# САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 4

Выполнил:

Викторова Анастасия

M3220d

Проверил: Добряков Д. И.

Санкт-Петербург

2024 г.

#### Задача

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения, а также настроить общение микросервисов между собой посредством RabbitMQ.

### Ход работы

Для начала настроила один из запросов (создание книги). Сделала для него соединение серверов через rabbitmq. Север авторизации отправляет сообщение с телом запроса в очередь, сервер книг принимает его и создает книгу.

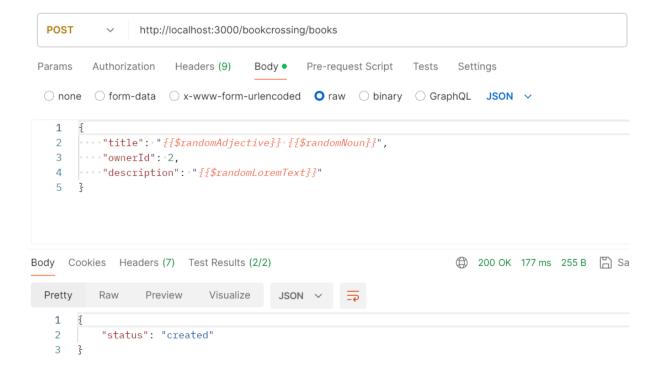
Функция отправки сообщения:

```
lab4 > auth > src > rabbit > TS producer.ts > ❷ default
      import amqplib, {Connection, Channel} from 'amqplib'
      let channel: Channel, connection: Connection
      async function connectRabbit(data: any) {
          try {
               const amqpServer = 'amqp://guest:guest@localhost:5672'
               connection = await amqplib.connect(amqpServer)
               channel = await connection.createChannel()
              await channel.assertQueue('request')
               channel.sendToQueue(
                   Buffer.from(
                      JSON.stringify({
                          ...data,
                          date: new Date(),
          } catch (error) {
               console.log(error)
       export default connectRabbit
```

Прием сообщения:

```
| Index | Dooks | Sinc | Sinc
```

## Результат:



```
Сервис книг получил информацию о запросе {"route":"create","body":{"title":"wireless interface","ownerId":2," description":"Qui rerum error dolorem sit doloremque architecto voluptatem odit aliquam. Quo aut accusamus. C upiditate non provident voluptas sequi sit fugit. Doloremque ex veritatis. Fuga libero in quas et ut repellat id. Qui dolorem unde."},"date":"2024-05-29T10:42:30.669Z"}

Executing (default): INSERT INTO `Books` (`id`,`title`,`ownerId`,`description`,`createdAt`,`updatedAt`) VALUE S (NULL,$1,$2,$3,$4,$5);

Executing (default): SELECT `id`, `title`, `ownerId`, `description`, `createdAt`, `updatedAt` FROM `Books` AS _`Book`;
```

### Далее для каждого сервера создала файл Docker

```
lab4 > books > Dockerfile > ...

1 FROM node
2
3 COPY . /book_server
4
5 WORKDIR /book_server
6
7 EXPOSE 8001
8
9 RUN npm install
10
11 CMD [ "npm", "start" ]
```

### A также общий файл docker-compose.yml

```
lab4 > 🔷 docker-compose.yml
      services:
         container name: rabbitmq
          image: rabbitmq
         restart: always
         restart: always
         depends on:
         - rabbitmq
          ports:
         - 8000:8000
          build:
          depends on:
 23
          - rabbitmq
          - 8001:8001
          restart: always
```

# Вывод

В ходе работы был изучен rabbitmq и приобретены навыки использования Docker