

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа №3

“Тестирование, разработка и документирование RESTful  
API”

Выполнил:

Екушев Владислав

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

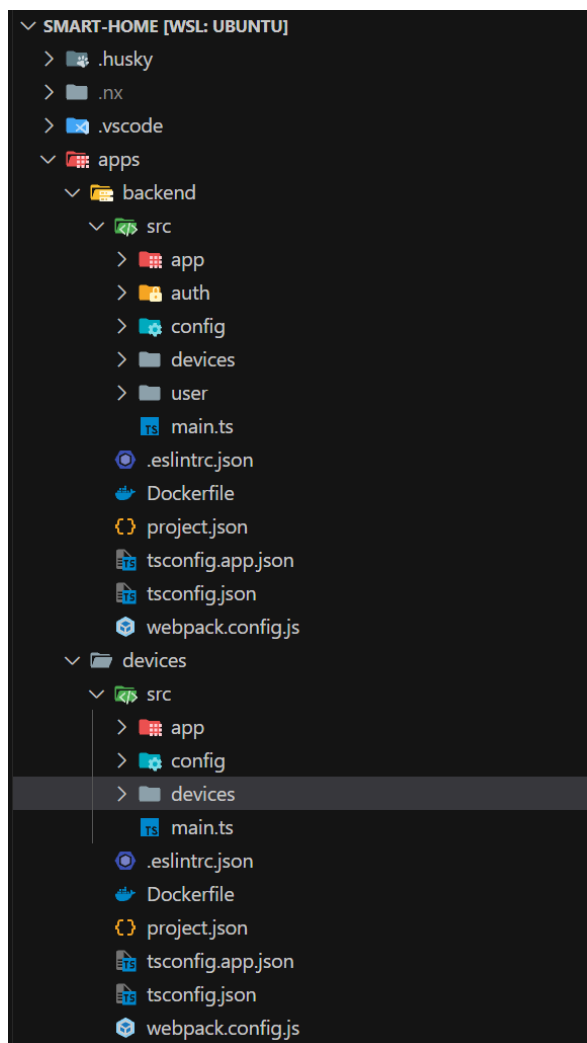
2024 г.

## Задача

Необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций вашего приложения.

## Ход работы

Будем выносить в отдельный микросервис сервис по управлению девайсами в умном доме (сервис devices)



Так как у нас была сделана структура монорепоzitория с использованием NxJS, можно легко вынести в отдельную папку новый сервис на NestJS.

Общение сервисов произведено через очередь RabbitMQ. Для того, чтобы использовать микросервис девайсов на бекенде, нужно включить его в провайдеры в модуле девайсов:

You, 6 days ago | 1 author (You)

```

11 @Module({
12   imports: [
13     MongooseModule.forFeature([{ name: Device.name, schema: DeviceSchema }]),
14     MongooseModule.forFeature([{ name: User.name, schema: UserSchema }]),
15   ],
16   providers: [
17     DevicesService,
18     ConfigService,
19     UserService,
20     {
21       provide: 'DEVICES_SERVICE',
22       useFactory: () => {
23         return ClientProxyFactory.create({
24           transport: Transport.RMQ,
25           options: {
26             urls: ['amqp://rabbitmq:5672'],
27             queue: 'devices_service_queue',
28             queueOptions: {
29               durable: false,
30             },
31           },
32         })
33       },
34     },
35   ],
36   controllers: [DevicesController],
37 })
38 export class DevicesModule {}

```

После этого можно обращаться к функциям сервиса внутри кода:

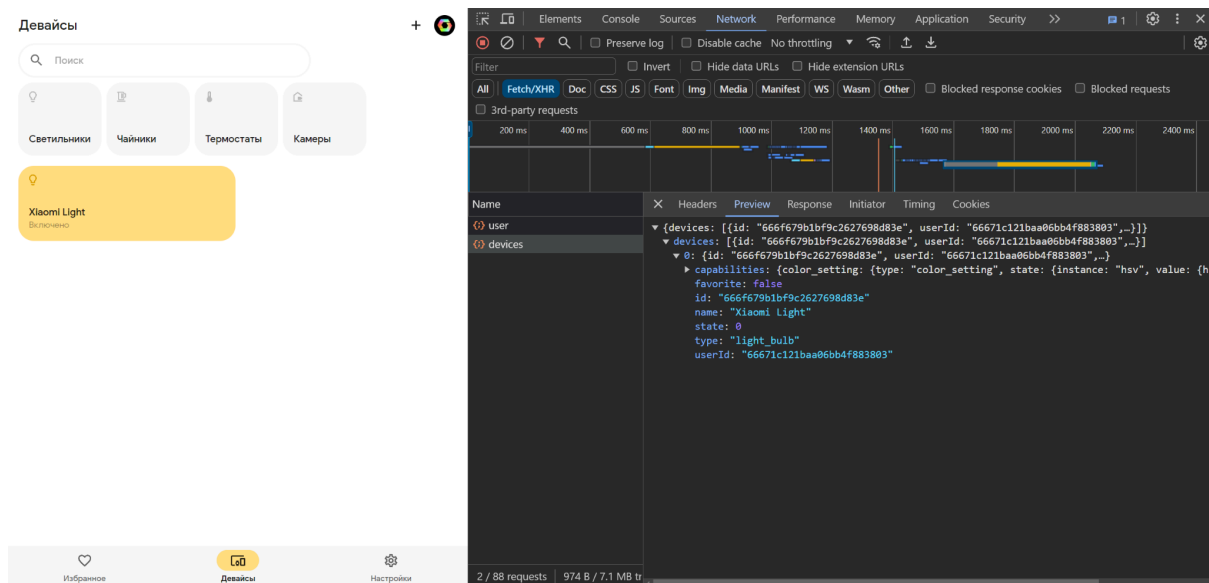
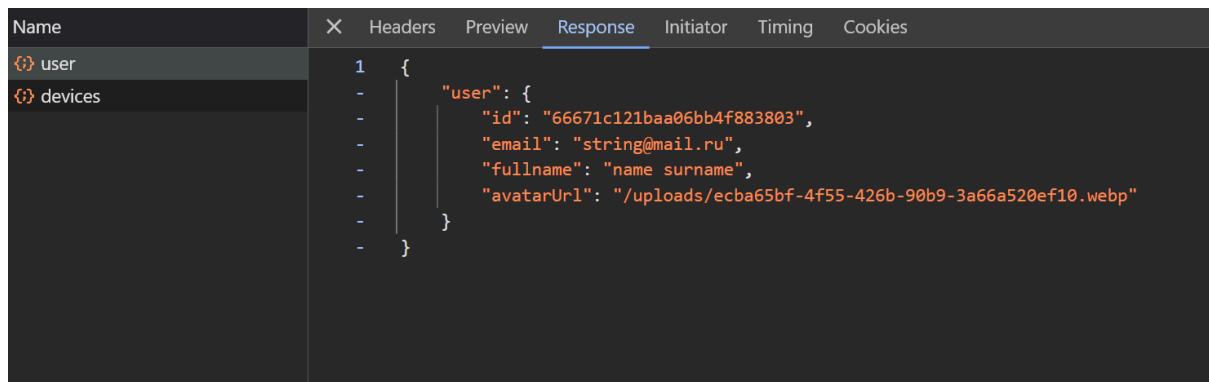
```

7  @Injectable()
8  export class DevicesService {
9    constructor(
10     @Inject('DEVICES_SERVICE') private readonly client: ClientProxy,
11     private userService: UserService
12   ) {}
13
14   async getDevices(userId: string): Promise<IDevice[]> {
15     return await firstValueFrom(
16       this.client.send({ cmd: 'getDevices' }, { userId })
17     )
18   }
19 }

```

You, 5 months ago • Add toggle on-off capability ...

Проверим работоспособность сервисов:



Видим, что клиент получает список девайсов от бекенда – следовательно, микросервис успешно общается и передает данные главному приложению.

## Вывод

В ходе лабораторной работы был вынесен в микросервис сервис для умных девайсов.

Результат представлен в репозитории –

<https://github.com/jarvis394/smart-home/tree/%40jarvis394/addMicroservice>