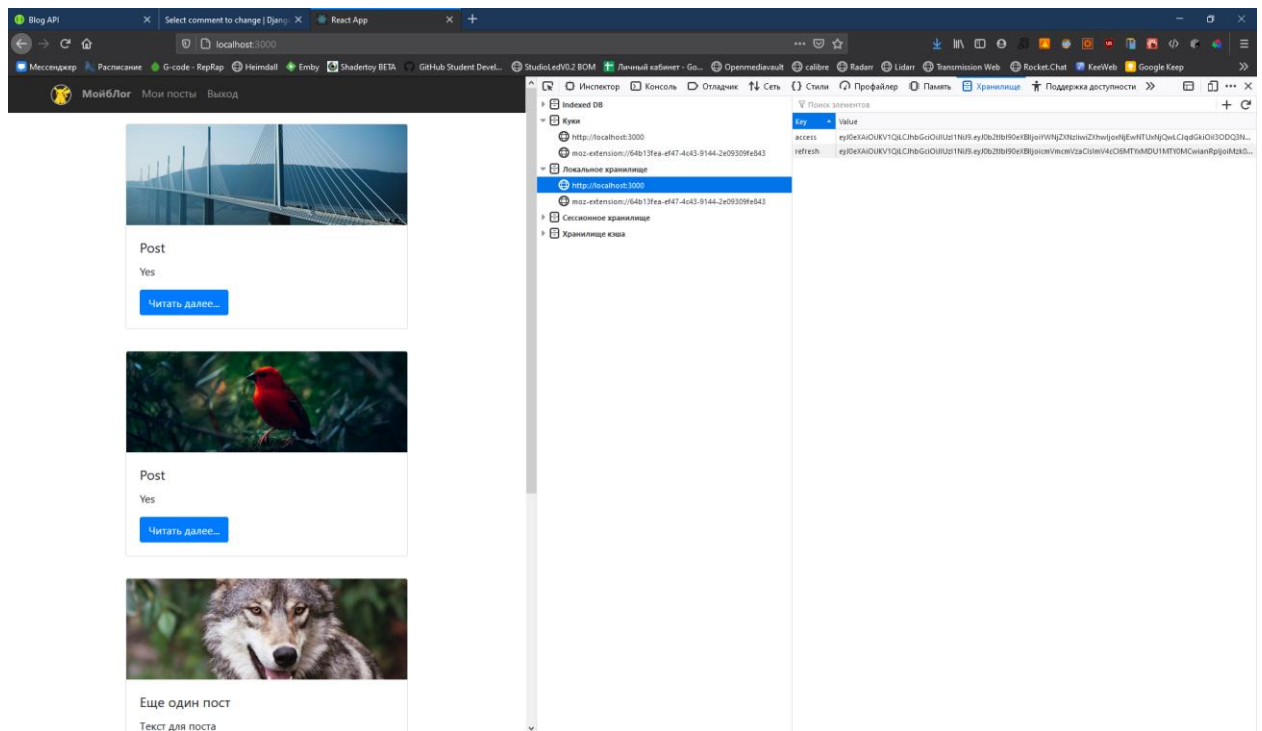


Рис. 3 Вход выполнен. Токен сохранен в localStorage

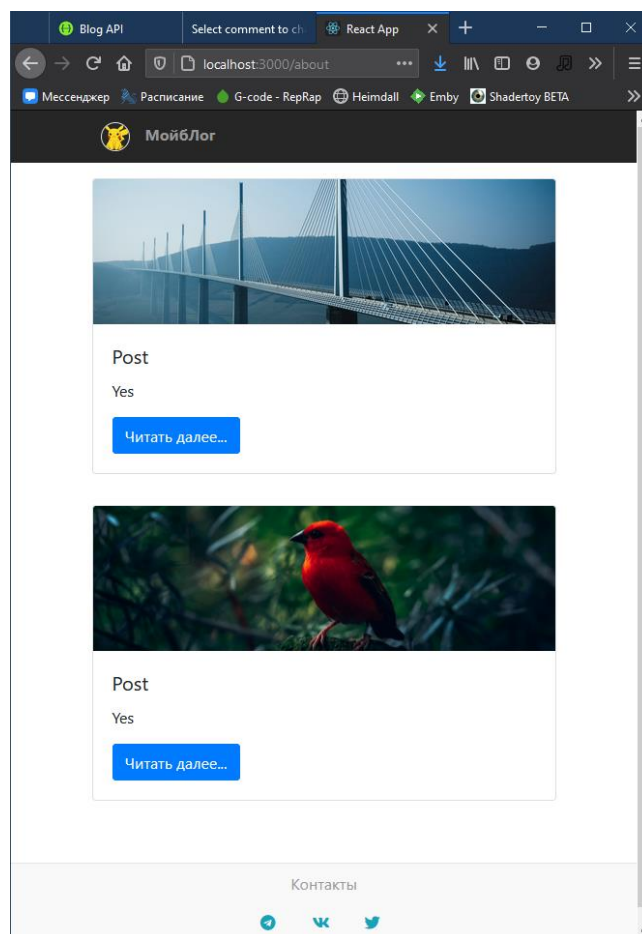
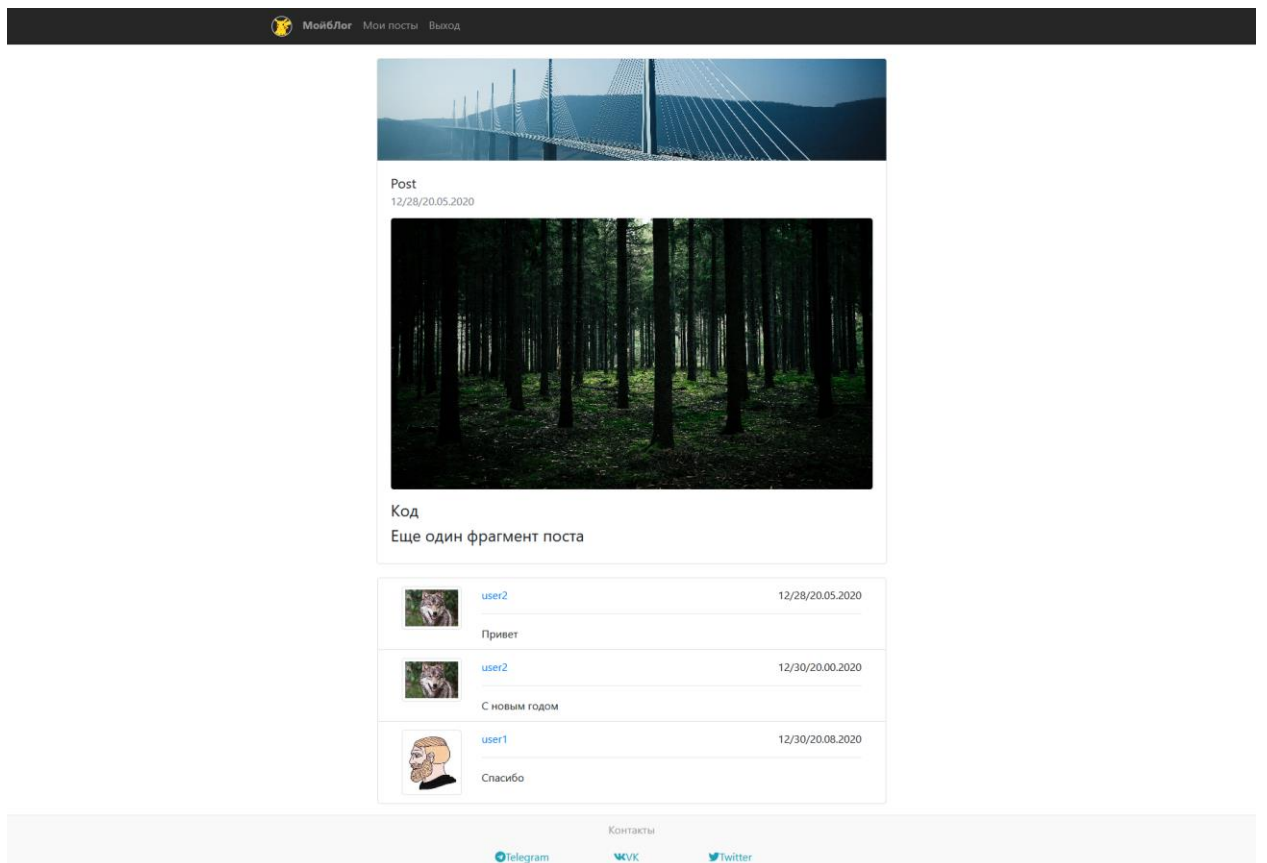


Рис. 4 При переходе на страницу «Мои посты» пользователь видит только свои посты



*Рис. 5 Вид одного поста. Под постом расположен блок комментариев*

## Приложение

Исходный код компонентов Comment и CommentList, служащих для отображения одного комментария и блока комментариев соответственно. Исходный код вспомогательных классов AuthService и ApiService.

### Comment.js

```
import React from 'react'
import './Comment.css'
import ApiService from "../Services/ApiService";
import {Link} from "react-router-dom";

class Comment extends React.Component {

  apiservice = new ApiService();

  state = {
    user: null,
    creation_time: null,
    content: null,
  }

  componentDidMount() {
    this.apiservice
      .getUser(this.props.user)
      .then((data) => {
        console.log(data);
        this.setState({
          user: data,
          content: this.props.content,
          creation_time: this.props.creation_time,
        });
      })
  }

  render() {
    return (
      <div className="container">
        <div className="row">
          <div className="col-2">
            <img className="img-thumbnail" src={this.state.user ? this.state.user.profile_pic: ""} alt="Не загружено"/>
          </div>
          <div className="col-10">
            <div className="text-inline justify-content-between d-flex">
              <Link to={"/user/" + this.props.user} className="">{this.state.user ? this.state.user.username : ""}</Link>
            </div>
          </div>
        </div>
      </div>
    )
  }
}
```

```

        <div className="comments-
date">{this.state.creation_time}</div>
        </div>
        <hr/>
        <div className="text-inline">
            {this.state.content}
        </div>
    </div>
</div>
</div>
)
}
}

```

```
export default Comment;
```

### CommentList.js

```

import React from 'react';
import './CommentList.css';
import Comment from "../Comment/Comment";
import ApiService from "../Services/ApiService";

class CommentList extends React.Component {
    apiservice = new ApiService()
    state = {
        comments: null
    }
    componentDidMount() {
        this.apiservice.getComments(this.props.post_id)
            .then((data) => {
                this.setState({
                    comments: data
                })
            })
    }
    render() {
        const listComments = this.state.comments ? this.state.comments.map((comment) =>
            <li className="list-group-item d-flex">
                <Comment user={comment.user} content={comment.content} creation_time={comment.creation_time}/>
            </li>
        ) : [];
        return (
            <div className="container px-0 mb-3">
                <div className="row justify-content-center">
                    <div className="col-12 col-md-8">
                        <ul className="list-group w-100">
                            {listComments}
                        </ul>
                    </div>
                </div>
            </div>
        )
    }
}

```

```

        </div>
      </div>
    </div>
  );
}
}

```

```
export default CommentList;
```

### APIService.js

```

import axios from "axios"

export default class ApiService {
  _apiBase = 'http://localhost:8000/api'
  async getResource(url) {
    try{
      const response = await axios({
        method: 'get',
        url: `${this._apiBase}${url}`,

      });
      return response.data;
    }
    catch (e){
      return null
    }
  }
  async getAllPosts() {
    return await this.getResource('/posts/')
  }
  async getPosts(user_id) {
    return await this.getResource(`/posts/?user=${user_id}`)
  }
  async getMyPosts() {
    const accessToken = localStorage.getItem('access')
    console.log(accessToken)
    try{
      const response = await axios({
        method: 'get',
        url: `${this._apiBase}/my_posts/`,
        headers: {
          Authorization : `Bearer ${accessToken}`
        }
      });
      return response.data;
    }
    catch (e){
      console.log("Не получилось")
      return null
    }
  }
}

```



```

    async getComments(post_id) {
        return await this.getResource(`/comments/?post=${post_id}`)
    }
    async getPost(id) {
        return await this.getResource(`/posts/${id}/`)
    }
    async getUser(id) {
        return await this.getResource(`/users/${id}/`)
    }
    async getContent(id) {
        return await this.getResource(`/contents/?parent=${id}`)
    }
}

```

### AuthService.js

```

import axios from "axios"
export default class AuthService {
    async login(username, password) {
        try {
            const response = await axios({
                method: 'post',
                url: "http://localhost:8000/api/token/",
                data: {
                    username: username,
                    password: password
                }
            })
            const data = await response.data
            window.localStorage.setItem('access', data.access)
            window.localStorage.setItem('refresh', data.refresh)
        }
        catch (e) {
            console.log(e);
        }
    }
    logout() {
        console.log('deleting')
        localStorage.removeItem("access")
        localStorage.removeItem("refresh")
    }
    isLoggedIn() {
        if (localStorage.getItem("access")) {
            return true
        }
        return false
    }
}

```