

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

**Лабораторная работа №3:
Микросервисы**

Выполнила:

Балдина Дарья

Группа

К33401

Проверил:

Добряков Д. И.

Санкт-Петербург

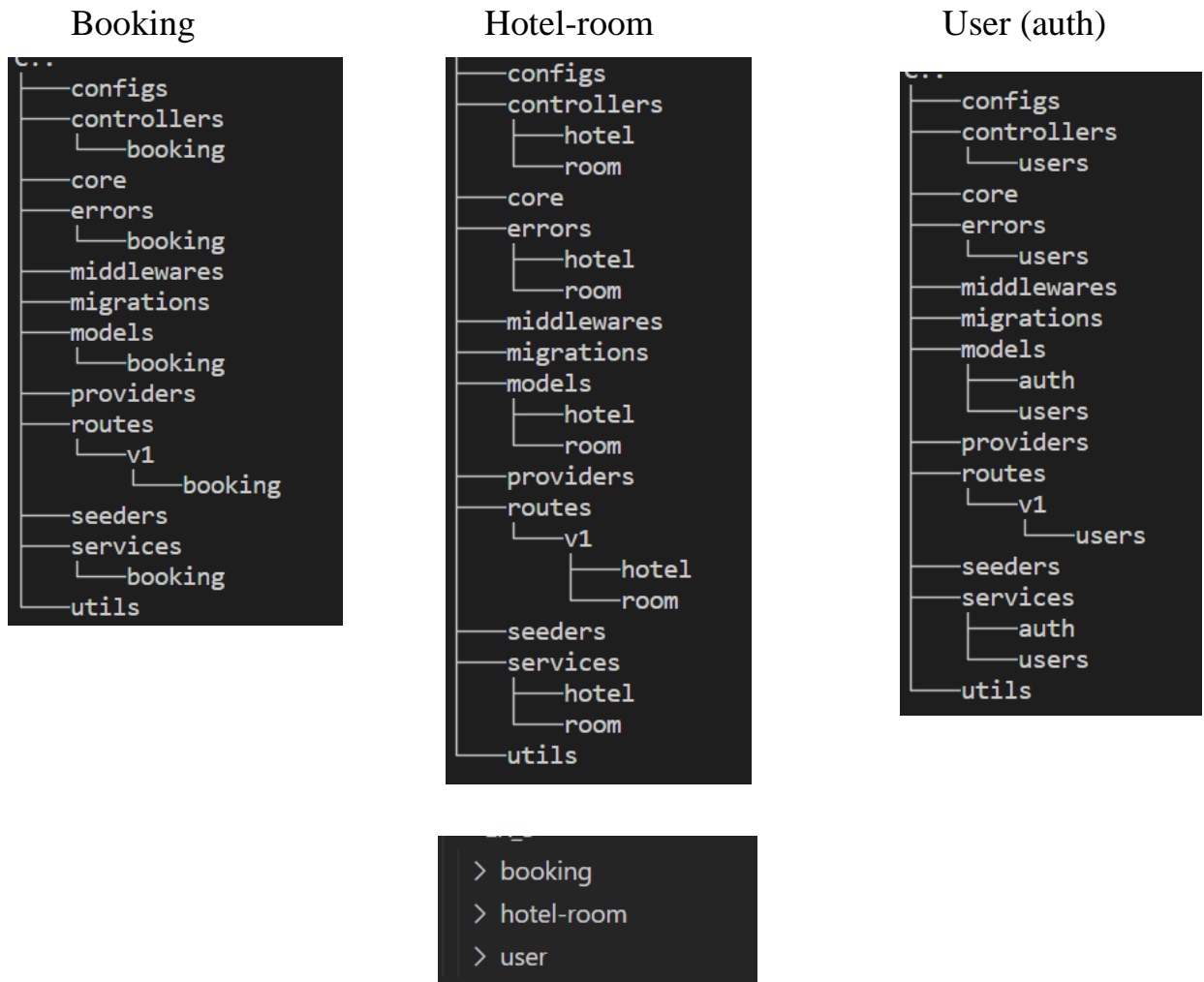
2023 г.

Задача

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

Ход работы

1. Структура



Я решила разделить монолитное приложение на 3 микросервиса: авторизация, сервис бронирования номеров отеля, сервис по работе с отелями и комнатами в нем. Как видно структура у микросервисов схожа между собой.

Эти 3 микросервиса находятся на 3 разных портах - <http://localhost:9000> – авторизация, <http://localhost:9001> – сервис с отелями и комнатами, и <http://localhost:9002> для сервиса бронирования.

2. Авторизация

Для проверки авторизации в модель пользователя был добавлен новый ЭНДПОИНТ:

```
validateToken = async (request: any, response: any) => {  
  const {body} = request  
  const {accessToken} = body  
  try {  
    const payload = jwt.verify(accessToken, jwtOptions.secretOrKey)  
    // @ts-ignore  
    const user = await this.userService.getById(payload.id)  
    response.send({'valid': true, 'user': user})  
  } catch (e: any) {  
    response.status(401).send({'valid': false})  
  }  
}
```

3. Middleware

Middleware в микросервисах бронирования и работы с отелями/комнатами был переписан: теперь для проверки авторизации мы делаем запрос к микросервису авторизации

```
const customJwtStrategy = new Strategy(async function (token: any, done: any) {  
  axios.post(  
    `http://localhost:9090/v1/users/validate`,  
    {'accessToken': token}  
  ).then((resp) => {  
    if (resp.status == 200 && resp.data.valid) {  
      const user = resp.data.user  
      done(null, user)  
    } else {  
      done(null, false)  
    }  
  })  
}).catch((error) => {  
  done(error)  
})  
}))  
  
passport.use(customJwtStrategy)  
  
export default passport
```

4. Бронирование

Для получения информации по комнате в микросервисе бронирования был написан вспомогательный запрос:

```
const findRoom = async (roomNumber:any) => {
  try {
    const response = await axios.get(`http://localhost:9091/v1/room/number/${roomNumber}`);

    const room = response.data;
    return room;
  } catch (error) {
    // Обработка ошибки
    console.error(error);
    return null;
  }
};

const room = await findRoom(roomId)
```

5. Тестирование

Микросервис авторизации:

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:9090/v1/users/reg
- Body Type:** JSON
- Request Body:**

```
1 {
2   "username": "mash",
3   "firstName": "Мария",
4   "lastName": "Прокопьева",
5   "email": "test1@mail.ru",
6   "password": "test1"
7 }
```
- Status Code:** 201 Created
- Response Body (Pretty):**

```
1 {
2   "id": 1,
3   "username": "mash",
4   "firstName": "Мария",
5   "lastName": "Прокопьева",
6   "email": "test1@mail.ru",
7   "password": "$2b$08$/puAcgfuj5KcnzGh7Rao70XrLC5J.w0UgS0pj70mv4FGS0qg05C8y",
8   "updatedAt": "2023-05-22T14:23:40.063Z",
9   "createdAt": "2023-05-22T14:23:40.063Z"
10 }
```

Микросервис с отелями/комнатами:

POST http://localhost:9091/v1/hotel/ Try ↗

Params Headers **Body**

raw JSON Beautify

```
1 {
2   ... "name": "Машина",
3   ... "address": "Бассейная,45",
4   ... "stars": "5"
5 }
```

Body Headers (8) Status code 201 Created

Pretty Raw Preview JSON

```
1 {
2   "id": 3,
3   "name": "Машина",
4   "address": "Бассейная,45",
5   "stars": "5",
6   "updatedAt": "2023-05-22T16:01:51.275Z",
7   "createdAt": "2023-05-22T16:01:51.275Z"
8 }
```

POST http://localhost:9091/v1/room/ Try ↗

Params Headers **Body**

raw JSON Beautify

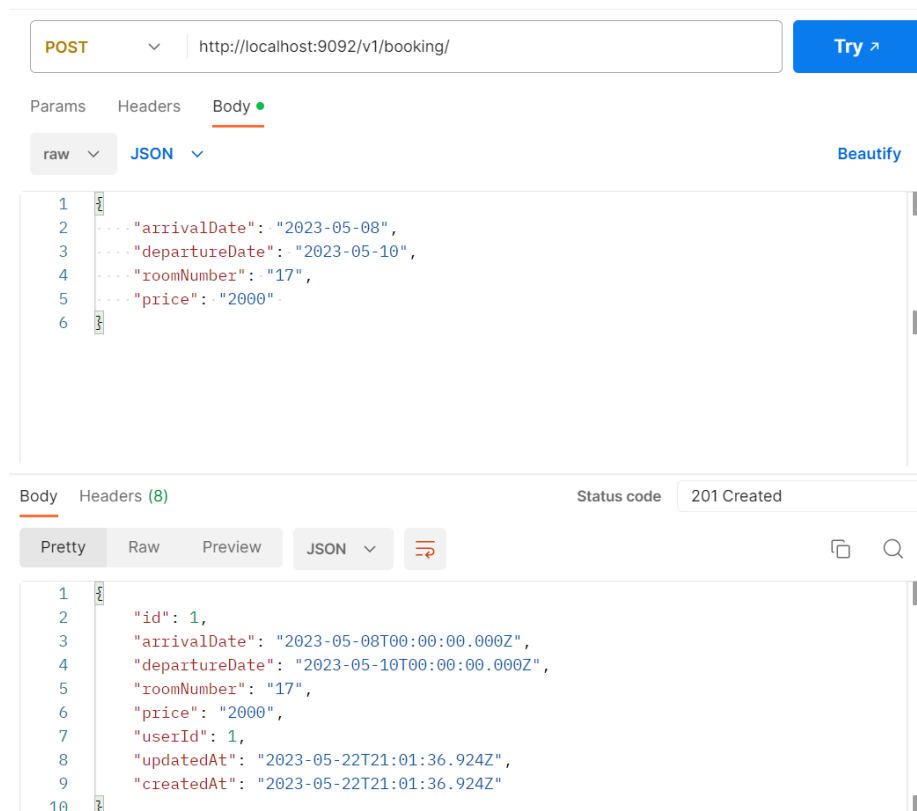
```
1 {
2   ... "roomNumber": "17",
3   ... "floor": "4",
4   ... "capacity": "2",
5   ... "priceOfNight": "1500",
6   ... "description": "Двухместный номер с видом на Неву",
7   ... "hotelId": 2
8 }
```

Body Headers (8) Status code 201 Created

Pretty Raw Preview JSON

```
1 {
2   "roomNumber": "17",
3   "floor": "4",
4   "capacity": "2",
5   "priceOfNight": "1500",
6   "description": "Двухместный номер с видом на Неву",
7   "hotelId": 2,
8   "updatedAt": "2023-05-22T20:16:18.103Z",
9   "createdAt": "2023-05-22T20:16:18.103Z"
10 }
```

Микросервис бронирования:



Вывод:

В ходе лабораторной работы весь монолитный проект для бронирования комнат в отеле был разделен на 3 микросервиса: авторизации, сервис работы с отелями/комнатами и сервис бронирования.