

**Описание функциональных характеристик  
программного обеспечения и информация,  
необходимая для установки и эксплуатации  
программного обеспечения  
iSimpleMessageGate.ActivePush**

## Оглавление

<b>1</b>	<b>Функциональные характеристики</b>	<b>3</b>
1.1	Ключевые принципы разработки ДБО iSimpleLab	3
1.2	Ключевые функции	4
<b>2</b>	<b>Информация перед установкой</b>	<b>5</b>
2.1	Требования к сопутствующему программному и аппаратному обеспечению	5
2.2	Конфигурация оборудования	6
2.2.1	Типовая конфигурация внедрения iSimpleMessageGate	6
2.2.2	Запуск с изоляцией всех компонент на выделенных виртуальных машинах	7
2.3	Логическая схема развертывания	9
<b>3</b>	<b>Подготовка к установке</b>	<b>11</b>
3.1	Загрузка дистрибутива системы	11
3.1.1	Структура пакета установки	11
3.2	Подготовка баз данных	13
3.2.1	Перечень баз данных	13
3.2.2	Настройка параметров подключения к базе данных	14
3.3	Установка сервера очередей RabbitMQ	15
3.3.1	Настройка для Windows	15
3.3.2	Настройка для Linux/Unix	15
3.3.3	Настройки для удобства администрирования	16
3.3.4	Кластеризация сервера очередей RabbitMQ	16
<b>4</b>	<b>Установка системы iSimpleMessageGate</b>	<b>20</b>
4.1	Основные компоненты системы	20
4.1.1	Администрирование – iSimpleMsgAdmin (обязательный компонент)	20
4.1.2	Диспетчер сообщений – iSimpleMsgRouter (обязательный компонент)	21
4.1.3	Сервис приема сообщений (опциональный компонент)	22
4.1.4	Сервис отправки сообщений (выбираем опционально все возможные отправщики или часть)	24
4.1.5	База данных перенесённых номеров (опционально)	25
4.1.6	Сервис проверки доступности сети Интернет на серверах отправщиков (опционально)	25

# 1 Функциональные характеристики

## 1.1 Ключевые принципы разработки ДБО iSimpleLab

- Кроссплатформенный сервер приложений, позволяющий удовлетворить ИТ-Стратегию кредитной организации любого масштаба;
- Промышленный открытый API, реализующий концепцию OpenAPI и позволяющий разработать своими силами или силами экспертной команды независимый гибкий слой интерфейса, а также прямую интеграцию с клиентскими информационными системами;
- Модульная архитектура, позволяющая реализовать независимые и отказоустойчивые компоненты для более стабильной работы всей платформы;
- Портал – автономный front-end платформы iSimpleBank 2.0, который взаимодействует с back-end системы посредством Rest-API. Данное решение позволяет реализовать любые замыслы дизайнеров и открывает новые горизонты по развитию и запуску в личном кабинете Банка любых внешних сервисов не связанных с функционалом классического ДБО;
- Использование инновационных средств безопасности, совмещающих удобство работы с ними и универсальность по отношению к каналу работы (web-кабинет, мобильное приложение на любой платформе, либо внешний портал, в который встроены сервисы банка);
- Готовые средства и возможности интеграции практически со всеми АБС, CRM, карточными процессингами, платежными системами и системами денежных переводов.

Платформа ДБО iSimpleBank 2.0 построена на современных технологиях и использует стандартные промышленные сервера приложений и СУБД, как коммерческие, так и с открытым кодом:

- Java в качестве кросс-платформенного языка разработки;
- Операционные системы: Windows, Unix, Linux;
- Сервера приложений: JBoss Application Server, Oracle WebLogic, IBM WebSphere Application Server;
- СУБД: Oracle, PostgreSQL, Microsoft SQL Server, Firebird/RED Database;
- XML-описания ресурсов системы, таких как конфигурации, визуальные формы, меню, схемы документов, языковые ресурсы;
- Groovy – открытые скрипты, используемые в системе;

Использование веб-сервисов для задач интеграции с другими системами. Клиентская часть интернет-клиента выполнена на технологии полностью тонкого клиента, и работает на следующих операционных системах и браузерах:

- Операционные системы: Windows, Unix, Linux, MacOS, iOS, Android;
- Браузеры: IE, Firefox, Chrome, Safari, Opera, Android и др в зависимости от выбранной Web-платформы.

## 1.2 Ключевые функции

Модуль **iSimpleMessageGate.ActivePush** является дополнительным модулем к базовому модулю **iSimpleMessageGate** и предоставляет функционал push-уведомлений, в которые может быть вставлен медиа контент в виде ссылок на интернет-ресурс или конкретное место внутри ДБО (мобильного приложения).

Система **iSimpleMessageGate** рассылки сообщений используется для гарантированной передачи сообщений от системы ДБО, а также от смежных, участвующих в интеграции систем по различным каналам (таким как SMS, Push-уведомления и e-mail). Путем автоматического перенаправления части SMS-сообщений в Push-уведомления достигается существенное снижение затрат. В случае, если отправка не удалась или клиент за установленный срок не получил Push-уведомление, клиенту отправляется дублирующее SMS-сообщение.

## 2 Информация перед установкой

### 2.1 Требования к сопутствующему программному и аппаратному обеспечению

Требования к сопутствующему программному\* и аппаратному обеспечению системы iSimpleMessageGate приведены в таблице:

Компонента	Тип категории	Наименование и версия
<b>Требования к программному обеспечению</b>		
<b>Система iSimpleMessageGate</b>	Операционная система	Microsoft Windows Linux, Unix
	Среда виртуальной машины	Oracle Java Runtime Environment (JRE) версия 1.8
<b>Сервер очередей</b>	Сервер очередей	Rabbit MQ 3.7.7
	Операционная система	Microsoft Windows Linux, Unix
<b>Сервер СУБД</b>	СУБД	Oracle Database 10g R2, Oracle Database 11g R2, Oracle Database 12c R1
		Microsoft SQL Server 2012 и выше
		PostgreSQL 9.6, 10 и выше
	Операционная система	Соответствующая выбранной СУБД (см. сопроводительную документацию к СУБД)
<b>Система ДБО iSimpleBank 2.0</b>	Опционально	Наличие в системе ДБО iSimpleBank 2.0 библиотек с версиями не ниже cust-notify-xmpp:0.15.0 cust-notify-isimplemsg:0.15.0
<b>Мобильные приложения</b>	Опционально	Для мобильных устройств на iOS и Android, работающих с push-сообщениями, должны использоваться следующие REST-сервисы: <a href="#">Регистрация устройства для push уведомлений</a> (в зависимости от используемого облака PUSH для МП будет отличаться параметр <i>platform</i> , подробнее см. в описании сервиса) <a href="#">Уведомление от устройства о смене адреса для push уведомлений</a> <a href="#">Удаление устройства для push уведомлений</a> <a href="#">Проверка регистрации устройства для push уведомлений</a> <a href="#">Изменение статуса push уведомления (для устройств, работающих с сервером сообщений)</a>

Компонента	Тип категории	Наименование и версия
		Для определения PUSH-отправщиков, которые понадобятся в зависимости от типа МП, см. схему по ссылке: <a href="#">Сервисы отправки сообщений</a>
<b>Рабочее место сотрудника банка</b>	Интернет-браузер	Internet Explorer 11 и выше
		Firefox - актуальная версия (обновляется автоматически)
		Google Chrome - актуальная версия (обновляется автоматически)
		Safari 9 и выше
	Opera - актуальная версия (обновляется автоматически)	
	Операционная система	Все операционные системы, поддерживаемые браузером. Список поддерживаемых операционных систем смотрите в сопроводительной документации к соответствующим браузерам

*\* При выборе типа и версии программного обеспечения рекомендуется ориентироваться на разработчика ПО и аппаратного обеспечения и использовать актуальные версии и модификации, предназначенные для промышленной эксплуатации.*

## 2.2 Конфигурация оборудования

- [Типовая конфигурация внедрения iSimpleMessageGate](#)
- [Запуск с изоляцией всех компонент на выделенных виртуальных машинах](#)

В разделе представлена конфигурация оборудования для установки системы **iSimpleMessageGate**.

### 2.2.1 Типовая конфигурация внедрения iSimpleMessageGate

СЕРВЕР 1: СУБД

<b>Сервер СУБД</b>	Процессор (ядра): CPU 3+ GHz 8 ядер Оперативная память: 32 Гб Жесткие диски: SAS ibm 10000rpm
--------------------	---

Размер быстрого дискового массива зависит от количества сообщений и времени необходимого хранения оперативной информации.

Рекомендуется не менее 1ТБ (для оперативного хранения данных за 1 год)

Размер архивного дискового массива зависит от периода эксплуатации системы и времени необходимого хранения архивной информации.

Рекомендуется не менее 5ТБ (для архивного хранения данных за 5 лет)

Для отказоустойчивости рекомендуется использование второго сервера, размещенного в другом дата-центре и настройка кластера БД или standby сервера для быстрого восстановления работы при выходе из строя основного сервера

## СЕРВЕР 2: RabbitMQ + Компоненты **iSimpleMessageGate**

<b>Сервер RabbitMQ + iSimpleMessageGate</b>	<p>При развертывании на одном сервере следующего набора компонент</p> <ul style="list-style-type: none"> <li>- RabbitMQ</li> <li>- isimple-msg-router (оркестратор сообщений)</li> <li>- isimple-msg-admin (административный интерфейс)</li> <li>- isimple-msg-bdprn (база данных перенесенных номеров)</li> <li>- приемников сообщений разных или одинаковых типов, не более 8 штук (2 REST, 2 FILE, 4 SMPP)</li> <li>- отправщиков сообщений разных или одинаковых типов, не более 8 штук (2 APNS, 2 FIREBASE, 4 SMPP)</li> </ul> <p>Конфигурация оборудования</p> <p>Процессор (ядра): CPU 3+ GHz 12 ядер</p> <p>Оперативная память: 16 Гб</p> <p>Жесткие диски: SAS ibm 10000rpm (объем зависит от ограничения на размер записи лог-файлов)</p>
---	---

Для отказоустойчивости рекомендуется использование второго сервера, размещенного в другом дата-центре, для кластеризации RabbitMQ и дублирования всех компонентов **iSimpleMessageGate**

### 2.2.2 Запуск с изоляцией всех компонент на выделенных виртуальных машинах

<p><b>isimple-msg-router</b> оркестратор сообщений</p>	<p>Процессор (ядра): CPU 3+ GHz 2 ядра</p> <p>Оперативная память для Java-процесса: 1 Гб</p> <p>Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи логов)</p> <p>Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах</p>
<p><b>isimple-msg-admin</b> административный интерфейс</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро</p> <p>Оперативная память для Java-процесса: 1 Гб</p> <p>Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи логов)</p> <p>Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах</p> <p>В качестве балансировки http-запросов рекомендуется Nginx</p>

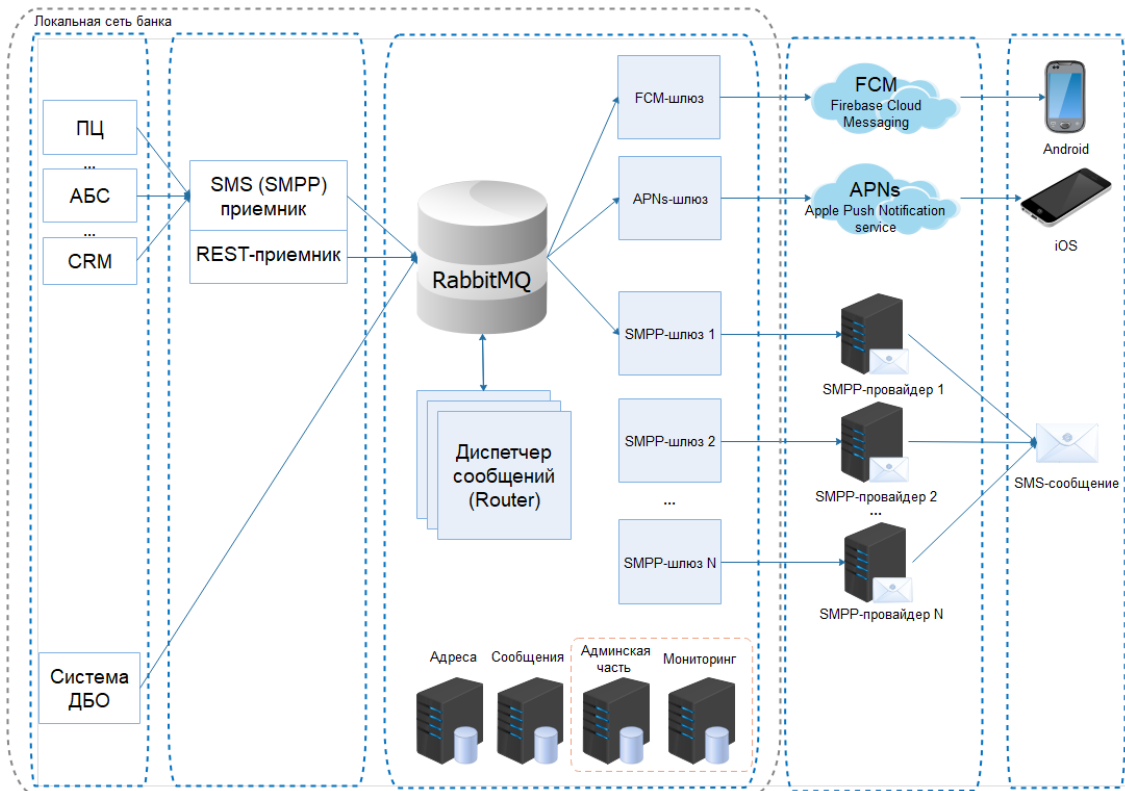
<p><b>isimple-msg-smpp-receiver</b> smpp приемник</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 500 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов (основной и резервный) в разных дата-центрах или на разных физических серверах для каждого клиентского приложения, которое отправляет сообщения по SMPP протоколу</p>
<p><b>isimple-msg-rest-receiver</b> rest-api приемник</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 500 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах для каждого клиентского приложения, которое отправляет сообщения по REST-API протоколу В качестве балансировки http-запросов рекомендуется Nginx</p>
<p><b>isimple-msg-file-receiver</b> файловый приемник</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 500 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов)</p>
<p><b>isimple-msg-apns-sender</b> APNS-отправщик</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 500 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах для каждого уникального проекта мобильного приложения</p>
<p><b>isimple-msg-firebase-sender</b> FIREBASE-отправщик</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 500 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах для каждого уникального проекта мобильного приложения</p>
<p><b>isimple-msg-smpp-sender</b> SMPP-отправщик</p>	<p>Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 750 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется использовать несколько сервисов к разным SMPP-провайлерам, сервисы размещаются в разных дата-центрах или на разных физических серверах</p>



<b>isimple-msg-email-sender</b> EMAIL-отправщик	Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 500 Мб Жесткие диски:HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах
<b>isimple-msg-bdprn</b> База данных перенесенных номеров	Процессор (ядра): CPU 3+ GHz 1 ядро Оперативная память для Java-процесса: 750 Мб Жесткие диски: HDD 100 Гб (объем зависит от ограничения на размер записи лог-файлов) Для отказоустойчивости рекомендуется запускать не менее 2-х сервисов в разных дата-центрах или на разных физических серверах
<b>Сервер СУБД</b>	Процессор (ядра): CPU 3+ GHz 8 ядер Оперативная память: 32 Гб Жесткие диски: SAS ibm 10000rpm Для отказоустойчивости рекомендуется настройка кластера БД или standby сервера в другом дата-центре для быстрого восстановления работы при выходе из строя основного сервера
<b>Сервер RabbitMQ</b>	Процессор (ядра): CPU 3+ GHz 2 ядра Оперативная память: 4 Гб Жесткие диски: SAS ibm 10000rpm Для отказоустойчивости рекомендуется настройка кластера RabbitMQ, состоящего не менее чем из 2-х серверов в разных дата-центрах или на разных физических серверах
<b>Сервер сбора метрик и мониторинга</b>	Требования не предъявляются Используются сервисы, принятые в банке При их отсутствии предлагается prometheus, grafana
<b>Сервер сбора и просмотра логов</b>	Требования не предъявляются Используются сервисы, принятые в банке При их отсутствии предлагается filebeat, logstash, elastic search, kibana

## 2.3 Логическая схема развертывания

На рисунке представлена логическая схема развертывания системы **iSimpleMessageGate**:



Описание основных компонентов см. в разделе [Основные компоненты системы iSimpleMessageGate](#).

## 3 Подготовка к установке

- [Загрузка дистрибутива системы](#)
  - а. [Структура пакета установки](#)
- [Подготовка баз данных](#)
- [Установка Java Virtual Machine](#)
- [Установка сервера очередей RabbitMQ](#)

При установке сопутствующего программного обеспечения ориентируйтесь на документацию, поставляемую разработчиком программного обеспечения.


### 3.1 Загрузка дистрибутива системы

Система **iSimpleMessageGate** поставляется в виде пакета установки (он же дистрибутив). Компания iSimpleLab размещает дистрибутивы на FTP-ресурсе компании (<ftp://support.isimplelab.com>). Персонализированные дистрибутивы доступны для загрузки под индивидуальным логином и паролем, предоставленным банку. Логин и пароль к FTP предоставляет Центр сервисов компании (<https://service.isimplelab.com>) по запросу.

Состав дистрибутива рассматривается в подразделе [Структура пакета установки](#).

#### 3.1.1 Структура пакета установки

Система **iSimpleMessageGate** поставляется пакетами установки отдельных [КОМПОНЕНТОВ](#) системы.

Имя	Размер	Тип	Изменено
		Родительский кат...	17.07.2019 9:25:43
bin		Папка с файлами	16.07.2019 19:39:26
core		Папка с файлами	16.07.2019 19:39:26
doc		Папка с файлами	16.07.2019 19:39:26
license		Папка с файлами	16.07.2019 19:39:26
logs		Папка с файлами	16.07.2019 19:39:26
receiver		Папка с файлами	16.07.2019 19:39:34
sender		Папка с файлами	16.07.2019 19:39:40
test		Папка с файлами	16.07.2019 19:39:50

Для удобства компоненты системы разбиты по каталогам:

- bin - включает служебные файлы, необходимые для запуска каждого компонента.
- core - включает два основных сервиса системы (Сервис администрирования и Диспетчер сообщений) и дополнительные сервисы общего характера (Сервис базы данных перенесённых номеров, Сервис проверки сети, Сервис архивации сообщений и Сервис администрирования архивных сообщений).
- doc - включает файл с описанием системы **iSimpleMessageGate** и ссылками на раздел документации.
- license - каталог для файла лицензии - license.lic.
- logs - каталог, в который настроена запись логов каждого компонента по умолчанию.
- receiver - включает пакеты установки компонентов приёма сообщений: REST-приёмник и SMPP-приёмник.

- sender - включает пакеты установки компонентов отправки сообщений: APNS-отправщик, EMAIL-отправщик, FCM-отправщик, FIREBASE-отправщик, SMPP-отправщик.
- test - включает пакеты установки компонентов для тестирования системы: Генератор сообщений, Отправщик PUSH-уведомлений, Тестовый отправщик сообщений, Эмулятор SMPP-сервера.

Каждый пакет установки (дистрибутив) компонента системы **iSimpleMessageGate** имеет постоянный состав ресурсов, назначение которых рассматривается в таблице.

Ресурс	Описание
isimple-msg-build-<номер версии> Далее обозначается как <пакет установки>	Каталог с пакетом установки (дистрибутивом) системы <b>iSimpleMessageGate</b> . Файл загружается с FTP-ресурса компании (см. пункт <a href="#">Загрузка дистрибутива системы</a> )
<каталог компонента>\application.properties Например: <пакет установки>\core\admin\application.properties	Конфигурационный файл, содержащий настройки соединения с базой данных (см. подраздел <a href="#">Подготовка баз данных</a> ), уникальный идентификатор приложения, настройки соединения с RabbitMq и прочие опции, многие из которых уникальны для каждого компонента.
<каталог компонента>\schema.properties	<b>Только для компонента router!</b> Конфигурационный файл, содержащий настройки приёмников, отправщиков и схем.
<каталог компонента>\logback.xml	Файл, содержащий настройки конфигурации логирования. По умолчанию он перечитывается каждые 30 секунд, что позволяет менять настройки логирования "на лету".
<каталог компонента>\isimple-msg-<наименование компонента>-<версия файла>.jar Например: <пакет установки>\core\admin\isimple-msg-admin-2.1.0.jar	Исполняемый jar-файл.
<каталог компонента>\start.bat либо <каталог компонента>\start.sh	Командный файл, выполняющий запуск компонента. В нём заранее прописаны ссылки на application.properties, logback.xml, каталог для записи логов, а для компонентов admin и router - ещё и на файл лицензии.
<каталог компонента>\isimple-msg-<наименование компонента>.xml	Конфигурационный файл для запуска компонента в режиме службы для ОС Windows
<каталог компонента>\README.md	Файл с описанием компонента, описанием базовых конфигураций и инструкцией по запуску компонента в режиме службы

## 3.2 Подготовка баз данных

1. [Перечень баз данных:](#)
  - [Обязательные базы данных:](#)
  - [Опционально создаются базы данных:](#)
2. [Настройка параметров подключения к базе данных](#)

### 3.2.1 Перечень баз данных:

В процессе установки системы создаётся несколько баз данных.

#### 3.2.1.1 Обязательные базы данных:

- **Адреса** - БД клиентских адресов `isimple-msg-address`.
- **Сообщения** - БД сообщений и событий по ним `isimple-msg`.
- **АРМ администратора** - БД настройки безопасности `isimple-msg-security`.

#### 3.2.1.2 Опционально создаются базы данных:

- **База данных перенесённых номеров** - БД для сервиса базы данных перенесённых номеров `isimple-messaging-bdprn`. Необходимо создавать, если планируется использовать сервис.
- **Архивные сообщения** - БД архивных сообщений `isimple-messaging-archive`. Необходимо создавать, если будет использоваться архиватор сообщений.
- **БД эмулятора SMPP-сервера** - `smpp-server`. Необходимо создавать, если для тестирования планируется использовать [эмулятор SMPP-сервера](#) вместо настоящего

SMPP-провайдера.  Только СУБД PostgreSQL!

Типы поддерживаемых СУБД и конфигурацию оборудования см. в разделах [Требования к сопутствующему программному и аппаратному обеспечению](#) и [Конфигурация оборудования](#).

Процедура подготовки в соответствии с выбранным типом базы данных приведена в следующих разделах:

- [Подготовка Microsoft SQL Server](#)
- [Подготовка Oracle Database](#)
- [Подготовка PostgreSQL](#)

Помимо общих мер подготовки базы данных, приведённых в разделах выше, для СУБД PostgreSQL и Oracle необходимо проверить, что максимальное число одновременных подключений к серверу БД не менее 300.

Для первоначального создания метаданных и наполнения данными созданных БД используется инструмент миграции [Flyway](#). SQL-скрипты располагаются в дистрибутиве **iSimpleMessageGate** по пути `classpath:db/migration/[подсистема]/{vendor}/`. Выполнение скриптов происходит автоматически при запуске [компонентов](#) системы **iSimpleMessageGate** (см. подраздел [Запуск системы](#)).



**Внимание!** Категорически запрещено менять скрипты, примененные при развертывании рабочей версии системы. **Flyway** контролирует контрольную сумму каждого скрипта и выдаст ошибку, если пролитый ранее скрипт окажется изменен.

## 3.2.2 Настройка параметров подключения к базе данных

Настройки по умолчанию хранятся в файле `application.properties` (см. подраздел [Настройка конфигурационного файла](#)).

Для настройки соединения с СУБД используйте параметры приведенные ниже. Примеры приведены для **PostgreSQL**.

### БД адресов:

```
client.datasource.driverClassName=org.postgresql.Driver
client.datasource.url=jdbc:postgresql://localhost:5432/isimple-messaging-
client
client.datasource.username=postgres
client.datasource.password=system
```

### БД сообщений:

```
spring.datasource.driverClassName=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/isimple-messaging
spring.datasource.username=postgres
spring.datasource.password=system
```

### БД АРМ администратора:

```
security.datasource.driverClassName=org.postgresql.Driver
security.datasource.url=jdbc:postgresql://localhost:5432/isimple-
messaging-security
security.datasource.username=postgres
security.datasource.password=system
```

### Динамическая передача параметров при запуске (опционально)

Каждый параметр может быть изменен при запуске каждого из сервисов передачей одноименного параметра в переменные окружения JAVA

```
java -Dfile.encoding=utf-8 -
Dspring.datasource.driverClassName=org.postgresql.Driver -
Dspring.datasource.url=jdbc:postgresql://localhost:5432/isimple-messaging
-Dspring.datasource.username=postgres -Dspring.datasource.password=system
-jar isimple-msg-admin-x.y.0.jar
```

Именно такой способ передачи параметров подключения к базе данных рекомендуется для большинства сервисов `iSimpleMessageGate`.

Подробности будут рассмотрены в разделе [Настройка компонентов системы](#).

## 3.3 Установка сервера очередей RabbitMQ

- [Настройка для Windows](#)
- [Настройка для Linux/Unix](#)
- [Настройки для удобства администрирования](#)
- [Кластеризация сервера очередей RabbitMQ](#)

Загрузить дистрибутив сервера очередей **RabbitMQ** можно на официальном портале: <https://www.rabbitmq.com/download.html>

Для установки воспользуйтесь руководством поставщика, предназначенным для вашей операционной системы: <https://www.rabbitmq.com/download.html#installation-guides>

### 3.3.1 Настройка для Windows

После установки сервера очередей добавляем службы просмотра очередей для Windows:

```
# Перемещаемся в <каталог, куда установлен RabbitMQ>\sbin :
cd ..\rabbitmq_server-3.7.7\sbin
# Выполняем команды:
# активируем консоль управления RabbitMQ:
rabbitmq-plugins enable rabbitmq_management
# Пользователь guest не сможет авторизоваться с удалённого ПК, поэтому
создаём пользователя и наделяем его правами администратора (заменяв
<login> на желаемый логин, а <pass> - на желаемый пароль):
rabbitmqctl add_user <login> <pass>
rabbitmqctl set_user_tags <login> administrator
rabbitmqctl set_permissions -p / login ".*" ".*" ".*"
```

В целях безопасности рекомендуется после создания пользователя с правами администратора заблокировать пользователя guest/guest.

Это можно осуществить, воспользовавшись руководством поставщика, при помощи консоли управления RabbitMQ, доступной по адресу ip-address:15672.

### 3.3.2 Настройка для Linux/Unix

Установив сервер очередей, активируем консоль управления RabbitMQ:

```
rabbitmq-plugins enable rabbitmq_management
chown -R rabbitmq:rabbitmq /var/lib/rabbitmq
```

Благодаря этому консоль управления RabbitMQ становится (после запуска) доступной по адресу: ip-address:15672.

Затем запускаем службу сервера очередей:

```
systemctl start rabbitmq-server
```

Пользователь `guest` не сможет авторизоваться с удалённого ПК, поэтому при помощи команд Linux создаём пользователя и наделяем его правами администратора (заменяя `<login>` на желаемый логин, а `<pass>` - на желаемый пароль):

```
rabbitmqctl add_user <login> <pass>
rabbitmqctl set_user_tags <login> administrator
rabbitmqctl set_permissions -p / <login> ".*" ".*" ".*"
```

В заключение в целях безопасности необходимо заблокировать пользователя `guest`.

### 3.3.3 Настройки для удобства администрирования

В ОС Linux конфигурационный файл, как правило, расположен по пути `/etc/rabbitmq/rabbitmq.conf`, в противном случае найти его можно с помощью [руководства](#).

RabbitMQ по умолчанию производит ротацию лога раз в 7 дней. За это время размер лог-файла может достичь нескольких сотен мегабайт или даже гигабайт. Для того, чтобы [включить ротацию](#) лог-файла при достижении определённого размера, необходимо в конфигурационном файле раскомментировать соответствующую настройку и указать максимальный размер файла:

```
# rotate when the file reaches 10 MiB
log.file.rotation.size = 10485760
```

### 3.3.4 Кластеризация сервера очередей RabbitMQ

- [Предисловие](#)
- [Настройка кластеризации](#)
  - [Установка независимых сервисов RabbitMQ на несколько узлов.](#)
  - [Разрешение имени хоста](#)
  - [Конфигурирование `erlang.cookie`](#)
  - [Настройка репликации очередей](#)
  - [Запуск кластера](#)
- [Особенности конфигурирования компонентов `iSimpleMessageGate` для работы с кластером RabbitMQ](#)
- [Особенности конфигурирования системы дистанционного банковского обслуживания `iSimpleBank 2.0` для работы с кластером RabbitMQ](#)

#### 3.3.4.1 Предисловие

Официальная документация по кластеризации сервера очередей RabbitMQ представлена на портале поставщика: <https://www.rabbitmq.com/clustering.html>

Текущая статья является кратким изложением некоторых моментов, связанных с кластеризацией продукта.

Особенность кластера RabbitMQ заключается в том, что отсутствует понятие глобального мастера. Роль мастера отводится какому-либо серверу только в рамках одного события в очереди сообщений.



RabbitMQ может быть кластеризован несколькими способами, начиная от независимой установки нескольких узлов и объединения их через конфигурационный файл, заканчивая использованием специально предназначенных плагинов или развертыванием кластера с помощью Kubernetes. Подробнее с этими способами можно ознакомиться в официальной документации поставщика RabbitMQ.

В данной статье мы рассмотрим только один способ - обнаружение на основе DNS.

### 3.3.4.2 Настройка кластеризации

#### 3.3.4.2.1 Установка независимых сервисов RabbitMQ на несколько узлов.

Устанавливаем RabbitMQ на два или более узла в соответствии с руководством поставщика <https://www.rabbitmq.com/download.html> и последующая [настройка](#).

#### 3.3.4.2.2 Разрешение имени хоста

Узлы RabbitMQ обращаются друг к другу, используя доменные имена, короткие или полные (FQDN). Поэтому имена узлов всех элементов кластера должны быть разрешаемыми со всех узлов кластера, а также с компьютеров, на которых могут использоваться такие инструменты командной строки, как `rabbitmqctl`. Разрешение имени хоста может использовать любой из стандартных методов, предоставляемых ОС:

- DNS записи
- Локальные файлы хоста (например, `/etc/hosts`)

#### 3.3.4.2.3 Конфигурирование `erlang.cookie`

Чтобы узлы кластера могли пройти аутентификацию на первой ноде, значение Erlang cookie должно быть одинаковым на всех узлах. Например, в ОС Linux его можно получить из следующего файла:

```
/var/lib/rabbitmq/.erlang.cookie
```

Этот файл необходимо привести к одинаковому значению на каждом узле кластера. Он должен принадлежать пользователю `rabbitmq` и иметь права доступа 600.

После изменения файла необходим перезапуск демона `rabbitmq`.

#### 3.3.4.2.4 Настройка репликации очередей

Все данные / состояния, необходимые для работы сервера очередей RabbitMQ, реплицируются на все узлы. Исключением являются очереди сообщений, которые по умолчанию находятся на одном узле, хотя они видимы и доступны для всех узлов. Это означает, что остановка первого в кластере узла приведёт к недоступности очередей на остальных узлах.

Для того, чтобы реплицировать очереди, необходимо создать политику, которая будет описывать режим и тип репликации. Политики могут создаваться в любое время. Политики могут применяться ко всем очередям или к выборочным очередям (с фильтрацией имени очереди по шаблону регулярного выражения).

Можно создавать нереплицируемые очереди, а позже делать их реплицируемыми через создание политики.

Настроим политику по синхронизации очередей сообщений между нодами.

Необходимо определить, какой узел будет считаться первым в кластере (не мастером, а просто первым в кластере) и запомнить его имя хоста. Далее в примере - Node1

На первом узле кластера выполняем команду

```
rabbitmqctl set_policy ha-all "" '{"ha-mode":"all","ha-sync-mode":"automatic"}'
```

где ha-all – имя политики(может быть произвольным)

"" – нет фильтрации по имени / паттерну очереди, на которую необходимо применить эту политику, т.е. политика будет применяться на все очереди. Здесь можно указывать шаблоны по имени очереди, для которых нужно применять политику.

"ha-mode":"all" – режим репликации очередей all – очереди реплицируются на все ноды в кластере

"ha-sync-mode":"automatic" - тип репликации очередей – очереди будут автоматически синхронизированы при добавлении новой ноды

В выводе получим:

```
Setting policy "ha-all" for pattern "" to '{"ha-mode":"all","ha-sync-mode":"automatic"}' with priority "0" for vhost "/" ...
```

Проверка действующих политик:

```
rabbitmqctl list_policies
```

В выводе получим:

```
Listing policies for vhost "/" ...
vhost  name    pattern apply-to    definition    priority
/      ha-all  all      {"ha-mode":"all","ha-sync-
mode":"automatic"}    0
```

#### 3.3.4.2.5 Запуск кластера

Первый узел кластера должен быть запущен.

На втором и остальных узлах кластера следует выполнить следующие команды:

```
rabbitmqctl stop_app
rabbitmqctl reset
rabbitmqctl join_cluster rabbit@Node1
rabbitmqctl start_app
```

После этого вывод `rabbitmqctl cluster_status` и WEB-интерфейс в разделе Overview на любом из серверов кластера будут содержать имена всех серверов:

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@AMDSupport	46 32768 available	3 29401 available	1184 1048576 available	140MB 12GiB high watermark 48MB low watermark	157GiB 48MB low watermark	14m 43s	basic disc 1 rss	This node All nodes	
rabbit@CentOS	46 32768 available	3 29401 available	775 1048576 available	98MB 19GiB high watermark 48MB low watermark	6.5GiB 48MB low watermark	13m 9s	basic disc 4 rss	This node All nodes	

### 3.3.4.3 Особенности конфигурирования компонентов iSimpleMessageGate для работы с кластером RabbitMQ

При стандартном развёртывании системы iSMG подключение к RabbitMQ задаётся через конфигурационный файл `application.properties` следующим образом:

```
spring.rabbitmq.dynamic=true
spring.rabbitmq.host = RabbitMQHost
spring.rabbitmq.port = 5672
spring.rabbitmq.virtual-host=/
spring.rabbitmq.username = guest
spring.rabbitmq.password = guest
```

В случае, когда RabbitMQ запущен в виде кластера из нескольких экземпляров, необходимо в настройках подключения каждого компонента к серверу очередей указать адреса всех экземпляров. Делается это следующим образом:

```
spring.rabbitmq.dynamic=true
spring.rabbitmq.addresses=RabbitMQHost1:5672,RabbitMQHost2:5672,...,RabbitMQHostN:5672
spring.rabbitmq.virtual-host=/
spring.rabbitmq.username = guest
spring.rabbitmq.password = guest
```

### 3.3.4.4 Особенности конфигурирования системы дистанционного банковского обслуживания iSimpleBank 2.0 для работы с кластером RabbitMQ

При стандартной настройке системы ДБО iSimpleBank 2.0 для интеграции с RabbitMQ адрес указывается в полях **Хост** и **Порт**.

В случае, когда RabbitMQ запущен в виде кластера из нескольких экземпляров, необходимо оставить поля **Хост** и **Порт** незаполненными, а адреса всех экземпляров сервера очередей перечислить в поле **Список адресов RabbitMQ**.

Описание перечисленных полей см. в [соответствующем разделе документации](#).

## 4 Установка системы iSimpleMessageGate

### 4.1 Основные компоненты системы

- [Администрирование – iSimpleMsgAdmin \(обязательный компонент\)](#)
- [Диспетчер сообщений – iSimpleMsgRouter \(обязательный компонент\)](#)
- [Сервис приема сообщений \(опциональный компонент\)](#)
  - [SMPP-приёмник](#)
  - [REST-приёмник](#)
  - [Файловый приёмник](#)
  - [RUL приёмник](#)
  - [Smart reporter](#)
- [Сервис отправки сообщений \(выбираем опционально все возможные отправщики или часть\)](#)
  - [SMPP-отправщик](#)
  - [Google cloud \(Firebase Cloud Messaging\) отправщик](#)
  - [Apple cloud \(APNS\) отправщик](#)
  - [Новый Google cloud \(Firebase Cloud Messaging\) отправщик](#)
- [База данных перенесённых номеров \(опционально\)](#)
- [Сервис проверки доступности сети Интернет на серверах отправщиков \(опционально\)](#)

#### 4.1.1 Администрирование – iSimpleMsgAdmin (обязательный компонент)

См. подраздел [Администрирование iSimpleMsgAdmin](#).

##### Решаемые задачи:

- Обновление трёх БД при запуске приложения (при наличии новых скриптов миграции).
- Защищенный вход в систему.
- Управление пользователями / ролями / процессом аутентификации.
- Мониторинг входа / выхода.
- Поиск и просмотр сообщений и их событий (включая статусы, референсы, ошибки).
- Поиск / просмотр / блокировка адресов клиентов / мониторинг.
- Отчеты.

**Встроенные средства мониторинга / управления:** Health check, loggers, heapdump, threaddump, metrics и другие.

**Используемые БД:**

- БД сообщений и событий;
- БД адресов и истории их изменений;
- БД настроек;
- БД перенесённых номеров (опционально).

#### 4.1.2 Диспетчер сообщений – iSimpleMsgRouter (обязательный компонент)

См. подраздел [Диспетчер сообщений](#).

##### Решаемые задачи:

##### Работа с адресами

- Прием событий об изменении адресов клиента и обновление адресов в БД.
- Предоставление набора незаблокированных адресов клиента по одному (обычно SMS) адресу.

##### Работа с сообщениями

- Прием сообщений из MQ от приемников сообщений.
- Поддержка приоритезации сообщений.
- Валидация / коррекция адресов абонента (пока реализатор только для SMS-адресов).
- Последовательная передача на отправку согласно описанной схеме, при наличии нужного отправку адреса (у каждого сообщения может быть своя схема).
- Поддержка срока действия сообщения (атрибутом сообщения или настройкой схемы).
- Поддержка протекания сообщения в очереди на отправку (использование средств MQ).
- Управление таймаутом ожидания квитанции об успешной отправке (задается в схеме).
- Управление таймаутом ожидания квитанции об успешной / неуспешной обработки (задается в схеме).
- Отслеживание статуса работоспособности сервисов отправки сообщений по healthcheck и включение/отключение их в схемах.
- Сохранение всех событий по сообщению в БД.
- Наличие view в БД для просмотра статусов сообщений.
- Уведомление отправителя сообщения посредством MQ об обработке сообщения (ошибка / не доставлен / доставлен / прочитан).

**Встроенные средства мониторинга / управления:** Health check, loggers, heapdump, threaddump, metrics и другие.

##### Используемые БД:

- БД сообщений и событий.
- БД адресов и истории их изменений;
- БД перенесённых номеров (опционально).

## 4.1.3 Сервис приема сообщений (опциональный компонент)

### 4.1.3.1 SMPP-приёмник

(см. подраздел [Сервис приема SMPP-сообщений](#))

#### Решаемые задачи:

- приём подключения по протоколу (с проверкой типа системы / имени системы / пароля / двойного подключения);
- прием сообщений по протоколу SMPP;
- передача ответных квитков по протоколу SMPP;
- поддержка типов сообщений SAR / UDH-1, UDH-2 (с учетом разбиения на сегменты);
- поддержка кодировок сообщений UTF-8 / UTF-16;
- поддержка ограничения входящих сообщений в секунду (0 - без ограничений);
- поддержка отложенной доставки уведомлений отправителю (при разрыве сессии);
- отправка сообщений для доставки клиенту в MQ в диспетчер сообщений;
- поддержка приоритизации сообщений;

**Встроенные средства мониторинга / управления:** Health check, loggers, heapdump, threaddump, metrics и другие;

**Используемые БД:** БД сообщений и событий.

### 4.1.3.2 REST-приёмник

(см. подраздел [Сервис приема сообщений по протоколу REST API](#))

#### Решаемые задачи:

- приём сообщений по протоколу REST API (см. подраздел [Описание API внешнего обмена данными с сервисом приёма сообщений по протоколу REST API](#));
- получение статуса сообщения по REST API;
- получение статуса набора сообщений по REST API;
- обновление адресов клиента по REST API;
- отправка сообщений для доставки клиенту в MQ в диспетчер сообщений;
- поддержка лимита входящих сообщений за час / день / месяц (0 - без ограничений);
- поддержка приоритизации сообщений.

**Встроенные средства мониторинга / управления:** Health check, loggers, heapdump, threaddump, metrics и другие;

**Используемые БД:** Не используются

### 4.1.3.3 Файловый приёмник

(см. подраздел [Приёмник сообщений по файловым форматам](#))

#### Решаемые задачи:

- приём авторизованного подключения операциониста через web-интерфейс сервиса;
- ручной импорт файла сообщений операционистом через web-интерфейс сервиса;
- автоматический приём сообщений из файла при его добавлении в целевой каталог;
- отправка сообщений для доставки клиенту в MQ в диспетчер сообщений;
- поддержка приоритизации сообщений;

- изменение формата импорта через конфигурационный файл.

**Встроенные средства мониторинга / управления:** Heath check, loggers, heapdump, threaddump, metrics и другие;

**Используемые БД:** Не используются

#### 4.1.3.4 RUL приёмник

(см. подраздел [Сервис приёма сообщений из файла по RUL-форматам](#))

##### Решаемые задачи:

- приём авторизованного подключения операциониста через web-интерфейс сервиса;
- ручной импорт файла сообщений операционистом через web-интерфейс сервиса;
- автоматический приём сообщений из файла при его добавлении в целевой каталог;
- отправка сообщений для доставки клиенту в MQ в диспетчер сообщений;
- поддержка приоритизации сообщений.

**Встроенные средства мониторинга / управления:** Heath check, loggers, heapdump, threaddump, metrics и другие;

**Используемые БД:** Не используются

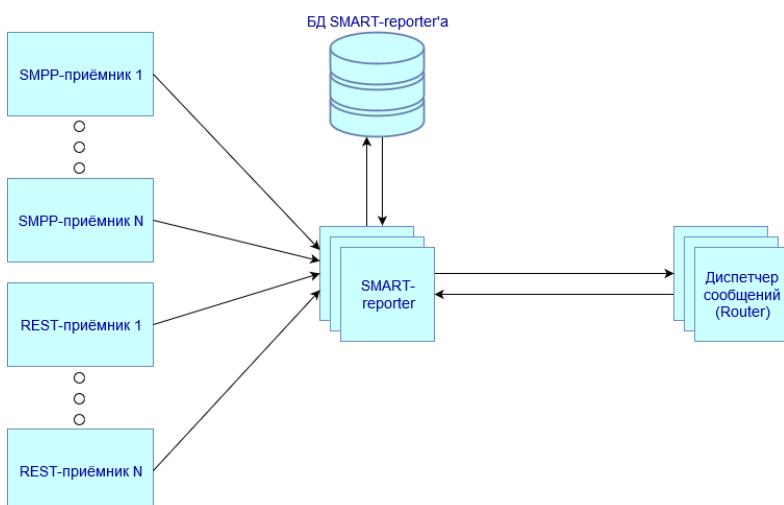
#### 4.1.3.5 Smart reporter

(см. подраздел [Smart reporter. Сервис приостановления сообщений по правилам и генерации сводных отчетов рассылки](#))

##### Решаемые задачи:

- приём сообщений;
- немедленная отправка сообщений, не удовлетворяющих правилам, в router;
- немедленная отправка PUSH-сообщений по приостановленным сообщениям для адресов, у которых есть PUSH\_ID;
- приостановка обработки сообщений, удовлетворяющих правилам и содержащих сумму и продукт;
- периодическая и событийная генерация сводных отчётов по сообщениям.

Таким образом, SMART-reporter принимает сообщения от приёмников, исполняя роль посредника между приёмниками и диспетчером сообщений (Router'ом).



Сервис не обязательный и не включается в общую поставку. При необходимости сервис может быть включен в поставку по запросу от банка после согласования с коммерческой дирекцией.

**Встроенные средства мониторинга / управления:** Heath check, loggers, heapdump, threaddump, metrics и другие;

**Используемые БД:**

- БД сообщений и событий;
- БД адресов и истории их изменений;
- БД настроек.

#### 4.1.4 Сервис отправки сообщений (выбираем опционально все возможные отправщики или часть)

##### 4.1.4.1 SMPP-отправщик

(см. подраздел [Сервис отправки сообщений по протоколу SMPP](#)).

**Решаемые задачи:**

- Поддержка настроек соединения, включая резервные адреса SMPP-сервера.
- Поддержание сессии с SMPP-сервером в активном состоянии.
- Прием сообщения посредством MQ от диспетчера сообщений (за исключением превысивших время по таймауту).
- Автоматический отказ сообщений, если сессии с SMPP-сервером нет.
- Поддержка типов сообщений SAR / UDH-1, UDH-2 (с учетом разбиения на сегменты).
- Поддержка кодировок сообщений UTF-8 / UTF-16.
- Прием квитков о доставке / недоставке / сборка квитков для сообщений, разбитых на сегменты.

**Встроенные средства мониторинга / управления:** Heath check, loggers, heapdump, threaddump, metrics и другие;

**Используемые БД:** БД сообщений и событий.

##### 4.1.4.2 Google cloud (Firebase Cloud Messaging) отправщик

(см. подраздел [Сервис отправки сообщений по протоколу FCM \(Firebase\)](#))

Данный сервис отправки сообщений используется в целях совместимости с текущими мобильными приложениями iSimpleMobile для iOS и Android.

Отправка PUSH-уведомлений по протоколу Firebase legacy HTTP protocol.

##### 4.1.4.3 Apple cloud (APNS) отправщик

(см. подраздел [Сервис отправки сообщений по протоколу APNs \(Apple Push Notification Service\)](#))

Данный сервис отправки сообщений будет использоваться в обновленных мобильных приложениях iSimpleMobile для iOS. Отправка сообщений по протоколу APNS.

##### 4.1.4.4 Новый Google cloud (Firebase Cloud Messaging) отправщик

(см. подраздел [Сервис отправки сообщений по протоколу FCM \(Firebase\)](#))



Данный сервис отправки сообщений будет использоваться в обновленных мобильных приложениях iSimpleMobile для Android.

**Решаемые задачи:**

- Отправка сообщений через Firebase Admin SDK.
- Интеграция с интернет-банком
- Отправка сообщений в MQ / получение статусов из MQ.
- Отправка уведомлений о доставке PUSH-уведомления в мобильный телефон: доставка / прочтение.
- Разовая отправка адресов всех клиентов.
- Отправка адресов клиента при изменении

#### 4.1.5 База данных перенесённых номеров (опционально)

См. подраздел [Сервис базы данных перенесенных номеров](#).

**Решаемые задачи:**

- обновления публичных данных выделенных номеров;
- обновления базы данных перенесенных номеров;
- предоставление поиска номера телефона или набора номеров телефонов по REST API;

**Встроенные средства мониторинга / управления:** Health check, loggers, heapdump, threaddump, metrics и другие.

**Используемые БД:** БД перенесённых номеров

#### 4.1.6 Сервис проверки доступности сети Интернет на серверах отправщиков (опционально)

См. подраздел [Сервис проверки доступности сети Интернет \(check\\_network\) на серверах отправщиков](#).

**Решаемые задачи:**

- проверка доступности сети Интернет в узлах системы, занимающихся отправкой сообщений во внешние системы (SMPP-провайдеру, в PUSH-облако или на сервер SMTP);
- передача сигнала диспетчеру сообщений для отключения узлов, на которых недоступна сеть Интернет;
- передача сигнала диспетчеру сообщений для включения узлов, на которых вновь доступна сеть Интернет;

**Встроенные средства мониторинга / управления:** Health check, loggers, heapdump, threaddump, metrics и другие.

**Используемые БД:** не используется