

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Акулов Даниил

К3439

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

Задача

Варианты остаются прежними. Теперь Вам нужно привязать то, что Вы делали в ЛР1 к внешнему API средствами fetch/axios/xhr. Реализуйте моковое API средствами JSON-сервера и подключите к нему авторизацию, как в примерах, которые мы рассматривали в рамках тем "Имитация работы с API".

Например, для приложения для просмотра прогнозов погоды задание выглядит следующим образом:

Реализовать получение погоды (прогноз на ближайшие 7 дней) из открытого API OpenWeatherMap, в зависимости от геолокации пользователя. Реализовать вывод полученного прогноза в виде 7 карточек в три ряда (первый ряд - крупная карточка, второй ряд - три карточки в меньшем размере, третий ряд - четыре карточки в маленьком размере).

Ход работы

Что реализовано:

- Добавлен файл [api.js](#) с функциями для обращения к api

```
const API_URL = 'http://localhost:8080/api';

async function fetchPost(url, data, headers = {}) {
    const response = await fetch(url, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            ...headers
        },
        body: JSON.stringify(data)
    })
    return await response.json()
}

const api = {
    async login(email, password) {
        try {
            const data = await fetchPost(` ${API_URL}/auth/login`, {email,
password});
        }
    }
}
```

```

        if (data.error) {
            return { success: false, message: data.error?.message || 'Fetch failed' };
        }

        const {token, user} = data;
        return { success: true, token, user };
    } catch (error) {
        console.error('Login error:', error);
        return { success: false, message: 'Server Error' };
    }
},
}

```

- Главная страница (index.html)

```

<script>

document.addEventListener('DOMContentLoaded', async () => {
    const workoutGrid = document.getElementById('workoutGrid');
    const workouts = await api.getWorkoutPlans();

    workouts.forEach(workout => {
        const col = document.createElement('div');
        col.className = 'col';
        col.innerHTML = `

            <div class="card h-100 workout-card"
data-level="${workout.level}" data-type="${workout.type}">

                

                    <div class="card-body">
                        <h5 class="card-title">${workout.title}</h5>
                        <p class="card-text">${workout.description}</p>
                    <div class="d-flex justify-content-between
align-items-center">

                        <span class="badge
${getBadgeClass(workout.level)}">${getBadgeText(workout.level)}</span>

                    </div>
                </div>

            <div class="card-footer bg-white border-top-0">
                <a href="workout.html?id=${workout.id}" class="btn btn-outline-primary w-100">Перейти</a>
            </div>
        
```

```

        </div>
    </div>
    ;
    workoutGrid.appendChild(col);
}) ;

const event = new Event("WorkoutCardsLoaded")
document.dispatchEvent(event)
);

function getBadgeClass(level) {
    if (level === 1) return 'bg-success';
    if (level === 2) return 'bg-warning text-dark';
    return 'bg-danger';
}

function getBadgeText(level) {
    if (level === 1) return 'Новичок';
    if (level === 2) return 'Средний';
    return 'Продвинутый';
}

```

- Авторизация

```

<script>

    document.getElementById('loginForm').addEventListener('submit', async
function(e) {
    e.preventDefault();
    const email = document.getElementById('email').value;
    const password = document.getElementById('password').value;

    const result = await api.login(email, password);

    if (result.success) {
        localStorage.setItem('currentUser', JSON.stringify(result.user));
        localStorage.setItem('token', JSON.stringify(result.token));
        alert('Успешный вход!');
        window.location.href = 'dashboard.html';
    }
}

```

```
    } else {
        alert(result.message);
    }
}) ;
</script>
```

- Подгрузка тренировок
- Сохранить тренировку в профиле
- Подгрузка постов блога

Вывод

В результате работы реализовано сетевое взаимодействие фронтенда с бекеном сервиса для тренировок.