

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

**Лабораторная работа 2: Реализация межсервисного
взаимодействия посредством очередей сообщений**

Выполнил:

Оспельников Алексей

Группа К3440

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2026 г.

Задание:

- подключить и настроить rabbitMQ/kafka;
- реализовать межсервисное взаимодействие посредством rabbitMQ/kafka.

Ход работы

Кафку было решено использовать как брокер сообщений для чата. Первым делом были созданы файлы [rabbit.js](#) и [worker.js](#). Код rabbit показан на рисунке 1

```
const amqp = require('amqplib');

let channel;

async function connectRabbit() {
  const connection = await amqp.connect('amqp://guest:guest@localhost:5672');
  channel = await connection.createChannel();

  await channel.assertQueue('tasks', { durable: true });

  console.log('RabbitMQ connected');
}

function getChannel() {
  if (!channel) throw new Error('RabbitMQ channel not initialized');
  return channel;
}

module.exports = { connectRabbit, getChannel };
```

Рисунок 1 - Код [rabbit.js](#)

Также был реализован отдельный эндпоинт для сообщений

```
app.post('/send-message', async (req, res) => {

  const channel = getChannel();

  const message = JSON.stringify(req.body);

  channel.sendToQueue(
    'tasks',
    Buffer.from(message),
    { persistent: true }
);
```

Вывод

За время работы я успел развернуть RabbitMQ в качестве брокера сообщений для чата. В будущем можно будет прикрутить Websocket и получится полноценный чат