

# Решение задания на проектирование интеграции через RabbitMQ

## Очереди и их типы Exchange:

- 1. **task\_queue** (Direct Exchange): для хранения всех созданных задач.
- 2. **status\_update\_queue** (Topic Exchange): для хранения информации об обновлении статусов задач.
- 3. **notification\_queue** (Fanout Exchange): для хранения уведомлений, которые нужно отправить сотрудникам.
- 4. **priority\_queue** (Direct Exchange): для хранения задач с высоким приоритетом.
- 5. **audit\_queue** (Direct Exchange): для логгирования всех изменений и аудита.

## Формат сообщений:

- **Задача:**
  - ID (UUID)
  - Описание (string)
  - Статус (string: "новая", "в процессе", "завершена")
  - Дедлайн (timestamp)
  - Ответственный (UUID сотрудника)
- **Сотрудник:**
  - ID (UUID)
  - Имя (string)
  - Должность (string)
  - Отдел (UUID)
- **Отдел:**
  - ID (UUID)
  - Название (string)

- **Уведомление:**

- ID задачи (UUID)
- ID сотрудника (UUID)
- Сообщение (string)

### **План обработки ошибок:**

- Для каждой очереди должен быть Dead Letter Exchange, куда будут попадать сообщения в случае ошибок.
- Реализация автоматического повтора отправки сообщения после определенного времени.
- Логгирование всех ошибок и событий в системе для последующего анализа.

### **Производительность и отказоустойчивость:**

- Использование нескольких инстансов RabbitMQ для обеспечения высокой доступности.
- Использование Quorum Queues для обеспечения отказоустойчивости.

### **Дополнительные функции:**

#### **1. Система уведомлений о дедлайнах:**

- Уведомления будут отправляться автоматически на основе поля "Дедлайн" задачи.
- Сообщение будет помещаться в `notification_queue` и оттуда отправляться ответственному сотруднику.

#### **2. Система приоритетов для задач:**

- Задачи с высоким приоритетом будут помещаться в `priority_queue`.
- Обработчик сообщений из этой очереди будет иметь высший приоритет.

### **Бонус:**

#### **1. Система логгирования всех изменений статусов задач:**

- Каждое изменение статуса задачи будет логгироваться и храниться в `audit_queue` .

## 2. Аудит изменений задач и сотрудников:

- Все изменения, касающиеся задач и сотрудников, будут храниться в `audit_queue` для последующего анализа.