
Сервис Rapporto Push Notification System

Руководство администратора
по эксплуатации ПО

Содержание

Введение.....	3
Список терминов и сокращений	4
1. Описание программного окружения и компонентов сервиса	6
1.1. Программное окружение	6
1.2. Компоненты сервиса.....	6
2. Описание базы данных сервиса RPNS	9
2.1. Таблица application.....	11
2.2. Таблица apns_application.....	11
2.3. Таблица fcm_application	12
2.4. Таблица hms_application.....	12
2.5. Таблицы service схемы processing.....	12
2.6. Порядок подключения приложения.....	13
3. Конфигурационный файл application.yml	15
1.1. Настройки подключения к базе данных сервиса	15
1.2. Настройки логирования, API, отправителя уведомлений в SDK, http-клиента для отправки сообщений в агрегаторы, обработки статусов отправки/доставки и кеширования	15
1.3. Настройки очередей	18
4. Запуск и остановка сервиса RPNS	20

Введение

Версия документа 2.1 от 21.05.2024.

В документе приведено описание программного окружения, компонентов, базы данных сервиса Rapporto Push Notification System (далее — RPNS), необходимых для его настройки.

При необходимости получения технической поддержки Вы можете обратиться в Службу технической поддержки: support@rapporto.ru.

Для получения консультации по процессу настройки RPNS и его функционирования Вы можете обратиться к следующим специалистам:

Алексей Васильев, Руководитель направления эксплуатации ИТ-инфраструктуры,
sd@rapporto.ru

Максим Кольцов, Инженер службы эксплуатации сервисных платформ,
mkolcov@rapporto.ru

Список терминов и сокращений

Заказчик	— организация-разработчик мобильного приложения, подключенная к платформе Сервис-провайдера с целью использования ее технологических возможностей
Мобильное приложение	— программа Заказчика, предназначенная для установки на мобильное устройство пользователя
Сервис-провайдер	— компания, предоставляющая Заказчику сервисы по приему/передаче push-уведомлений посредством собственной технологической платформы
Установка мобильного приложения	— экземпляр приложения Заказчика, установленного на мобильное устройство конечного пользователя
APNS	— Apple Push Notification Service — облачный сервис, предоставляемый компанией Apple. Предназначен для доставки push-уведомлений, отправляемых сторонними приложениями на IOS-устройства пользователя
FCM	— Firebase Cloud Messaging — кроссплатформенный облачный Google-сервис. Предназначен для доставки push-уведомлений, отправляемых сторонними приложениями на Android-устройства пользователя
HMS	— Huawei Mobile Services Push Kit — облачный Android-сервис. Предназначен для доставки push-уведомлений, отправляемых сторонними приложениями на Android-устройства под управлением ОС Harmony OS
PNS	— Push Notification Service — облачные сервисы-провайдеры push-уведомлений (APNS, FCM, HMS), которые обеспечивают доставку push-уведомлений в приложения на устройства с различными операционными системами (iOS, Android, Harmony)
Push-уведомление	— уведомление в виде всплывающего окна на экране мобильного устройства, предназначенное для информирования пользователей, а также для их взаимодействия с отправителем данных уведомлений — Заказчиком — разработчиком мобильного приложения
Rapporto Push Notification	— сервис, предназначенный для управления отправкой push-уведомлений на мобильные приложения Заказчика, а также для получения статусов по

System результатам отправки

RuStore — российский магазин приложений для устройств на Android.
Предоставляет API для отправки push-уведомлений как через
собственный сервис, так и через облачных провайдеров FCM и HMS

SDK — Software Development Kit — набор функционала (библиотек) и утилит
для разработки программного обеспечения. Заказчик встраивает в
собственное мобильное приложение SDK Сервис-провайдера для
подключения к Push-сервису с целью отправки конечным пользователям
push-уведомлений, а также получения статусов их доставки

1. Описание программного окружения и компонентов сервиса

1.1. Программное окружение

Сервис RPNS установлен на сервере с операционной системой Linux. Для функционирования сервиса используется следующее ПО:

- СУБД PostgreSQL версия 14, не ниже;
- JDK версия 17, не ниже;
- Apache Kafka;
- Nginx (используется как прокси-сервер).

1.2. Компоненты сервиса

Основные настройки сервиса указаны в файле `application.yml`:

- настройки подключения к базе данных сервиса;
- настройки подключения к Kafka (названия топиков, IP брокеров);
- порты для контроллеров сервиса;
- URL к папке с файлами ключей для приложения iOS.

Директория, в которой хранится файл `application.yml`, приведена в таблице 5.

Примечание. В терминале для перехода в требуемую директорию следует выполнить команду `cd /dir_1/.../dir_n/`. Для отображения содержимого директории — команду `ls -lh`.

1.2.1. Директория `/opt/`

Описание приведено в таблице 1.

Таблица 1

Директория	Назначение
<code>/opt/java/</code>	Для хранения программного пакета java.
<code>/opt/kafka/</code>	Для хранения программного пакета брокера Kafka.
<code>/opt/push/</code>	Для хранения настроек сервиса RPNS.
<code>/opt/push-gateway/</code>	Для хранения сертификатов iOS.

1.2.2. Директория /opt/push/

Описание содержимого директории /opt/push/ приведено в таблице 2.

Таблица 2

Компонент	Описание
/opt/push/bin/	Директория для размещения запускаемых файлов.
/opt/push/ios_cert/	Директория для хранения сертификатов iOS.
/opt/push/logs/	Директория для хранения логов.
/opt/push/modules/	Директория для хранения файла с настройками сервиса RPNS.
/opt/push/nginx/	Директория для настройки прокси-сервера.
/opt/push/pid.file	Файл содержит идентификационный номер процесса.

1.2.2.1. Директория /opt/push/bin

Описание файлов в директории /opt/push/bin приведено в таблице 3.

Таблица 3

Файл	Описание
start.sh	Файл запуска сервиса RPNS.
stop.sh	Файл остановки сервиса RPNS.

1.2.2.2. Директория /opt/push/modules

Описание директории /opt/push/modules приведено в таблице 4.

Таблица 4

Директория	Назначение
/opt/push/modules/push/	Для размещения запускаемых файлов.

1.2.2.3. Директория /opt/push/modules/push

Описание директории /opt/push/modules/push приведено в таблице 5.

Таблица 5

Директория	Назначение
/opt/push/modules/push/app.jar	Конфигурационный файл.
/opt/push/modules/push/config/	Директория для хранения файла с настройками сервиса RPNS.
/opt/push/modules/push/config/ application.yml	Файл <code>application.yml</code> с настройками сервиса RPNS.
/opt/push/modules/push/*.log	*.log-файлы.

1.2.3. Директория /opt/push-gateway/

Описание содержимого директории /opt/push-gateway/ приведено в таблице 6.

Таблица 6

Компонент	Описание
/opt/push-gateway/jwt	Хранение сертификатов iOS.

2. Описание базы данных сервиса RPNS

При запуске сервиса RPNS схема базы данных формируется на основе flyway-скриптов. База данных состоит из нескольких логических схем, приведенных в таблице 7. Перед запуском необходимо добавить параметры подключения к базе данных в параметры настроек.

В БД сервиса добавляется следующая информация:

- настройки мобильного приложения;
- настройки маршрутизации для мобильного приложения;
- ключи для отправки на разные мобильные операционные системы (iOS /Android/ Huawei);
- неперсонализированные данные о пользователях;
- информация об установках мобильного приложения пользователей;
- информация о принятых к отправке сообщениях;
- информация об отправках push-уведомлений на установки мобильных приложений пользователей.

Таблица 7

Схема	Описание
application	Содержит данные о приложениях, настройки для мобильных ОС (iOS /Android/ Huawei). Также в таблицах этой схемы хранится информация об установках мобильного приложения на устройства пользователя
common	Информация о платформах (iOS /Android/ Huawei) мобильных устройств
consumer	Содержит данные о пользователях мобильного приложения. Сервис RPNS хранит не персонализированные данные пользователя — идентификатор пользователя и номер телефона. В качестве идентификатора может использоваться логин, UID, email и т. п. Номер телефона передается и хранится в формате 7XXXXXXXXXX
processing	Содержит данные по принятым к отправке и отправленным push-уведомлениям

Для хранения данных приложения используются четыре таблицы:

- application — общая информация о приложении;
- apns_application — данные для доступа к APNS (отправка push на iOS);
- fcm_application — данные для доступа к FCM (отправка push на Android);
- hms_application — данные для доступа к FCM (отправка push для Huawei).

Параметры, необходимые для настройки сервиса RPNS, приведены в таблице 8.

Таблица 8

Наименование	Значение	Место хранения
ApplicationCode	Идентификатор приложения. Требуется для настройки Mobile SDK в мобильном приложении разработчиками Заказчика	Таблица application, поле id
Название мобильного приложения	Используется для отображения в отчетах по сообщениям и на экране поиска сообщения	Таблица application, поле name
APNS certificate	Сертификат APNs для устройств с iOS.	Таблица apns_application, поле team_id
		Таблица apns_application, поле key_id
FCM Server Key	Данные для доступа к функции отправки Mobile Push на платформе Android, Google Play	Таблица fcm_application, поле sender_id
		Таблица fcm_application, поле api_id
		Таблица fcm_application, поле service_account_key
HMS App Id	Данные для доступа к функции отправки Mobile Push на платформе Android, App Gallery	Таблица hms_application, поле application_id
HMS App Secret		Таблица hms_application, поле app_secret

2.1. Таблица application

Используется для хранения основной информации о приложении. Описание полей приведено в таблице 9.

Таблица 9

Наименование поля	Тип данных	Описание
id	int8	Первичный ключ/идентификатор приложения
name	varchar	Название приложения
description	varchar	Описание приложения
creation_date	timestamp	Дата создания приложения
state	varchar	Состояние приложения (Active / NotActive)
installation_count	int8	Количество установок приложения
api_key	varchar	Ключ, используемый для подписи запросов, исходящих от SDK

2.2. Таблица apns_application

Хранит данные для доступа к APNS (отправка push на iOS). Описание полей приведено в таблице 10.

Таблица 10

Наименование поля	Тип данных	Описание
id	int8	Идентификатор приложения
team_id	varchar	Идентификатор Apple Developer Account
key_id	varchar	Идентификатор файла .p8, содержащего ключ доступа
topic	varchar	Идентификатор приложения iOS

2.3. Таблица fcm_application

Содержит данные для доступа к FCM (отправка push на Android). Описание полей приведено в таблице 11.

Таблица 11

Наименование поля	Тип данных	Описание
id	int8	Идентификатор приложения
sender_id	int8	Идентификатор отправителя в firebase cloud messaging
api_id	varchar	Ключ доступа к firebase cloud messaging
service_account_key	varchar	Ключ аккаунта

2.4. Таблица hms_application

Содержит данные для доступа к FCM (отправка push для Huawei). Описание полей приведено в таблице 12.

Таблица 12

Наименование поля	Тип данных	Описание
id	int8	Идентификатор приложения
application_id	varchar	Идентификатор на площадке
app_secret	varchar	Секретный ключ для доступа

2.5. Таблицы service схемы processing

Настройки каналов для отправки сообщений необходимо задавать в таблицах processing.service и processing.service_route.

Таблица service_route содержит совокупность каналов отправки (SDK, api), каждому из которых присвоен определенный приоритет. Таблица service описывает совокупность каналов для отправки сообщений.

Описание полей таблицы `service` приведено в таблице 13.

Таблица 13

Наименование поля	Тип данных	Описание
id	int8	Идентификатор сервиса
name	varchar	Наименование сервиса
description	varchar	Описание сервиса
creation_date	timestamp	Дата создания сервиса
state	varchar	Состояние сервиса (Active / NotActive)

Описание полей таблицы `service_route` приведено в таблице 14.

Таблица 14

Наименование поля	Тип данных	Описание
id	int8	Первичный ключ таблицы
service_id	int8	Внешний ключ <code>processing.service</code>
route_type	varchar	Тип канала (SDK, Api)
route_id	int8	Идентификатор канала. В случае с типом SDK, соответствует идентификатору приложения (<code>application.application.id</code>) в системе
order_number	int4	Приоритет канала

2.6. Порядок подключения приложения

Добавление данных в БД сервиса RPNS необходимо производить при помощи вставки и редактирования строк в соответствующие таблицы.

Для этого необходимо добавить строку в таблицу `application` для получения `id` нового приложения. Далее можно вносить новые данные в другие таблицы, указывая полученный идентификатор как ключ.

Данные для доступа приложения к отправке `push` добавляются в таблицы `apns_application`, `fcm_application`, `hms_application`. Каналы для отправки сообщений задаются в

таблицах `service` и `service_route`.

3. Конфигурационный файл application.yml

Для работы сервиса RPNS используется файл *app.jar*. У файла есть определенные настройки, которые хранятся в подпапке *config* в конфигурационном файле *application.yml*.

Файл *application.yml* содержит настройки, приведенные ниже.

1.1. Настройки подключения к базе данных сервиса

```
server:
  port: ${SERVER_PORT:9998}
  trustedPort: 9999
  forward-headers-strategy: native

spring:
  application:
    name: Push Gateway
  datasource:
    platform: postgres
    driver-class-name: org.postgresql.Driver
    url:
jdbc:postgresql://${POSTGRES_HOST:XXX.XXX.XXX.XXX}:${POSTGRES_PORT:5432}/${POSTGRES_DB_NAME:push-gateway}?ApplicationName=${spring.application.name}
#если СУБД с шифрованием то:
  url:
jdbc:postgresql://${POSTGRES_HOST:XXX.XXX.XXX.XXX}:${POSTGRES_PORT:5432}/${POSTGRES_DB_NAME:push-gateway}?sslmode=require&ApplicationName=${spring.application.name}
    username: ${POSTGRES_USER:<Имя пользователя>}
    password: ${POSTGRES_PASSWORD:"<Пароль>"}
  hikari:
    connection-timeout: 60000
    maximum-pool-size: 10
  autoconfigure:
    exclude:
#
-
org.springframework.boot.autoconfigure.web.servlet.error.ErrorMvcAutoConfiguration
- org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration
-
org.springframework.boot.actuate.autoconfigure.security.servlet.ManagementWebSecurityAutoConfiguration
```

1.2. Настройки логирования, API, отправителя уведомлений в SDK, http-клиента для отправки сообщений в агрегаторы, обработки статусов отправки/доставки и кеширования

```
# логирование: путь, уровни логирования
logging:
```

```
#file: /var/log/push-gateway/push-gateway.log
level:
  org.springframework: info
  org.apache.http: info
  com.zgr.service.push: debug
  #org.springframework.jdbc.core.JdbcTemplate: debug
file:
  name: /opt/push/logs/push-gateway.log
logback:
  rollingpolicy:
    max-file-size: 10000MB
    max-history: 30
oracle:
  jdbc:
    level: INFO

# настройки API
api:
  validation:
    phone-regexp: \d{1,20}
    max-text-length: 1000
    min-ttl-in-sec: 10
    max-ttl-in-sec: 86400
  key: Ahk9uo
  security-enabled: true
  client:
    base-url: /client/api
  sdk:
    base-url: /sdk/api
  aggregator:
    base-url: /external/api

# настройки рассылщика уведомлений в SDK
notification:
  fcm:
    http:
      host: https://fcm.googleapis.com
      connect-timeout-millis: 10000
      read-timeout-millis: 10000
    xmpp:
      enabled: false
      host: fcm-xmpp.googleapis.com
      port: 5236
  apns:
    host: ${NOTIFICATION_APNS_HOST:api.push.apple.com}
    keystore-path: ${NOTIFICATION_APNS_KEYSTORE_PATH:/opt/push/ios_cert/}
    connect-timeout-millis: 10000
    graceful-timeout-millis: 10000
  hms:
    http:
```

```
auth: https://oauth-login.cloud.huawei.com/oauth2/v3/token
send: https://push-api.cloud.huawei.com
connect-timeout-millis: 10000
request-timeout-millis: 10000
socket-timeout-millis: 10000
max-total-connections: 100
default-keep-alive-time-millis: 30000
rustore:
  http:
    send: https://vkpns.rustore.ru/v1/projects/{projectId}/messages:send
    connect-timeout-millis: 30000
    request-timeout-millis: 30000
    socket-timeout-millis: 30000
    max-total-connections: 50
    default-keep-alive-time-millis: 20000
    close-idle-connection-wait-time-secs: 30

# настройки http клиента для отправки сообщений в агрегаторы
aggregator:
  http:
    connect-timeout-millis: 30000
    request-timeout-millis: 30000
    socket-timeout-millis: 30000
    max-total-connections: 50
    default-keep-alive-time-millis: 20000
    close-idle-connection-wait-time-secs: 30

# настройки обработки статусов отправки/доставки
status:
  retry-attempts: 3
  retry-delay-seconds: 30
  processing:
    delay-millis: 5000
    batch-size: 1000
    executor:
      core-pool-size: 10
      max-pool-size: 10
      keep-alive-time: 20

# настройки кеширования
cache:
  instance:
    ttl: 44
    unit: MINUTES
    concurrency: 5
    max-size: 24
apns:
  ttl: 720
  unit: MINUTES
  concurrency: 5
```

```
max-size: 24
dataloading:
number-hours-for-loading-notification: 24
concurrent:
```

1.3. Настройки очередей

Указываются настройки подключения к Kafka:

```
# настройки очередей
broker:
  use: kafka
  kafka:
    broker: XXX.XXX.XXX.XXX:9092 #указать ip адрес сервера кафки, если кафка
кластерная то формат: ipserver1:9092,ipserver2:9092,ipserver3:9092
    status:
      topic:
        name: status-prod0_1 #изменяемый параметр
        groupId: status-group-id-prod0_1 #изменяемый параметр
        max-poll-interval-ms: 30000
        max-poll-records: 100
        concurrent: 5
        partitions: 5
        replicas: 3 #значение параметра должно быть на единицу меньше, чем
существующее количество нод кафки
        retention-ms: 1680000
    aggregator:
      request:
        topic:
          name: external-api-request-prod0_1
          groupId: external-api-group-id-prod0_1
          max-poll-interval-ms: 30000
          max-poll-records: 100
          concurrent: 5
          partitions: 5
          replicas: 3
          retention-ms: 1680000
    sdk:
      transaction:
        topic:
          name: sdk-transaction-prod0_1
          groupId: sdk-transaction-group-id-prod0_1
          max-poll-interval-ms: 30000
          max-poll-records: 100
          concurrent: 5
          partitions: 5
          replicas: 3
          retention-ms: 1680000
      notification:
```

```
android:
  topic:
    name: android-push-notification-prod0_1
ios:
  topic:
    name: ios-push-notification-prod0_1
huawei:
  topic:
    name: huawei-push-notification-prod0_1
rustore:
  topic:
    name: rustore-push-notification-prod0_1
topic:
  name: push-notification-prod0_1
  groupId: push-notification-group-id-prod0_1
  max-poll-interval-ms: 30000
  max-poll-records: 100
  concurrent: 5
  partitions: 5
  replicas: 3
  retention-ms: 1680000
```

4. Запуск и остановка сервиса RPNS

Для запуска сервиса RPNS необходимо выполнить команду:

```
/opt/push/bin/start.sh
```

Для остановки сервиса RPNS необходимо выполнить команду:

```
/opt/push/bin/stop.sh
```