



# Сервис уведомлений

## Тестовое задание для кандидатов-разработчиков

Тестовое задание – дополнительный способ для нас убедиться в вашей квалификации и понять, какого рода задачи вы выполняете эффективнее всего.

Расчётное время на выполнение тестового задания: **3-4 часа**, время засекается **нестрого**. Приступить к выполнению тестового задания можно в **любое удобное для вас время**.

У текущего тестового задания есть только общее описание требований, конкретные детали реализации остаются на усмотрение разработчика. Решение по стеку технологий, которые необходимы для решения задачи, также принимает кандидат по своему усмотрению.

### Задача

Необходимо разработать сервис управления рассылками API администрации и получения статистики.

### Описание

- Необходимо реализовать методы создания новой рассылки, просмотра созданных и получения статистики по выполненным рассылкам.
- Реализовать сам сервис отправки уведомлений на внешнее API.
- Опционально вы можете выбрать любое количество дополнительных пунктов описанных после основного.

Для успешного принятия задания как выполненного достаточно корректной и рабочей реализации требований по основной части, но дополнительные пункты помогут вам продемонстрировать ваши навыки в смежных технологиях.

## Критерии приёмки

- Выполненное задание необходимо разместить в публичном репозитории на [gitlab.com](https://gitlab.com)
- Понятная документация по запуску проекта со всеми его зависимостями
- Документация по API для интеграции с разработанным сервисом
- Описание реализованных методов в формате OpenAPI
- Если выполнено хотя бы одно дополнительное задание - написать об этом в документации, указав на конкретные пункты из списка ниже.

## Основное задание

Спроектировать и разработать сервис, который по заданным правилам запускает рассылку по списку клиентов.

Сущность "рассылка" имеет атрибуты:

- уникальный id рассылки
- дата и время запуска рассылки
- текст сообщения для доставки клиенту
- фильтр свойств клиентов, на которых должна быть произведена рассылка (код мобильного оператора, тег)
- дата и время окончания рассылки: если по каким-то причинам не успели разослать все сообщения - никакие сообщения клиентам после этого времени доставляться не должны

Сущность "клиент" имеет атрибуты:

- уникальный id клиента
- номер телефона клиента в формате **7XXXXXXXXXX** (X - цифра от 0 до 9)
- код мобильного оператора
- тег (произвольная метка)
- часовой пояс

Сущность "сообщение" имеет атрибуты:

- уникальный id сообщения
- дата и время создания (отправки)
- статус отправки
- id рассылки, в рамках которой было отправлено сообщение
- id клиента, которому отправили

Спроектировать и реализовать API для:

- добавления нового клиента в справочник со всеми его атрибутами
- обновления данных атрибутов клиента
- удаления клиента из справочника
- добавления новой рассылки со всеми её атрибутами
- получения общей статистики по созданным рассылкам и количеству отправленных сообщений по ним с группировкой по статусам
- получения детальной статистики отправленных сообщений по конкретной рассылке
- обновления атрибутов рассылки
- удаления рассылки
- обработки активных рассылок и отправки сообщений клиентам

## Логика рассылки

- После создания новой рассылки, если текущее время больше времени начала и меньше времени окончания - должны быть выбраны из справочника все клиенты, которые подходят под значения фильтра, указанного в этой рассылке и запущена отправка для всех этих клиентов.
- Если создаётся рассылка с временем старта в будущем - отправка должна стартовать автоматически по наступлению этого времени без дополнительных действий со стороны пользователя системы.
- По ходу отправки сообщений должна собираться статистика (см. описание сущности "сообщение" выше) по каждому сообщению для последующего формирования отчётов.
- Внешний сервис, который принимает отправляемые сообщения, может долго обрабатывать запрос, отвечать некорректными данными, на какое-то время вообще не принимать запросы. Необходимо реализовать корректную обработку подобных ошибок. Проблемы с внешним сервисом не должны влиять на стабильность работы разрабатываемого сервиса рассылок.

## API внешнего сервиса отправки

Для интеграции с разрабатываемым проектом в данном задании существует внешний сервис, который может принимать запросы на отправку сообщений в сторону клиентов.

OpenAPI спецификация находится по адресу:  
<https://probe.fbrq.cloud/docs>

В этом API предполагается аутентификация с использованием JWT. Токен доступа предоставлен вам вместе с тестовым заданием.

## Дополнительные задания

Опциональные пункты, выполнение любого количества из приведённого списка повышают ваши шансы на положительное решение о приёме

1. организовать тестирование написанного кода
2. обеспечить автоматическую сборку/тестирование с помощью GitLab CI
3. подготовить docker-compose для запуска всех сервисов проекта одной командой
4. написать конфигурационные файлы (`deployment`, `ingress`, ...) для запуска проекта в Kubernetes и описать как их применить к работающему кластеру
5. сделать так, чтобы по адресу `/docs/` открывалась страница со Swagger UI и в нём отображалось описание разработанного API. Пример: <https://petstore.swagger.io>
6. реализовать администраторский Web UI для управления рассылками и получения статистики по отправленным сообщениям
7. обеспечить интеграцию с внешним OAuth2 сервисом авторизации для административного интерфейса.  
Пример: <https://auth0.com>
8. реализовать дополнительный сервис, который раз в сутки отправляет статистику по обработанным рассылкам на email
9. удаленный сервис может быть недоступен, долго отвечать на запросы или выдавать некорректные ответы.  
Необходимо организовать обработку ошибок и откладывание запросов при неуспехе для последующей повторной отправки. Задержки в работе внешнего сервиса никак не должны оказывать влияние на работу сервиса рассылок.

10. реализовать отдачу метрик в формате **prometheus** и задокументировать эндпоинты и экспортируемые метрики
11. реализовать дополнительную бизнес-логику: добавить в сущность "рассылка" поле "временной интервал", в котором можно задать промежуток времени, в котором клиентам можно отправлять сообщения с учётом их локального времени. Не отправлять клиенту сообщение, если его локальное время не входит в указанный интервал.
12. обеспечить подробное логирование на всех этапах обработки запросов, чтобы при эксплуатации была возможность найти в логах всю информацию по
  - id рассылки - все логи по конкретной рассылке (и запросы на арі и внешние запросы на отправку конкретных сообщений)
  - id сообщения - по конкретному сообщению (все запросы и ответы от внешнего сервиса, вся обработка конкретного сообщения)
  - id клиента - любые операции, которые связаны с конкретным клиентом (добавление/редактирование/ отправка сообщения/...)