

Общество с ограниченной ответственностью

«Транспортные информационные системы» ООО «ТИС»

## Развертывание программного обеспечения

"Телематическая платформа «TIS-Online»"

на ОС Astra Linux SE 1.6 "Смоленск"

### г. Москва - 2021

**Содержание**

1. [Установка и настройка БД PostgreSQL 9.6](#_bookmark0) [1](#_bookmark0)
2. [Создание БД, роли и схемы телематических сервисов](#_bookmark1) [2](#_bookmark1)
3. [Установка исполнения Java JRE](#_bookmark2) [3](#_bookmark2)
4. [Создание системной учетной записи телематических сервисов](#_bookmark3) [4](#_bookmark3)
5. [Установка сервисов приема и обработки данных](#_bookmark4) [5](#_bookmark4)
6. [Установка сервиса выгрузки данных по подпискам](#_bookmark5) [10](#_bookmark5)
7. [Сервис импорта данных в приложение мониторинга автотранспорта](#_bookmark6) [13](#_bookmark6)
8. [Создание подписок](#_bookmark7) [19](#_bookmark7)
9. [Приложение для очистки устаревших данных](#_bookmark8) [21](#_bookmark8)

# Установка и настройка БД PostgreSQL 9.6

## Установка

Установочные пакеты СУБД входят в репозиторий Astra Linux SE. Администратору системы необходимо выполнить установку используя штатный менеджер пакетов, включая требуемые зависимости:

sudo apt-get install postgresql-9.6 postgresql-client-9.6

В процессе установки будет автоматически создана служебная БД **postgres** и учетная запись администратора **postgres**.

Для корректной работы аутентификации пользователей СУБД в конфигурационном

🛈 файле /etc/parsec/mswitch.conf следует изменить значение параметра

#### zero\_if\_notfound в yes.

## Настройка

Конфигурационные файлы СУБД расположены в каталоге /etc/postgresql/9.6/main/.

Для запуска телематических сервисов необходимо указать учётную запись (УЗ) для подключения сервисов к БД в файле pg\_hba.conf и настроить параметры инстанса БД в файле postgresql.conf.

В файл pg\_hba.conf следует добавить запись:

host tmsdb tms 10.10.10.20/32 md5

где:

* **10.10.10.20/32** - ip адрес сервера телематической платформы;
* **tmsdb** - база данных телематических сервисов;
* **tms** - учетная запись для подключения телематических сервисов.

В файле postgresql.conf необходимо настроить параметры:

* **listen\_addresses** - ip адрес сервера БД телематики;
* **max\_connections** - указать из расчета не менее 200 соединений на один телематический сервис.

Также необходимо указать параметры использования памяти и рабочих процессов, исходя из конфигурации сервера для профиля нагрузки OLTP, для продуктивных систем переключить параметр **wal\_level** в значение **logical** и включить архивирование журналов WAL.

# Создание БД, роли и схемы телематических сервисов

Необходимо выполнить следующие шаги:

1. Подключиться к БД с правами администратора с помощью команды

sudo -u postgres psql

#### Дальнейшие действия выполняются в консоли администратора СУБД

1. Создать учетную запись в БД для телематических сервисов:

**CREATE USER** "tms" **WITH** LOGIN NOSUPERUSER **NOCREATEDB** NOCREATEROLE INHERIT NOREPLICATION **CONNECTION LIMIT** -**1 ENCRYPTED** PASSWORD

'secure\_password';

1. Создать телематическую базу

**CREATE DATABASE** tmsdb **OWNER** = "tms" **TEMPLATE** = template0 **ENCODING** = 'UTF-8' LC\_COLLATE = 'ru\_RU.UTF-8' LC\_CTYPE = 'ru\_RU.UTF-8' **CONNECTION LIMIT** = -**1**;

1. Создать схему для хранения телематических данных, последовательно выполнив команды:

\**connect** tmsdb;

**CREATE SCHEMA** "tms" **AUTHORIZATION** tms;

Необходимые для работы структуры данных создаются автоматически при первом запуске телематических сервисов.

🛈

# 3. Установка исполнения Java JRE

В состав дистрибутива ОС среду исполнения Java JRE не входит, поэтому необходимо использовать стороннее ПО.

В данной инструкции описывается установка среды исполнения **GosJava SE**.

Необходимо выполнить следующие шаги:

1. Установить носитель с дистрибутивом в CD привод системы, выполнить команду добавления источника в список доступных репозиториев:

sudo apt-cdrom -d=/media/cdrom add

1. Добавить ключ репозитория в систему командой

sudo apt-key add /media/cdrom/lab50.gpg

1. Обновить список доступных для установки пакетов с помощью команды

sudo apt-get update

1. Установить среду исполнения GosJava SE командой

sudo apt-get install gosjava-jre gosjava-jre-headless

🛈 Все зависимые пакеты будут установлены автоматически.

1. Проверить корректность установки с помощью команды

java -version

Результатом выполнения команды должна быть информации о версии установленной среды исполнения Java.

# Создание системной учетной записи телематических сервисов

Телематические сервисы запускаются с правами непривилегированной учетной записи ОС. На данном этапе необходимо создать группу **tms** и учетную запись **tms**, используя команды:

sudo addgroup --system tms

sudo adduser --system --home /home/tms --ingroup tms tms

где:

* ключ **--home** указывает расположение домашнего каталога УЗ;
* ключ **--ingroup** назначает основную группу УЗ.

🛈 Домашний каталог УЗ будет создан автоматически в момент создания УЗ.

# Установка сервисов приема и обработки данных

В типовой конфигурации на сервере телематической платформы разворачивается несколько сервисов приема и обработки данных.

Для удобства сопровождения и администрирования системы имена сервисов должны соответствовать шаблону **telesrvNNN**, где NNN последние 3 цифры порта, назначенного сервису.

В качестве примера будут развернуты два сервиса приема и обработки данных на портах 58011 и

58014 соответственно. Для этого следует выполнить несколько шагов:

1. В домашнем каталоге УЗ tms необходимо создать соответствующие каталоги командой

sudo mkdir ~tms/telesrv011 ~tms/telesrv014

1. Развернуть архив с дистрибутивом сервиса приема и обработки данных в созданные выше каталоги.
2. Создать каталоги конфигурационных файлов командой

sudo mkdir ~tms/telesrv011/config ~tms/telesrv014/config

Предусмотрены следующие конфигурационные файлы:

* communication.json - параметры подключения к БД и сетевые параметры сервиса;
* ds\_pool.json - параметры пула соединений с БД;
* logback.xml - конфигурация журналирования;
* mappingAT.json - таблица перезаписей id терминалов БНСО.

1. Установить необходимые права доступа командой

sudo chown -R tms.tms ~tms/

## Примеры конфигурационных файлов

communication.json

{

"versionConfig" : "1.0",

"uid" : "TIS",

"services" : [ {

"uid" : "jdbc/main\_ds",

"url" : "jdbc:postgresql://172.30.48.195:5432/tmsdb?stringtype=unspecified", "user" : "tms",

"password" : "ZXtZe3s/0Vm2B5Mn+1D6nw==", "encrypt" : **true**,

"props" : {

"dataSourceClassName" : "org.postgresql.ds.PGSimpleDataSource", "socketTimeout" : **120**

}

}, {

"uid" : "TELEMATIC\_SERVER", "url" : "58011",

"user" : "gpn\_trans",

"password" : "UsGjRz0vabi688FHppba0mA==", "encrypt" : **true**,

"props" : {

"port" : **58011**, "protokol": "EGTS",

"bufferDir" : "/home/tms/np-telesrv011/buffJson", "cntThreadBuff":**10**,

"maxPoolRead": **100**,

"ftpDir" : "/srv/ftp/vftpusers/ScoutFTP/in", "photoDir" : "/home/tms/np-telesrv011/photoStore"

}

} ]

}

ds\_pool.json

[ {

"version\_config" : **3**

}, {

"id" : "jdbc/main\_pool",

"dataSourceClassName" : "com.zaxxer.hikari.HikariDataSource", "props" : {

"poolName" : "main\_pool", "dataSource" : "jdbc/main\_ds", "maximumPoolSize" : **100**,

"connectionTimeout" : **5000**,

"idleTimeout" : **60000**,

"keepaliveTime" : **20000**,

"maxLifetime" : **1800000**,

"minimumIdle" : **50**

}

} ]

mappingAT.json

{ "861694034442747":"2051763",

"356306058698136":"408397",

"862631032530999":"2051768",

"861694034510915":"2051764"

}

1. Далее необходимо создать unit файлы для управления сервисами средствами ОС.

Для этого следует создать файл /lib/systemd/system/np-telesrv.service со следующим содержимым

[Unit]

Description=Telematic Server

[Service] Type=simple ExecStart=/bin/true ExecReload=/bin/true RemainAfterExit=on

[Install]

WantedBy=multi-user.target

а также создать файл /lib/systemd/system/np-telesrv@.service c содержимым:

[Unit]

Documentation=man:systemd-sysv-generator(8) Description=Selector (Telematic) Server Before=multi-user.target

Before=graphical.target After=remote-fs.target After=network-online.target Wants=network-online.target

[Service] User=tms Group=tms

Environment=PATH=/home/tms/telesrv%i:/usr/lib/jvm/default-java/bin:/usr/local/bin:/usr/bin:/bin Environment=JAVA\_HOME=/usr/lib/jvm/default-java

UMask=002 WorkingDirectory=/home/tms/telesrv%i Type=simple

KillMode=process GuessMainPid=yes RemainAfterExit=no

ExecStart=/usr/bin/java -XX:+AggressiveOpts -Xms4096m -Xmx8192m -jar ThreadServer.jar

/home/tms/telesrv%i/config SuccessExitStatus=143 TimeoutStopSec=10 Restart=on-failure RestartSec=10 LimitNOFILE=65000 LimitNPROC=65000

TasksMax=30000

[Install]

WantedBy=multi-user.target

где значения параметров **Xms** и **Xmx** в строке **ExecStart** должны быть скорректированы в соответствии с конфигурацией памяти телематического сервера.

Параметр **SuccessExitStatus=143** необходим для корректной обработки **systemd** кода завершения

java машины.

1. Необходимо разрешить автоматический запуск сервисов командами

sudo systemctl enable telesrv@011 sudo systemctl enable telesrv@014

1. Запустить сервисы

Для систем с большим количеством развернутых сервисов приема данных БНСО для управления удобнее всего использовать циклы, например:

**for i in 011 014**; \

**do** \

echo -n “$i: ” \

**sudo systemctl** start **telesrv**@$i; \

**done**;

Аналогичным образом можно выполнять другие массовые операции над телематическими сервисами, например, для проверки состояния сервисов достаточно в примере выше заменить команду **start** на **is-active**.

# 6. Установка сервиса выгрузки данных по подпискам

Для установки сервиса выгрузки данных по подпискам необходимо выполнять следующие шаги:

1. Создать каталог сервиса и каталог конфигурационных файлов командой

sudo mkdir ~tms/rsserver/ ~tms/rsserver/config/

1. Распаковать дистрибутив в каталог
2. Создать конфигурационные файлы сервиса Перечень конфигурационных файлов:

* communication.json - конфигурация параметров подключения к БД и сетевых настроек сервиса;
* ds\_pool.json - конфигурация пула подключений к БД;
* logback.xml - конфигурация журналирования.

## Примеры конфигурационных файлов

communication.json

{

"versionConfig" : "1.0",

"uid" : "TIS",

"services" : [ {

"uid" : "jdbc/main\_ds",

"url" : "jdbc:postgresql://172.30.48.195:5432/telenavdb2?stringtype=unspecified", "user" : "telenav2",

"password" : "ZXtZe3s/0Vm2B5Mn+1D6nw==", "encrypt" : **true**,

"props" : {

"dataSourceClassName" : "org.postgresql.ds.PGSimpleDataSource", "socketTimeout" : **30**

}

}, {

"uid" : "TELEMATIC\_SERVER", "url" : "http://172.30.48.194:8004/",

"encrypt" : **true**, "props" : {

"authorization":**true**, "ftpDirectory":"/srv/ftp/vftpusers/ScoutFTP/geofences"

}

} ]

}

ds\_pool.json

[ {

"version\_config" : **3**

}, {

"id" : "jdbc/main\_pool",

"dataSourceClassName" : "com.zaxxer.hikari.HikariDataSource", "props" : {

"poolName" : "main\_pool", "dataSource" : "jdbc/main\_ds", "maximumPoolSize" : **50**

}

} ]

1. Далее необходимо создать unit файл сервиса

В файл /lib/systemd/system/np-rsserver.service следует добавить следующий текст

[Unit]

Documentation=man:systemd-sysv-generator(8) Description=Rest Server Telematic

Before=multi-user.target Before=graphical.target After=remote-fs.target After=network-online.target Wants=network-online.target

[Service] User=tms Group=tms

Environment=PATH=/home/tms/rsserver:/usr/lib/jvm/default-java/bin:/usr/local/bin:/usr/bin:/bin Environment=JAVA\_HOME=/usr/lib/jvm/default-java

UMask=002 WorkingDirectory=/home/tms/rsserver Type=simple

KillMode=process GuessMainPid=yes RemainAfterExit=no

ExecStart=/usr/bin/java -jar RestServer.jar /home/tms/rsserver/config SuccessExitStatus=143

TimeoutStopSec=10 Restart=on-failure RestartSec=10

[Install]

WantedBy=multi-user.target

1. Включить автоматический запуск сервиса командой

sudo systemctl enable rsserver

1. Запустить сервис командой

sudo systemctl start rsserver

# Сервис импорта данных в приложение мониторинга автотранспорта

Сервис импорта данных запускается с правами непривилегированной учетной записи ОС. Необходимо выполнить следующие шаги:

1. Создать группу **tsimport** и учетную запись **tsimport** используя две команды:

sudo addgroup --system tsimport

sudo adduser --system --home /home/tsimport --ingroup tsimport tsimport

где

* ключ **--home** указывает расположение домашнего каталога УЗ,
* ключ **--ingroup** назначает основную группу УЗ

🛈 Домашний каталог УЗ будет создан автоматически в момент создания УЗ.

1. Создать рабочий каталог сервиса импорта командой:

sudo mkdir ~tsimport/tsimp

1. Распаковать архив с дистрибутивом сервиса импорта в каталог ~tsimport/tsimp
2. Создать структуру каталогов сервиса.

— каталог конфигурационных файлов:

sudo mkdir ~tsimport/tsimp/etc

— каталог файлов журналов:

sudo mkdir ~tsimport/tsimp/logs

— каталоги дисковых буферов:

sudo mkdir ~tsimport/tsimp/var ~tsimport/tsimp/var/BUFFERS sudo mkdir ~tsimport/tsimp/var/BUFFERS/Rabbit

sudo mkdir ~tsimport/tsimp/var/BUFFERS/TS\_BDD sudo mkdir ~tsimport/tsimp/var/BUFFERS/TS\_DVR\_File

sudo mkdir ~tsimport/tsimp/var/BUFFERS/TS\_DVR\_Photo sudo mkdir ~tsimport/tsimp/var/BUFFERS/TS\_NAV

1. Создать конфигурационные файлы
   * communication.json - конфигурация потоков данных и параметров подключения к БД приложения мониторинга автотранспорта;
   * logback.xml - настройки журналирования;
   * config.properties - конфигурация потоков и буферов данных;
   * ds\_pool.json - конфигурация пула подключения к БД;
   * rmq\_config.json - настройки подключения к брокеру сообщений сервера мобильных клиентов.

🛈 Примеры конфигурационных файлов изложены ниже.

1. Установить необходимые права доступа

sudo chown -R tsimport.tsimport ~tsimport/

## Примеры конфигурационных файлов

communication.json

{

"versionConfig" : "1.0",

"uid" : "Telematic",

"services" : [ {

"uid" : "TELEMATIC\_SERVER", "url" : "http://172.30.48.194:8003",

"user" : "gpn\_trans",

"password" : "UsAjRz0vabi688Hppba0mA==", "encrypt" : **true**,

"props" : **null**

}, {

"uid" : "TS\_NAV",

"url" : "http://172.30.48.194:8003/nav", "user" : "gpn\_trans",

"password" : "UsAjRz0vabi688Hppba0mA==", "encrypt" : **true**,

"props" : **null**

}, {

"uid" : "TS\_BDD",

"url" : "http://172.30.48.194:8003/bdd", "user" : "gpn\_trans",

"password" : "UsAjRz0vabi688Hppba0mA==", "encrypt" : **true**,

"props" : **null**

}, {

"uid" : "TS\_DVR\_File",

"url" : "http://172.30.48.194:8003/dvr", "user" : "gpn\_trans",

"password" : "UsAjRz0vabi688Hppba0mA==", "encrypt" : **true**,

"props" : **null**

}, {

"uid" : "TS\_DVR\_Photo",

"url" : "http://172.30.48.194:8003/dvr",

"user" : "gpn\_trans",

"password" : "UsAjRz0vabi688Hppba0mA==", "encrypt" : **true**,

"props" : **null**

}, {

"uid" : "jdbc/main\_ds",

"url" : "jdbc:postgresql://172.30.48.198:5433/ektpdb?stringtype=unspecified", "user" : "ppdk\_main",

"password" : "QvTSdX04zvLR22Raa+kXOw==", "encrypt" : **true**,

"props" : {

"dataSourceClassName" : "org.postgresql.ds.PGSimpleDataSource"

}

}, {

"uid" : "jdbc/nav\_ds",

"url" : "jdbc:postgresql://172.30.48.198:5433/ektpdb?stringtype=unspecified", "user" : "ppdk\_nav",

"password" : "i4ooVuk4NFskPInSXvhqDQ==", "encrypt" : **true**,

"props" : {

"dataSourceClassName" : "org.postgresql.ds.PGSimpleDataSource"

}

} ]

}

config.properties

# countFileRead - количество одновременно обрабатываемых файлов в потоке

# TimeSleep - период опроса в милисекундах при наличии данных (как быстро "забирать"

данные, минимальное значение: 100)

versionConfig=1 TS\_NAV.Enable=true

TS\_NAV.OutDir=/home/tsimport/tsimp/var/BUFFERS/TS\_NAV TS\_NAV.MaxDirSize=10000

TS\_NAV.TimeSleep=100 TS\_NAV.DbInsert=true TS\_NAV.CountSize=1000 TS\_NAV.countFileRead=4 TS\_NAV.TimeLoadNext=100

TS\_BDD.Enable=true TS\_BDD.OutDir=/home/tsimport/tsimp/var/BUFFERS/TS\_BDD TS\_BDD.MaxDirSize=10000

TS\_BDD.TimeSleep=3000 TS\_BDD.DbInsert=true TS\_BDD.CountSize=100 TS\_BDD.countFileRead=4 TS\_BDD.TimeLoadNext=100

TS\_DVR\_Photo.Enable=true TS\_DVR\_Photo.OutDir=/home/tsimport/tsimp/var/BUFFERS/TS\_DVR\_Photo TS\_DVR\_Photo.MaxDirSize=10000

TS\_DVR\_Photo.TimeSleep=3000 TS\_DVR\_Photo.DbInsert=true TS\_DVR\_Photo.CountSize=100 TS\_DVR\_Photo.countFileRead=4 TS\_DVR\_Photo.TimeLoadNext=100

TS\_DVR\_File.Enable=true TS\_DVR\_File.OutDir=/home/tsimport/tsimp/var/BUFFERS/TS\_DVR\_File TS\_DVR\_File.MaxDirSize=10000

TS\_DVR\_File.TimeSleep=3000 TS\_DVR\_File.DbInsert=true TS\_DVR\_File.CountSize=100 TS\_DVR\_File.countFileRead=4 TS\_DVR\_File.TimeLoadNext=100

#Активирует подключение к службам RabbitMQ Rabbit.Enable=true

#Следующие параметры необходимы только для режима MASTER #Буферная папка для режим Master службы RabbitMQ Rabbit.OutDir=/home/tsimport/tsimp/var/BUFFERS/Rabbit

#Включает загрузку данных из буферной папки в базу данных

Rabbit.DbInsert=true Rabbit.countFileRead=4

#Как часто отправлять телематические данные

Rabbit.TimeSleep=1000

#Сколько данных передавать в одном сообщении. Rabbit.CountSize=300

ds\_pool.json

[ {

"version\_config" : **3**

}, {

"id" : "jdbc/main\_pool",

"dataSourceClassName" : "com.zaxxer.hikari.HikariDataSource", "props" : {

"poolName" : "main\_pool", "dataSource" : "jdbc/main\_ds", "maximumPoolSize" : **20**

}

}, {

"id" : "jdbc/nav\_pool",

"dataSourceClassName" : "com.zaxxer.hikari.HikariDataSource", "props" : {

"poolName" : "nav\_pool", "dataSource" : "jdbc/nav\_ds", "maximumPoolSize" : **20**

}

} ]

rmq\_config.json

{

"enabled" : **false**, "role" : "master",

"msgLogEnabled" : **true**, "versionConfig" : "1.0", "localNode" : {

"appUid" : "EKTP", "appDescr" : "Узел ЕКТП", "host" : "172.30.48.196",

"port" : **5672**,

"tls" : **false**, "vhost" : "/",

"username" : "ektp",

"password" : "L2yxMi4KcOKVq5ke6GYBtQ==", "pwdEncrypted" : **true**,

"services" : {

"rt" : {

"name" : "Транспортные услуги", "role" : ""

},

"gm" : {

"name" : "Обслуживание бортового оборудования", "role" : ""

}

} },

"remoteNodes" : [ {

"appUid" : "",

"appDescr" : "",

"host" : "", "port" : **0**,

"tls" : **false**, "vhost" : "/",

"username" : "",

"password" : "E/LR5REAdQOqD/qjJiiOvg==", "pwdEncrypted" : **true**,

"services" : {

"rt" : {

"name" : "Транспортные услуги", "role" : ""

},

"gm" : {

"name" : "Обслуживание бортового оборудования", "role" : ""

}

}

} ]

}

# Создание подписок

Управление подписками производится из интерфейса приложения мониторнинга автотранспорта.

Для включения функции управления подписками из приложения необходимо

🛈 значение **"uid"** установить в **"Telematic"** в конфигурационном файле

communication.json сервиса импорта данных.

В рамках данного руководства описывается работа с подписками на уровне API сервиса выгрузки данных.

## Добавление подписчика

Создание подписки производится с помощью вызова функции API сервиса выгрузки данных. Для этого необходимо воспользоваться утилитой curl:

curl --location --request POST 'http://172.30.48.194:8003/v1.0/addServer?userName=gpn\_trans' \

--header 'Content-Type: application/json'

После обработки запроса сервер вернет сообщение, содержащие ключ (**userkey**) созданной подписки.

Сообщение имеет следующий примерный вид:

{"result":"ok","message":"Добавлено","userKey":"i25f6OKk9VFV"}

**Имя подписки** и **полученный ключ** необходимо внести в конфигурационный файл

🛈

communication.json сервиса импорта данных (tsimport).

## Добавление подписки

Добавление подписки происходит при установке переключателя **"регистрация"** в приложении мониторинга транспорта с помощью вызова API.

Для этого необходимо воспользоваться утилитой curl:

curl --location --request POST 'http://172.30.48.194:8003/v1.0/createsubscribe' \

--header 'Content-Type: application/json' \

--header 'Authorization: Basic Z3BuX3RyYW5zOmkyNWY2T0trOVZGVgo=' \

--data-raw '["865733024484982"]'

где **URL** берется из настроек в файле communication.json, к нему добавляется **createsubscribe**.

В **header** следует указать тип контекста и Basic авторизация (Basic Auth, используется логин и пароль из файла communication.json).

В теле запроса (Body) передается **id** терминала, по которому нужно создать подписку: **--data-raw**

'["865733024484982"]'

Заголовок **Authorization: Basic** содержит закодированную в base64 строку с именем подписки и

ключом в формате **подписка:ключ**.

Сформировать строку можно используя консольную утилиту base64, например:

echo "gpn\_trans:i25f6OKk9VFV" | base64 Z3BuX3RyYW5zOmkyNWY2T0trOVZGVgo=

🛈 Добавление терминала при создании подписки происходит автоматически.

## Удаление подписки

Удаление подписки на терминал происходит при снятии переключателя **"регистрация"** в приложении мониторинга транспорта с помощью вызова API.

Для этого необходимо воспользоваться утилитой curl:

curl --location --request POST 'http://172.30.48.194:8003/v1.0/removesubscribe' \

--header 'Content-Type: application/json' \

--header 'Authorization: Basic Z3BuX3RyYW5zOmkyNWY2T0trOVZGVgo=' \

--data-raw '["865733024484982"]'

Если вместо номера терминала в теле запроса указать **"ALL"**, то будут удалены все подписки.

🛈

## Проверка подписок

Получение списка подписок предусмотрено с помощью вызова функции API **checkSubscribe**

Для этого необходимо воспользоваться утилитой curl:

curl --location --request POST 'http://172.30.48.194:8003/v1.0/checkSubscribe' \

--header 'Content-Type: application/json' \

--header 'Authorization: Basic Z3BuX3RyYW5zOmkyNWY2T0trOVZGVgo=' \

--data-raw '["865733024484982"]'

Результатом выполнения команды будет список терминалов, для которых существует подписка.

# Приложение для очистки устаревших данных

Для установки приложения необходимо выполнить следующие шаги:

1. Создать каталог

sudo mkdir ~tms/cleandata

1. Распаковать дистрибутив в созданный каталог
2. Создать ссылку на каталог конфигурации rsserver

sudo ln -s /home/tms/rsserver/config config

1. Добавить в /etc/crontab задание на запуск очистки

00 6 \* \* 1 tms java -jar /home/tms/np-cleandata/CleanOldData.jar /home/tms/np- cleandata/config 90

где

* 00 6 \* \* 1 - обозначает запуск задачи в 06:00 утра каждый понедельник,
* tms - УЗ, с правами которой выполняется задание,
* 90 - период хранения телематических данных.