

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе № 3-4 «Микросервисы. Docker. Docker-compose»
по дисциплине «Бэкенд-разработка»

Автор: Власов М. И.

Факультет: ИКТ

Группа: К33402

Преподаватель: Добряков Д. И.

Дата: 29.05.22



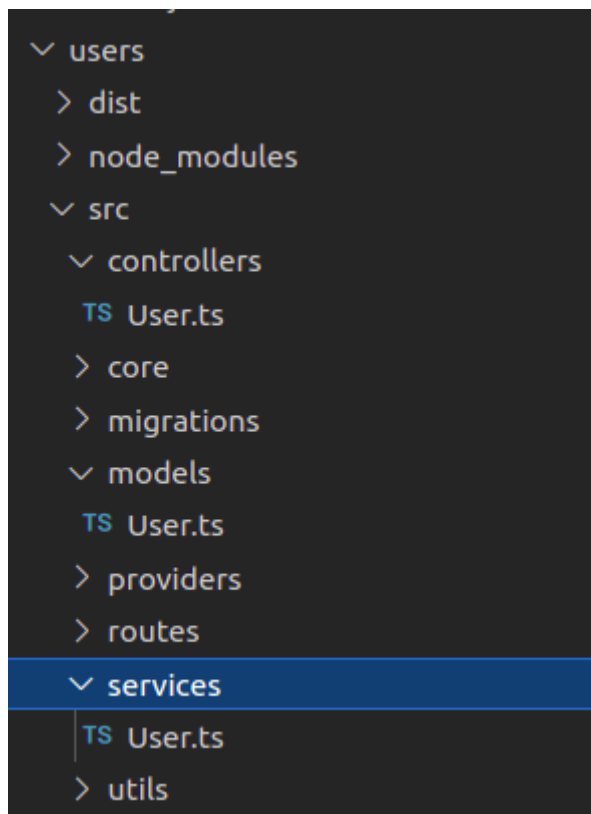
УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2022

Цель: реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций приложения, а затем упаковать приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения.

Создание микросервиса

Выделили *users* в отдельный микросервис



Во втором сервисе *hotels* остались *property* и *bookings*

Gateway

Один из вариантов взаимодействия сервисов - *express-http-proxy*

```
const express = require('express');
const proxy = require('express-http-proxy');

const app = express();

app.use('/users', proxy('http://localhost:8002'))
app.use('/', proxy('http://localhost:8001'))

app.listen(8000, () => {
  console.log(`Running server on port 8000`)
})
```

Dockerfile

```
FROM node:12
WORKDIR /app/users/
COPY package.json /app/users/package.json
RUN npm install
COPY . .
EXPOSE 8002
CMD ["npm", "start"]
```

```
matvey@matvey-Aspire-A315-55KG:~/ITM0-ICT-Backend-2022/labs/K33402/Vlasov_Matvey/LW3_4/users$ sudo docker build . -t users
```

```
matvey@matvey-Aspire-A315-55KG:~/ITM0-ICT-Backend-2022/labs/K33402/Vlasov_Matvey/LW3_4/users$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
users latest f3f1de486f5a 59 seconds ago 1.11GB
```

```
matvey@matvey-Aspire-A315-55KG:~/ITM0-ICT-Backend-2022/labs/K33402/Vlasov_Matvey/LW3_4/hotels$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
330be779d1a2 hotels "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:49162->8001/tcp, :::49162->8001/tcp
917f5ddb77f3 users "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:49161->8002/tcp, :::49161->8002/tcp
```

Docker-compose

```
version: '3'
services:
  db:
    image: postgres:13
    volumes:
      - /var/lib/postgresql/13/main
    env_file:
      - .env
    expose:
      - 5432

  users:
    build:
      dockerfile: Dockerfile
      context: ./users
    container_name: users
    ports:
      - "8002:8002"
    depends_on:
      - "db"
    volumes:
      - ./app

  hotels:
    build:
      dockerfile: Dockerfile
      context: ./hotels
    container_name: hotels
    ports:
      - "8001:8001"
    depends_on:
      - "db"
    volumes:
      - ./app
```

```
db_1      | 2022-05-29 19:35:24.843 UTC [1] LOG:  database system is ready to accept connections
users     | > lw1@1.0.0 start /app/users
users     | > nodemon dist/index.js
hotels    | > lw1@1.0.0 start /app/hotels
hotels    | > nodemon dist/index.js
```

```
users      Running server on port 8002
users      [
users      {
users      path: '/create',
users      methods: [ 'POST' ],
users      middlewares: [ 'anonymous' ]
users      },
users      {
users      path: '/get/:id',
users      methods: [ 'GET' ],
users      middlewares: [ 'anonymous' ]
users      },
users      {
users      path: '/update/:id',
users      methods: [ 'POST' ],
users      middlewares: [ 'anonymous' ]
users      },
users      {
users      path: '/delete/:id',
users      methods: [ 'POST' ],
users      middlewares: [ 'anonymous' ]
users      },
users      { path: '/list', methods: [ 'GET' ], middlewares: [ 'anonymous' ] },
users      { path: '/login', methods: [ 'POST' ], middlewares: [ 'anonymous' ] }
users      ]
```

```
hotels Running server on port 8001
hotels [
hotels {
hotels   path: '/property/create',
hotels   methods: [ 'POST' ],
hotels   middlewares: [ 'anonymous' ]
hotels },
hotels {
hotels   path: '/property/get/:id',
hotels   methods: [ 'GET' ],
hotels   middlewares: [ 'anonymous' ]
hotels },
hotels {
hotels   path: '/property/update/:id',
hotels   methods: [ 'POST' ],
hotels   middlewares: [ 'anonymous' ]
hotels },
hotels {
hotels   path: '/property/delete/:id',
hotels   methods: [ 'POST' ],
hotels   middlewares: [ 'anonymous' ]
hotels },
hotels {
hotels   path: '/property/list',
hotels   methods: [ 'GET' ],
hotels   middlewares: [ 'anonymous' ]
hotels },
hotels {
hotels   path: '/bookings/create',
hotels   methods: [ 'POST' ],
```

Вывод: в ходе лабораторной работы мы реализовали отдельные микросервис *users* и *hotels*, а затем упаковали их в docker-контейнеры и обеспечили сетевое взаимодействие между ними с помощью docker-compose.