

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 2

Выполнил:

Власов Владислав

**Группа
К3439**

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2026 г.

Задача

Необходимо привязать то, что было сделано в ЛР1 к внешнему API средствами fetch/axios/xhr. Реализуйте моковое API средствами JSON-сервера и подключите к нему авторизацию

Ход работы

Для лабораторной работы были реализованы методы авторизации, получения данных о пользователе, создание пользователя, получения списка навыков, создание нового навыка.

Данный функционал был сделан на базе кода из лабораторной работы 1. http запросы были реализованы с помощью axios, а данные сохраняются в localstore.

Вот геттеры и сеттеры для хранилищ:

```
const getToken = () => localStorage.getItem('token');
const setToken = (token) => localStorage.setItem('token', token);
const removeToken = () => localStorage.removeItem('token');

const getUser = () => {
  const userStr = localStorage.getItem('user');
  return userStr ? JSON.parse(userStr) : null;
};

const setUser = (user) => localStorage.setItem('user', JSON.stringify(user));
const removeUser = () => localStorage.removeItem('user');

const getUserId = () => localStorage.getItem('userId');
const setId = (userId) => localStorage.setItem('userId', userId);
const removeUserId = () => localStorage.removeItem('userId');

const getSkillsArray = () => {
  try {
    const skillsJson = localStorage.getItem('skills');
    if (!skillsJson) return {};

    const skills = JSON.parse(skillsJson);
    return skills;
  } catch (error) {
    console.error('Ошибка при чтении навыков из localStorage:', error);
    return {};
  }
};

const setSkillsArray = (skillsArray) => {
  try {
```

```

        const formattedSkills = {}
        skillsArray.forEach(skill => {
            formattedSkills[skill.name] = skill
        })

        localStorage.setItem('skills', JSON.stringify(formattedSkills));
        return true;
    } catch (error) {
        console.error('Ошибка при сохранении навыков в localStorage:', error);
        return false;
    }
};

const removeSkillsArray = () => localStorage.removeItem('skills');

```

А вот методы взаимодействия с внешним API:

```

async function getAllSkillApi() {
    return await axios.get(`http://localhost:3003/characters/api/skills/`, {
headers: {'Authorization': `Bearer ${getToken()}`} })
}

async function createSkillApi(data) {
    return await
axios.post(`http://localhost:3003/characters/api/skills/create`, data, {
headers: {'Authorization': `Bearer ${getToken()}`} })
}

```

И пример их интеграции:

```

async function saveSkill(event)
{
    event.preventDefault()

    const skillData = {
        'name': event.target.querySelector('input[name="name"]').value,
        'attribute':
event.target.querySelector('select[name="attribute"]').value
    }

    if (!skillData['name'] || !skillData['attribute']) {
        return
    }

    console.log(skillData)

    await createSkillApi(skillData)
};

```

```
<form id="createForm" onsubmit="saveSkill(event)">
    <div class="mb-3">
```

Вывод

В ходе работы сайт был дополнен взаимодействием с внешним арі и теперь на нём могут отображаться не статичные данные-заглушки, а актуальные данные из БД.