**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

# **Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа 5

Выполнил:

Крохин Владимир

БР1.1

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

**Задача**

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

**Ход работы**

Первым шагом была настройка RabbitMQ в файле docker-compose.ymlю

RabbitMQ настроен с веб-интерфейсом управления на порту 15672, AMQP протоколом на порту 5672, пользователем admin/admin, постоянным хранилищем данных

Далее была создана общей библиотеки для RabbitMQ

В пакете packages/common создана общая библиотека для работы с RabbitMQ:

* Типы и конфигурация (types.ts)
* RabbitMQ клиент (rabbitmq.ts)

Создан класс RabbitMQClient с методами:

* connect() - подключение к RabbitMQ
* publish() - отправка сообщений
* consume() - получение сообщений
* close() - закрытие соединения

Была проведена настройка микросервисов для работы с RabbitMQ

Auth Service:

* Конфигурация (config/rabbitmq.ts)
* Сервис (services/rabbitmq.service.ts)
* Интеграция в контроллере (controllers/user.controller.ts)

Training Service:

* Конфигурация (config/rabbitmq.ts)
* Сервис (services/rabbitmq.service.ts)
* Интеграция в контроллере (controllers/training.controller.ts)

В app.ts каждого сервиса добавлена инициализация RabbitMQ:

Была проведена настройка зависимостей между сервисами (в docker-compose.yml)

В заключение, было проведено тестирование и отладка.

В процессе были решены следующие проблемы:

* Настройка module-alias для корректной работы импортов
* Корректировка Dockerfile для правильного копирования модулей
* Настройка алиасов для разрешения путей

**Вывод**

В ходе работы было достигнуто следующее:

* RabbitMQ успешно настроен и работает (контейнер RabbitMQ запущен с веб-интерфейсом управления)
* Созданы очереди: auth.events и training.events
* Настроены exchange'ы для каждого сервиса
* Межсервисное взаимодействие реализовано:
  + Auth Service отправляет события при создании/обновлении/удалении пользователей
  + Training Service отправляет события при создании/обновлении/удалении тренировок
* Все сервисы подключаются к RabbitMQ при запуске