Rapporto Push Notification System

Инструкция по установке и настройке

Версия 1.0

Москва 2023

Содержание

Введение

Версия документа 1.0. от 28.11.2023.

Документ предназначен для технических специалистов организации Заказчика (далее — Заказчик), осуществляющих самостоятельную интеграцию сервиса Rapporto Push Notification System (далее — RPNS) в свой периметр.

В документе приведено описание процесса установки и настройки сервиса RPNS в периметр Заказчика. Сервис RPNS интегрируется с системой Заказчика или Платформой Сервис-провайдера и принимает запросы на отправку push-уведомлений в мобильные приложения.

Для интеграции и штатной эксплуатации системы необходимо привлечение следующего персонала:

- системный администратор;
- администратор баз данных.

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
Пользователь	— конечный получатель push-уведомлений на свое устройство с установленным мобильным приложением Заказчика
Сервис-провайдер	— компания, предоставляющая Заказчику сервисы по приему/передаче push-уведомлений посредством собственной технологической платформы
Токен	— уникальный идентификатор устройства. Токен генерируется и выдается облачным сервисом PNS (APNS, FCM, HMS, RuStore) в зависимости от операционной системы устройства
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 System	— сервис, предназначенный для управления отправкой push-уведомлений на мобильные приложения Заказчика, а также для получения статусов по 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. Назначение и область применения

1.1. Область применения

Сервис RPNS позволяет управлять отправкой push-уведомлений на мобильные приложения Заказчика, а также получать статусы доставки этих уведомлений по результатам отправки.

1.2. Назначение сервиса

Сервис RPNS выполняет следующие функции:

- ведет и актуализирует регистрационную информацию (устройства, токены, номера телефонов, подписки и пр.);
- принимает уведомления, переданные Заказчиком в транспортную систему;
- отправляет уведомления конечным пользователям через облачные сервисы PNS;
- получает статусы доставки;
- возвращает финальные статусы доставки;
- хранит историю уведомлений;
- хранит не персонализированную информацию о пользователях мобильных приложений (идентификатор и номер телефона).

Сервис RPNS взаимодействует с SDK Сервис-провайдера, встраиваемым в мобильное приложение. Для отправки push-уведомлений Заказчику необходимо интегрировать в свое мобильное приложение комплект для разработки ПО (SDK) Сервис-провайдера. SDK Сервис-провайдера поддерживает работу с push-уведомлениями в следующих операционных системах (далее – ОС):

- IOS 11 и выше;
- Android 5.0 и выше;
- HarmonyOS.

2. Архитектура сервиса Rapporto Push Notification System

2.1. Схема взаимодействия сервиса Rapporto Push Notification System с системой Заказчика

Функциональная схема системы отправки push-уведомлений системой Заказчика с использованием сервиса RPNS приведена на рис. № 1.

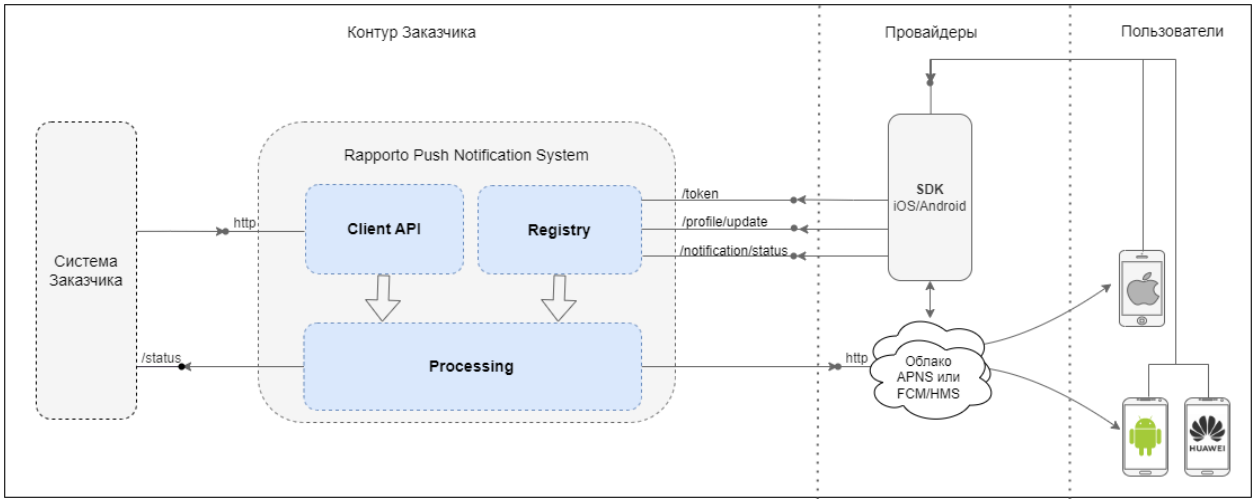

Рисунок № 1

Система Заказчика – выполняет запросы с использованием технологии REST для формирования задачи на отправку push-уведомлений и получает статусы доставки.

SDK – это библиотеки с наборами методов и функций, которые встраиваются в мобильное приложение Заказчика и позволяют взаимодействовать с сервисом RPNS для отправки на мобильные устройства пользователей push-уведомлений Заказчика.

2.2. Схема взаимодействия сервиса Rapporto Push Notification System с Платформой Сервис-провайдера

Функциональная схема системы отправки push-уведомлений по интеграционным подключениям к Платформе Сервис-провайдера с использованием сервиса RPNS в контуре Заказчика приведена на рис. № 2.

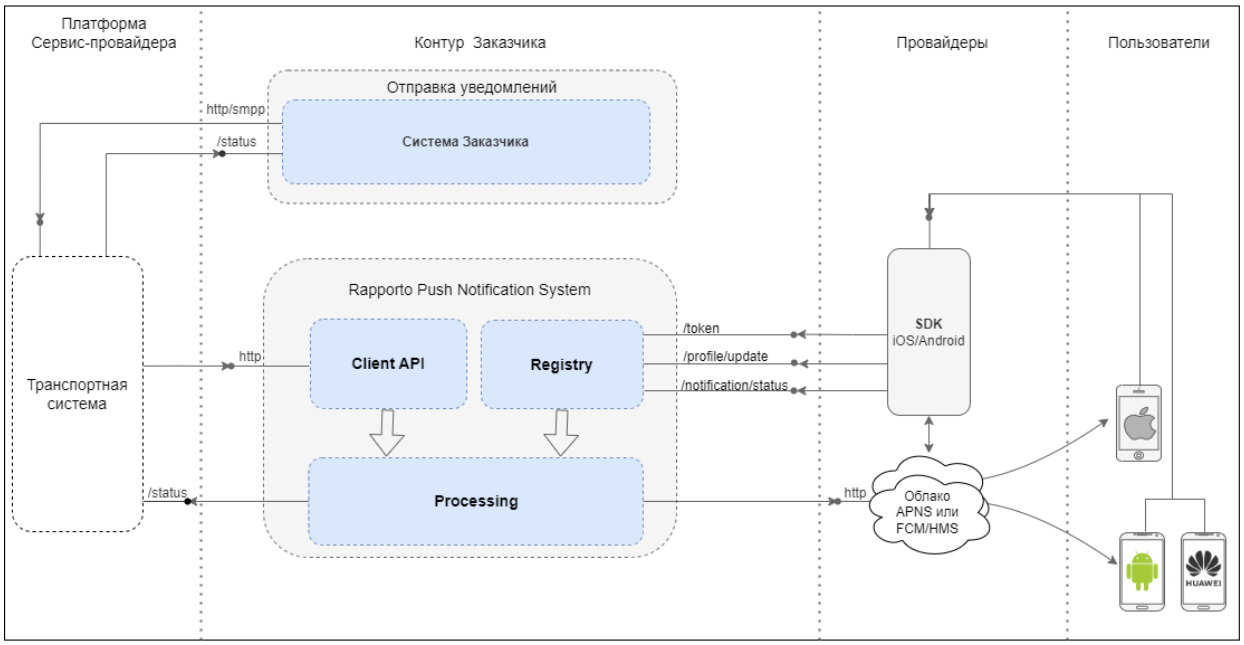

Рисунок № 2

В качестве Транспортной системы возможно использование многофункциональной Платформы Сервис-провайдера.

Платформа предназначена для генерации, маршрутизации, формирования статистики и обработки различных видов сообщений. Возможен прием запросов на отправку как одиночных push-уведомлений, так и push-уведомлений в составе каскада из различных типов сообщений (Viber, WhatsApp, VK, SMS).

В этой схеме Система Заказчика выполняет запросы по технологии REST и протоколу SMPP к программному интерфейсу Платформы Сервис-Провайдера.

3. Запуск сервиса Rapporto Push Notification System

3.1. Требования к окружению

Сервис RPNS устанавливается на сервере с операционной системой Linux. Для функционирования сервиса необходимо использовать следующее ПО:

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

Далее описан алгоритм действий для успешного запуска сервиса RPNS в периметре Заказчика.

3.2. Создание директорий для хранения компонентов окружения

Все действия по выполнению команд в терминале на сервере с операционной системой Linux должны выполняться с правами суперпользователя root.

Необходимо создать директории, в которых будут храниться компоненты требуемого окружения.

3.3. Создание базы данных сервиса Rapporto Push Notification System

Сервис использует СУБД PostgreSQL не ниже 14 версии. Для настройки базы данных необходимо создать пользователя с помощью команды:

```
su postgres -c 'psql'
create role <Имя пользователя> with login password '<Задать пароль>';
```

Далее создать пустую базу данных с владельцем:

```
create database <Имя базы, например, push_gateway> with owner=<Имя пользователя>;
```

Для настройки сервиса RPNS необходимо в созданной базе данных в пользовательском интерфейсе приложения вручную создать пустую схему admin с помощью команды:

```
CREATE SCHEMA "admin" AUTHORIZATION <Имя пользователя>;
GRANT ALL ON SCHEMA "admin" TO <Имя пользователя>;
```

База данных сервиса содержит следующую информацию:

- настройки мобильного приложения;
- настройки маршрутизации для мобильного приложения;
- ключи для отправки на разные мобильные операционные системы (iOS /Android/ Huawei);
- не персонализированные 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.

3.4. Настройка окружения

Все действия по выполнению команд в терминале должны выполняться с правами суперпользователя root.

3.4.1. Установка Java

После создания требуемых директорий и базы данных необходимо установить java, выполнив следующие действия:

1. Перейти в ранее созданную директорию.
2. Загрузить и распаковать архив, содержащий java:

   ```
   wget  https://download.java.net/openjdk/jdk17/ri/openjdk-17+35_linux-x64_bin.tar.gz
   tar xvf openjdk-17+35_linux-x64_bin.tar.gz
   ```
3. Удалить архив:

   ```
   rm openjdk-17+35_linux-x64_bin.tar.gz
   ```
4. Установить программный пакет java:

   ```
   apt-get install openjdk-17-jdk
   ```
5. Установить путь к переменной окружения JAVA_HOME:

   ```
   export JAVA_HOME= /opt/java/jdk-17
   ```

   Для проверки пути необходимо ввести команду `echo $JAVA_HOME`
6. Добавить каталог JAVA bin в переменную PATH:

   ```
   export PATH=\$PATH:\$JAVA_HOME/bin
   ```

   Для проверки использовать команду `echo $PATH`

7. После завершения установки OpenJDK и переменных JAVA_HOME и PATH проверить корректность установки java:

```
cd /opt/java/jdk-17/bin
java -version
```

3.4.2. Установка и настройка Kafka

Для установки брокера Kafka необходимо загрузить архив Kafka не ниже версии 3.5.0. и распаковать его, используя команду:

```
cd /opt/
wget https://downloads.apache.org/kafka/3.5.0/kafka_2.13-3.5.0.tgz
tar xzf kafka_2.13-3.5.0.tgz
```

После распаковки архива удалить его командой `rm kafka_2.13-3.5.0.tgz`

Для настройки брокера в качестве сервиса необходимо создать 2 юнит-файла – один для Zookeeper, второй для Kafka, используя следующие команды:

- для Zookeeper:

```
nano /etc/systemd/system/zookeeper.service
----
[Unit]
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=root
Environment="JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64/"
Environment=JMX_PORT=xxxx
ExecStart=/opt/kafka/bin/zookeeper-server-start.sh
/opt/kafka/config/zookeeper.properties

ExecStop=/opt/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal


[Install]
WantedBy=multi-user.target
----
```

- для Kafka:

```
nano /etc/systemd/system/kafka.service
----
[Unit]
Requires=zookeeper.service
After=zookeeper.service

[Service]
Type=simple
User=root
Environment="JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64/"
Environment=JMX_PORT=xxxx
ExecStart=/bin/sh -c '/opt/kafka/bin/kafka-server-start.sh
/opt/kafka/config/server.properties > /opt/kafka/kafka.log 2>&1'
ExecStop=/opt/kafka/bin/kafka-server-stop.sh
Restart=on-abnormal

[Install]
WantedBy=multi-user.target
----
```

Чтобы запустить Zookeeper и Kafka, необходимо ввести команды:

```
systemctl start zookeeper
systemctl start kafka
```

Далее описан вариант тестирования работы Kafka. Для этого необходимо создать топик с помощью команды:

```
cd /opt/kafka
bin/kafka-topics.sh --create --topic <имя топика> --bootstrap-server
localhost:xxxx --replication-factor 1 --partitions 4
```

3.4.3. Установка и настройка Nginx

Для установки Nginx необходимо воспользоваться официальной документацией от производителя: https://nginx.org/en/docs/install.html

В конфигурационном файле **.conf* задать настройки, представленные ниже:

```
server {
  listen       443 reuseport ssl http2;
  listen       [::]:443 ssl http2;
  server_name  xxxxx.xxx;
  root         /usr/share/nginx/html;
```

```
            ssl_certificate "xxxxx.xxx.crt";
        ssl_certificate_key "xxxxx.xxx.key";

            ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;

            ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;

             location /sdk/api {
                proxy_pass xxx.x.x.x:xxxx
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                }
    }
```