

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Микросервисы**

**РАБОТУ ВЫПОЛНИЛА:**  
Штрейх Анна Андреевна

**РАБОТУ ПРОВЕРИЛ:**  
Добряков Давид Ильич

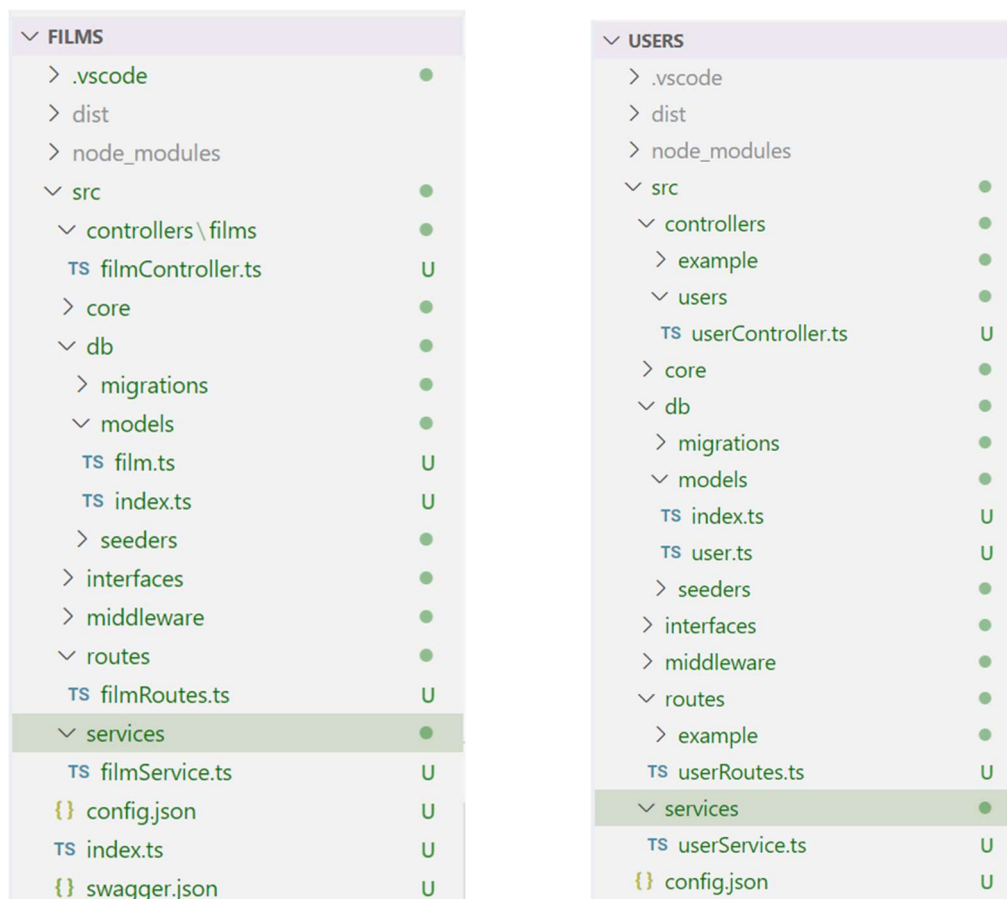
Санкт-Петербург  
2022 г.

### Цель работы:

Реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций приложения.

### Выполнение:

В отдельный микросервис был вынесен сервис взаимодействия с базой данных фильмов – их добавление, изменение, фильтрация и поиск.



Рисунки 1 и 2 –сервисы users и films

Ранее в приложении использовались куки, но при микросервисной архитектуре их использование усложнилось тем, что теперь мы имеем два разных порта, и необходимо настраивать передачу заголовков через nginx. Для упрощения работы решено было перейти на токены jwt.

```

loggingIn = async (request: any, response: any) => {
  const logInData = request.body;
  try {
    const user = await this.userService.getByEmail(logInData.email)
    const isPasswordMatching = await bcrypt.compare(logInData.password, user.password);
    if (isPasswordMatching) {
      user.password = 'password';
      const tokenData = this.userService.createToken(user);
      //response.setHeader('Set-Cookie', [this.createCookie(tokenData)]);
      response.send(tokenData);
    } else {
      response.send({ "error": "Wrong password." })
    }
  }
} catch (error: any) {
  response.status(404).send({ "error": error.message })
}
}

```

Рисунок 3 – Теперь мы получаем токен при авторизации

```

async function authMiddleware(request: any, response: any) {
  try {
    const accessTokenSecret = 'giveme678G563/';
    const authHeader = request.headers.authorization;
    if (authHeader) {
      const token = authHeader.split(' ')[1]
      const verificationResponse = jwt.verify(token, accessTokenSecret) as DataStoredInToken
      const id = verificationResponse._id;
      console.log(token)
      const curUser = `http://localhost:8333/v1/users/id/${id}`
      let resp = await fetch(curUser)
      let user = await resp.json()
      request.user = user;
      return user
    }
  }
  else {
    throw new Error(`You are not authored`)
  }
}

```

Рисунок 4 – Получение текущего пользователя в сервисе фильмов

## Выводы:

Получены навыки работы с микросервисной архитектурой проекта.