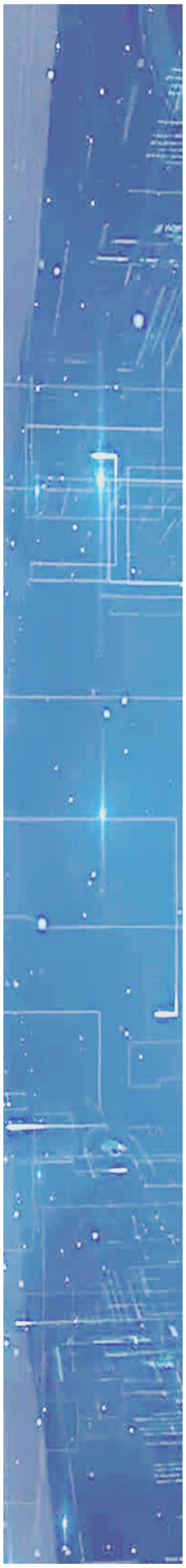
Навигатор BI

Версия 50

14.04.2026



# Руководство администратора

Москва
2026

Навигатор BI: Руководство администратора

М.: ПАО «Сбербанк», 2026

Программное обеспечение и настоящий документ не могут быть скопированы, размножены, использованы по частям для составления других текстов, переведены на другие языки, если это не оговорено в письменной форме в договоре на поставку программного обеспечения.

Программное обеспечение, описанное в настоящем Руководстве, поставляется по лицензионному соглашению и может использоваться или копироваться только в соответствии с условиями этого соглашения.

Разработчиком и генеральным распространителем программного продукта «Навигатор BI» является ПАО «Сбербанк».

Адрес: Россия, Москва, 117997, ул. Вавилова, д. 19

navplatform1@sberbank.ru

https://sberanalytics.ru/products/ksb/navigator

© ПАО «Сбербанк» 2026

# Содержание


**1 Настройка аутентификации пользователей ..... 5**

1.1 Особенности настройки отдельных типов аутентификации: ..... 5

1.1.1 Аутентификация по сертификату ..... 5

1.1.2 Аутентификация по логину\паролю ..... 5

1.1.3 LDAP – аутентификация ..... 6

1.1.4 OpenID - аутентификация ..... 6

**2 Действия внутри АРМ ..... 8**

2.1 Лицензия ..... 8

2.2 Системный журнал ..... 9

2.2.1 Элементы экрана ..... 9

2.2.2 Поиск и фильтрация событий ..... 10

2.2.3 Экспорт записей журнала ..... 12

2.3 Настройка отображения стандартных ролей в АРМ: Администрирование ..... 12

**3 Настройка подключений из внешних источников ..... 14**

3.1 Настройка подключения к БД ClickHouse ..... 14

3.2 Настройка подключения к внешней БД PostgreSQL\MS SQL\MySQL\Oracle ..... 14

3.3 Настройка подключения к файлам-источникам ..... 14

3.4 Настройка подключения к Google Sheets ..... 15

3.5 Настройка RLS при подключении к внешним БД PostgreSQL ..... 17

**4 Использование сторонних планировщиков задач ..... 19**

**5 Резервное копирование ..... 20**

5.1 Создание дампа БД ..... 20

5.2 Создание файловой копии БД ..... 20

5.3 Восстановление из резервной копии ..... 20

**6 Оптимизация работы БД ..... 22**

**7 Процедура вакуума БД ..... 24**

**8 Оптимизация WildFly и Java ..... 25**

**9 Настройка web-приложения в режиме iFrame ..... 26**

**10 Особенности настройки мобильных приложений ..... 27**

**11 Организация мониторинга системы ..... 28**

11.1 Работоспособность приложения ..... 28

11.1.1 Доступность ..... 28

11.1.2 Проверка срока действия лицензии Навигатора ..... 28
11.1.3 Проверка срока действия серверного сертификата ..... 29
11.1.4 Проверка используемой памяти Java Runtime ..... 29
11.2 Работоспособность и метрики СУБД ..... 30
11.3 Метрики сервера по данным операционной системы ..... 30

# 1 Настройка аутентификации пользователей

Навигатор по умолчанию поддерживает следующие типы аутентификации:

- Mutual-TLS;
- Login\password;
- LDAP;
- OpenID;

Основным режимом аутентификации для Навигатора является BASIC (авторизация по логину\паролю), однако, его можно изменить. Для этого необходимо в файле `путь_к_WildFly/standalone/configuration/standalone.xml` в разделе `<system-properties>` изменить значение параметра `<property name="java:/ru/sberbank/portal/inavigator/AUTH_METHOD" value="BASIC"/>`

следующим образом:

- X509 для аутентификации по сертификату;
- BASIC для аутентификации по логину\паролю (по умолчанию);
- LDAP для аутентификации по LDAP;
- OPENID при аутентификации при помощи сторонних серверов;

Данный параметр регулирует способ аутентификации для всех пользователей Навигатора.

## 1.1 Особенности настройки отдельных типов аутентификации:

### 1.1.1 Аутентификация по сертификату

Настройка аутентификации по сертификатам пользователей подробно описана в Инструкции по установке (см. п. 4.2.6).

Для работы аутентификации сертификат доверенного корневого центра сертификации должен быть установлен внутрь контейнера truststore.jks. Это можно сделать при помощи программы Keystore Explorer или при помощи штатных средств работы с сертификатами Java.

**!Внимание!** При использовании аутентификации по сертификатам функция выхода из системы недоступна.

### 1.1.2 Аутентификация по логину\паролю

Текущая реализация парольной политики подразумевает следующие проверки для вводимого пароля:

не менее 8 и не более 128 символов;

как минимум одна заглавная и одна строчная буква;

только латинские буквы;

как минимум одна цифра;

только арабские цифры;

без пробелов;

в пароле используется минимум 4 разных символа;

в пароле отсутствуют три рядом стоящих знака из следующих последовательностей (как слева направо, так и справа налево), например:

- 1234567890-=
- qwertyuiop[]asdfghjkl;'zxcvbnm,./
- qazwsxedcrfvtgbyhnujmik,ol.p;/['
- 741852963 и т.п.

Управление парольной политикой внутри Навигатора на текущий момент не осуществляется.

### 1.1.3 LDAP – аутентификация

Для реализации LDAP-аутентификации необходимо, чтобы при создании пользователей в Навигаторе e-mail и имя пользователя совпадали с соответствующими полями в Active Directory.

Механизм аутентификации следующий:

1. Пользователь вводит свой логин\пароль. Для логина осуществляется поиск в AD по любому из полей: cn, distinguishedName, samaccountname
2. Из найденной записи извлекаются поля: displayName, userPrincipalName и третье поле, которое задаётся в настройке Навигатора < `property name="java:/ru/sberbank/portal/inavigator2/SPRING_LDAP_EMAIL_ATTRIBUTE_NAME" value="`**`имя_поля_с_e-mail`**`"`
3. Затем делается сопоставление логина с ролевой моделью Навигатора по следующей логике:

rm.tUser.sLogin == displayName OR rm.tUser.sLogin == userPrincipalName OR rm.tUser.sEmail == {третье поле из настроек} OR rm.tUser.sEmailAD == {третье поле из настроек}

Для работы аутентификации через LDAP дополнительно в файл `путь_к_WildFly/standalone/configuration/standalone.xml` в раздел `<system-properties>` необходимо добавить обязательные параметры:

```
<property name="java:/ru/sberbank/portal/inavigator2/LDAP_DOMAIN_CONFIG"
value="{&quot;items&quot;: [ {&quot;domen&quot;: &quot;netbios
    _имя_домена&quot;, &quot;baseDN&quot;:&quot;DC=ваш_домен,DC=ru&quot;,
&quot;serverLDAP&quot;:&quot;ldap://ip
    -адрес_сервера_
    LDAP
    :порт&quot;} ]}"/>
```

```
<property name="java:/ru/sberbank/portal/inavigator2/LDAPSERVER_DOMAIN"
value="netbios_ имя _ домена"/>
```

### 1.1.4 OpenID - аутентификация

Для работы аутентификации через Open-ID дополнительно в файл `путь_к_WildFly/standalone/configuration/standalone.xml` в раздел `<system-properties>` необходимо добавить обязательные параметры:

`<property name="java:/ru/sberbank/portal/inavigator/OIDC_CLIENT_ID" value="ваше_значение"/>` - Идентификатор клиента (Client). Создается в провайдере услуги аутентификации.

`<property name="java:/ru/sberbank/portal/inavigator/OIDC_JWK_URL" value="url_к_ключам_валидации"/>` - По этому адресу Навигатор получает ключи для валидации JWT токена.

`<property name="java:/ru/sberbank/portal/inavigator/ OIDC_PROVIDER_AUTH_URL" value=" url_к_странице_логина"/>` - Путь к странице логина у провайдера услуги аутентификации.

`<property name="java:/ru/sberbank/portal/inavigator/ OIDC_TOKEN_URL" value="url_к_токену_провайдера"/>` - По этому адресу Навигатор запрашивает токен у провайдера услуги аутентификации.

Также существуют необязательные параметры, применяемые в случае необходимости или для отладки:

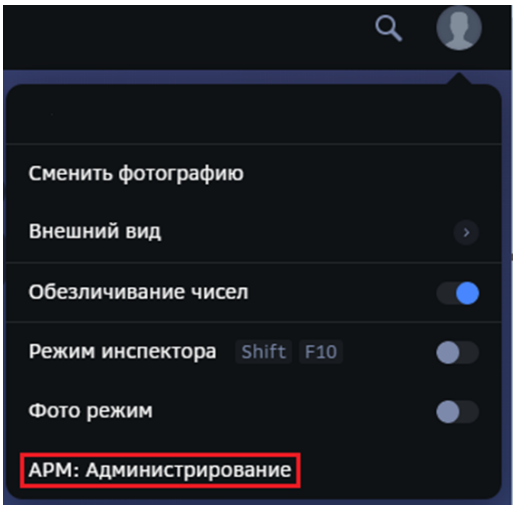
`<property name="java:/ru/sberbank/portal/inavigator/OIDC_CLIENT_SECRET" value="ваше_значение"/>` - Для некоторых провайдеров (Например, Google) нужно указывать дополнительно секрет, который создается для каждого клиента у провайдера услуги аутентификации.

`<property name="java:/ru/sberbank/portal/inavigator/OIDC_LOGOUT_URL" value="url_к_странице_логаута"/>` - Путь к странице логаута. После логаута в Навигаторе будет вызвано перенаправление на этот адрес

`<property name="java:/ru/sberbank/portal/inavigator/OIDC_DISABLE_SSL_VERIFICATION" value=""/>` - Используется для отладки. При наличии настройки в значении true серверный сертификат проверяться не будет, в значении false - проверяется стандартным образом.

# 2 Действия внутри АРМ

Внутри АРМ Навигатора, согласно ролевой модели, администратору доступно управление лицензиями и просмотр системного журнала. Оба эти раздела находятся в меню Система основного меню Навигатора. Для того, чтобы попасть в режим администрирования, нажмите на свой профиль в правом верхнем углу и выберете в раскрывшемся подменю «**АРМ: Администрирование**».

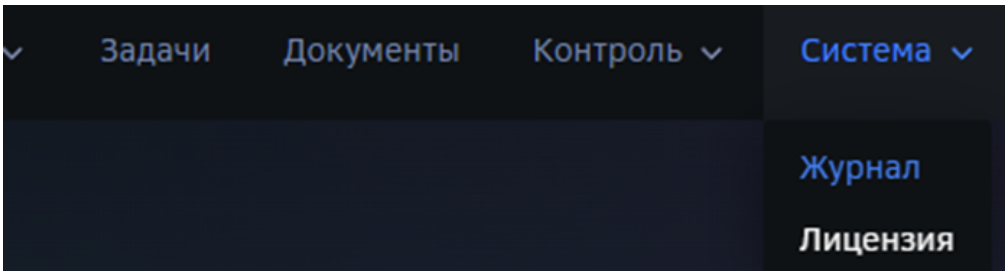



Для получения информации по остальным подменю раздела «**АРМ: Администрирование**» обратитесь к «Руководству пользователя».

## 2.1 Лицензия

Раздел Лицензия позволяет получить информацию об статусе действующей лицензии на продукт, в т.ч. период действия, количество пользователей, лимит дискового пространства для файлов и документов.

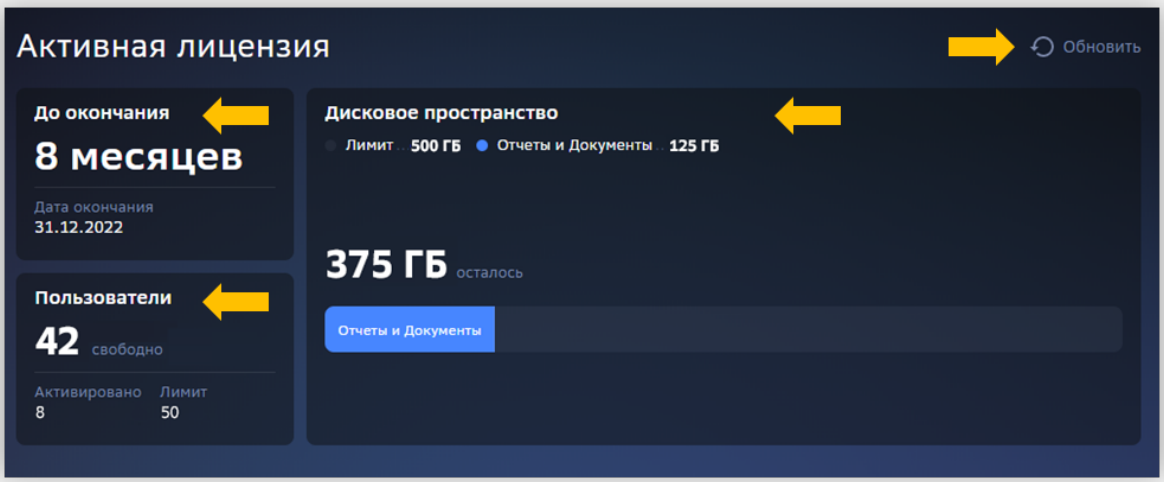
В основном меню АРМ пункт **Лицензия** находится внутри меню **Система**.




**При выборе** этого пункта на экране отображается **основная информация о** вашей **лицензии**:

- Количество месяцев и дата окончания
- Информация о количестве пользователей — лимит, активированных и свободных
- Дисковое пространство — лимит и текущий объем раздела Документы

**Для обновления** информации **нажмите** кнопку **Обновить** справа вверху экрана Лицензии.

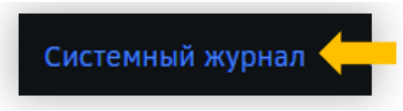



## 2.2 Системный журнал

Системный журнал используется для просмотра информации о системных событиях и действиях пользователя в АРМ Навигатора. Раздел позволяет просматривать записи, искать по различным параметрам и экспортировать для дальнейшего анализа. Кроме того, журнал можно использовать для отладки в среде разработки, т.к. он позволяет просматривать XML-параметры, передаваемые процедурам и функциям СУБД.

### 2.2.1 Элементы экрана

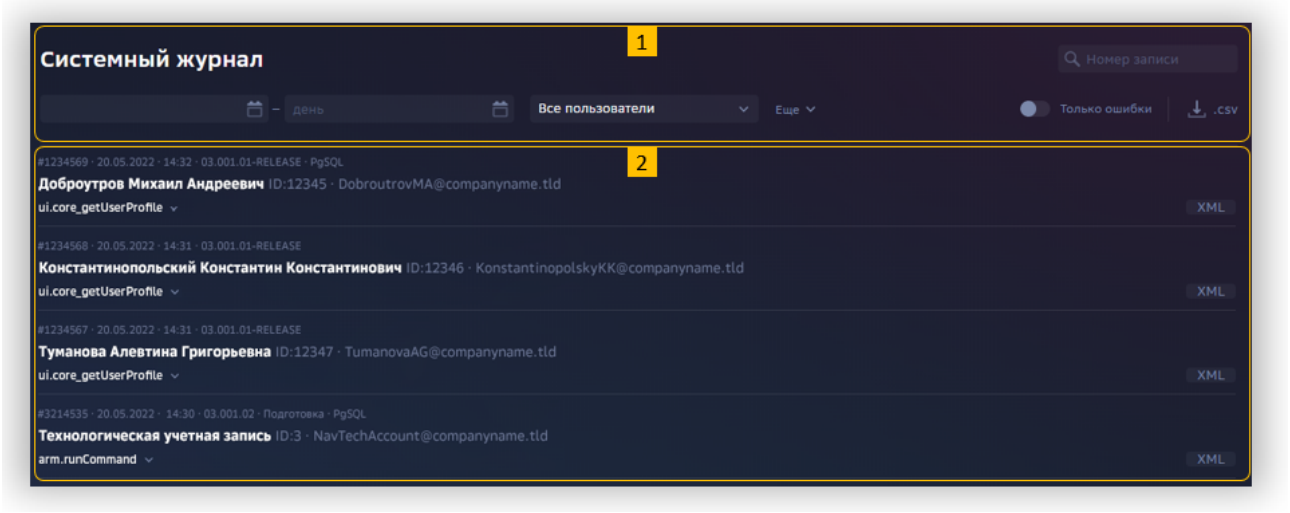
**1. Выберите в** основном **меню** АРМ пункт **Системный журнал**, чтобы начать работу с инструментом.




**2. В верхней части экрана** (1) располагаются **инструменты работы со списком** событий: **поиск**, **фильтрация** и **экспорт**

**В нижней части экрана** (2) отображаются **события журнала** с детальной информацией:

- Номер записи в журнале
- Дата и время события
- Версия продукта
- Среда (подготовка, пром-среда)
- ФИО пользователя или технологической учетной записи (ID в системе и email)
- Запущенная системная процедура и параметры
- XML-параметры, переданные в процедуру (если требуются)




## 2.2.2 Поиск и фильтрация событий

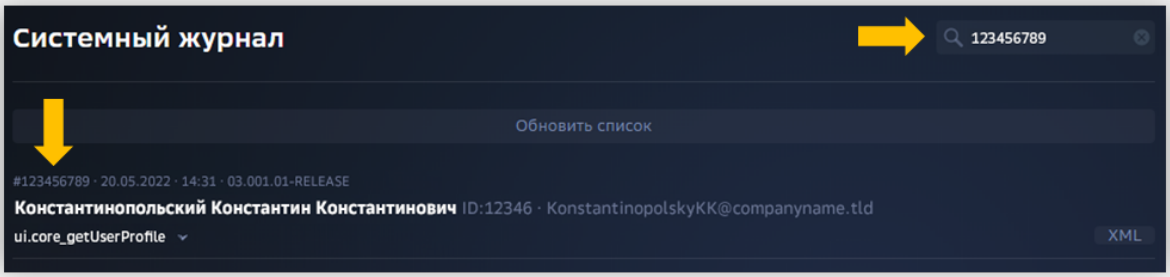
В журнале событий существуют различные инструменты и способы поиска и фильтрации событий.

**1.**

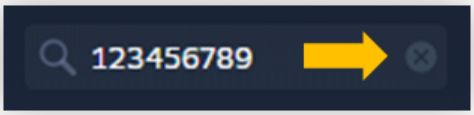
Самым быстрым является **поиск конкретной записи журнала**, если **известен** ее **номер**. Например, в сообщении об ошибке указан № записи в журнале, чтобы администратору можно было быстро ее найти. **Введите номер записи** в поле поиска.




В списке записей останется **только запись**, номер которой **соответствует введенному**.




Чтобы очистить поле поиска нажмите иконку справа от номера записи.




**2.**

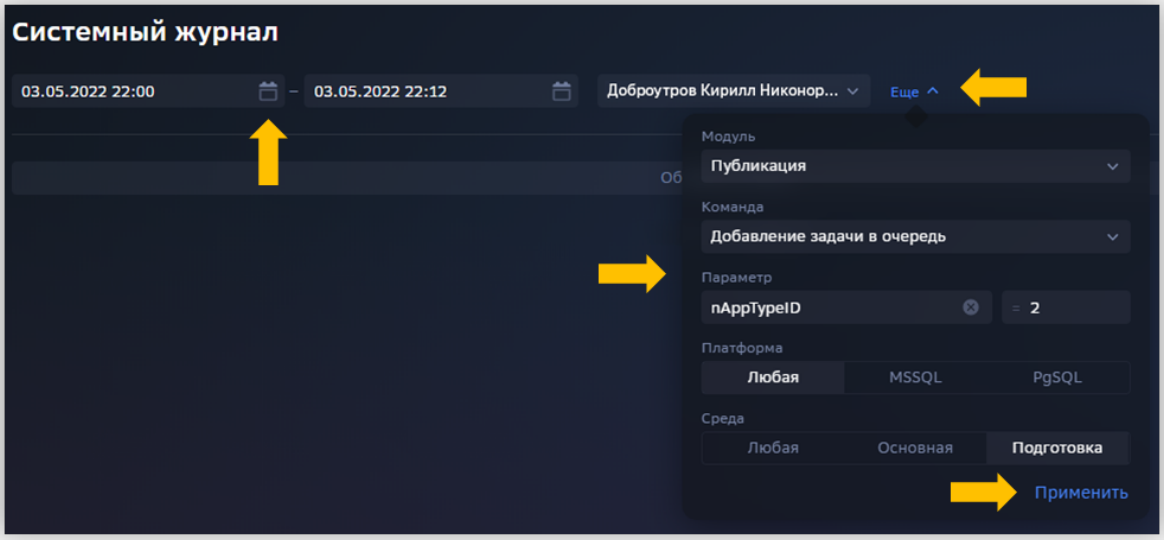
Переключатель **Только ошибки** позволяет отфильтровать в Системном журнале те записи, которые **завершились с ошибкой**. Они помечены **красным индикатором** в списке.

**3.**

**Панель настраиваемых фильтров** над списком событий позволяет отфильтровать записи журнала по **гибким критериям**, в т.ч.:

- **Период** (диапазон дат) записей
- **Пользователь**
- Расширенный **фильтр**
  - **Модуль** Навигатора
  - Выполняемая **команда**
  - Значение **параметра**
  - **Платформа**
  - **Среда**

Например, нужно найти все события по задачам добавления в очередь в модуле Публикация (с дополнительным параметром nAppTypeID = 2) в среде Подготовки данных пользователя Доброутрова К. Н. за период с 22:00 по 22:12, 3 мая. Тогда настроенные фильтры будут выглядеть следующим образом




После **нажатия** кнопки **Применить** в списке журнала останутся только события, **отвечающие фильтрам поиска**.

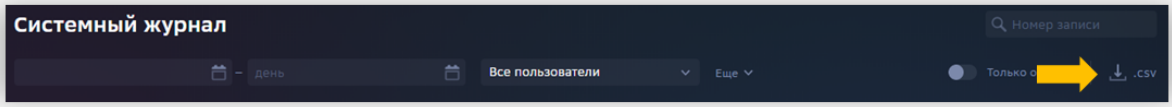
Чтобы **очистить фильтры** — **удалите** выбранные **значения** или **выберите** пункт **Системный журнал в меню** APM (откроется вид журнала по умолчанию).

### 2.2.3 Экспорт записей журнала

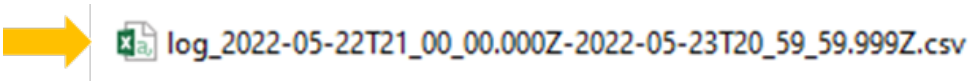
Записи Системного журнала можно экспортировать в формат CSV (Comma-Separated Values — значения, разделенные запятыми) для дальнейшего использования (отправки, анализа сторонними средствами и т.п.) В список экспорта попадают все записи, как с примененными фильтрами, так и без них).

1. **Сформируйте список записей журнала** с нужными параметрами — настройте фильтры, укажите нужно ли отображать ошибки или нет, **убедитесь**, что **обновление** списка **завершено**.

2. **Нажмите** кнопку **экспорта в CSV** (крайняя справа в строке расширенных филтров)




3. **По окончании экспорта**, файл с расширением **. CSV будет сохранен** в папку **Загрузки** вашего ПК. Файл можно **открыть с помощью текстового** редактора или **редактора электронных таблиц**.




## 2.3 Настройка отображения стандартных ролей в АРМ: Администрирование

В Навигаторе используется гибкая ролевая система пользовательских ролей с привязкой как к функционалу, так и к предметным областям. При установке Навигатора создаются базовые роли, которых достаточно в большинстве случаев. Однако, при необходимости, можно создать аналогичные роли или свои роли с определенным функционалом, а также отключить встроенные.

Для отключения встроенных ролей пользователя необходимо выполнить SQL-запрос:

```
UPDATE rm.trole SET ishide = true WHERE nid in ( id_роли );
```

где id_роли должно соответствовать nid предустановленной роли согласно таблице:

| nid | Имя роли (sname) |
|---|---|
| 2 | Администратор Навигатора |
| 5 | Бизнес-администратор Навигатора |
| 6 | Владелец предметной области |
| 7 | Специалист поддержки |
| 8 | Администратор доступа |
| 11 | Инспектор настроек |

| nid | Имя роли (sname) |
|---|---|
| 12 | Пользователь Навигатора |
| 15 | ИТ-администратор |

Для включения ранее отключенных встроенных ролей необходимо выполнить SQL-запрос:

```
UPDATE rm.trole SET ishide = false WHERE nid in ( id_роли );
```

# 3 Настройка подключений из внешних источников

Навигатор поддерживает использование в качестве источников подключения к сторонним БД:

- ClickHouse
- PostgreSQL
- MS SQL
- MySQL
- Oracle

А также подключения к следующим типам файлов:

- CSV
- XML
- JSON

**!Внимание!** Перед настройкой подключений из внешних источников следует убедится, что переключатель disable-external-sources (находится в меню АРМ: Администрирование - Система - Настройки - Опции реализации) установлен в положении **false**, а переключатель enable-connections - в положении **true**.

## 3.1 Настройка подключения к БД ClickHouse

Для настройки подключения к внешней БД ClickHouse необходимо выполнить следующие шаги:

- Установить Data Wrapper clickhouse_fdw (**не входит** в комплект, любой вариант со страницы https://wiki.postgresql.org/wiki/Foreign_data_wrappers)
- Создать расширение командой:

```
CREATE FOREIGN DATA WRAPPER clickhouse_fdw
ALTER FOREIGN DATA WRAPPER clickhouse_fdw OWNER TO as_admin;
GRANT USAGE ON FOREIGN DATA WRAPPER clickhouse_fdw TO as_admin;
```

Далее необходимо выполнить настройку пользовательских подключений в АРМ: Администрирование (см. Руководство пользователя АРМ).

## 3.2 Настройка подключения к внешней БД PostgreSQL\MS SQL\MySQL\Oracle

Для настройки подключения к внешней БД PostgreSQL\MS SQL\MySQL\Oracle необходимо выполнить следующие шаги:

- **Для MySQL, MS SQL, Oracle** установить Data Wrapper для необходимой БД (**не входит** в комплект, любой вариант со страницы https://wiki.postgresql.org/wiki/Foreign_data_wrappers)
- Создать расширение командой:

```
CREATE EXTENSION "название бд"_fdw
SCHEMA "ext";
GRANT USAGE ON FOREIGN DATA WRAPPER "название бд" TO as_admin;
```

Где "название бд" - имя вашей БД латиницей, например, mysql_fdw или oracle_fdw.

Далее необходимо выполнить настройку пользовательских подключений в АРМ: Администрирование (см. Руководство пользователя АРМ).

## 3.3 Настройка подключения к файлам-источникам

Для настройки подключения к файловым источникам необходимо выполнить следующие шаги:

- Задать права доступа к функциям для пользователя as_admin:

```
GRANT EXECUTE ON FUNCTION pg_read_file(text) TO as_admin;
GRANT EXECUTE ON FUNCTION pg_ls_dir(text) TO as_admin;
GRANT EXECUTE ON FUNCTION pg_read_binary_file(text) TO as_admin;
GRANT pg_read_server_files TO as_admin;
GRANT EXECUTE ON FUNCTION pg_ls_dir(text,bool,bool) TO as_admin;
```

- **Для CSV** установить Data Wrapper (**не входит** в комплект) со страницы https://wiki.postgresql.org/wiki/Foreign_data_wrappers. Либо выполнить команду CREATE EXTENSION file_fdw.
- В Навигаторе реализован умный поиск в строке подключения файлов по наименованию папок и файлов, однако, для его работы требуется установить расширение, запустив команду CREATE EXTENSION pg_trgm.

Далее необходимо выполнить настройку пользовательских подключений в АРМ: Администрирование (см. Руководство пользователя АРМ).

## 3.4 Настройка подключения к Google Sheets

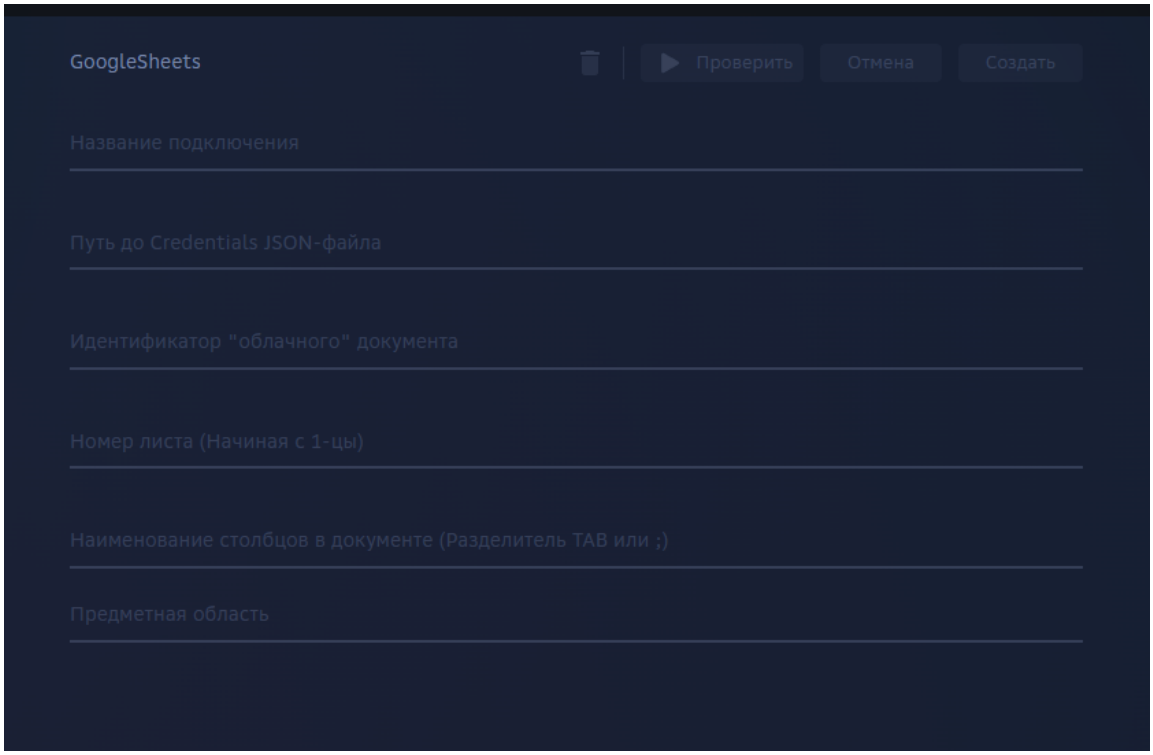
**!Внимание!** Коннектор к GoogleSheets на текущий момент позволяет подключение только к первому листу документа.
Заголовки должны быть у всех столбцов таблицы, так как коннектор в качестве строки заголовков использует первую сверху вниз строку с наибольшей длиной.

Для работы коннектора нужно установить пакеты postgresql-server-dev-15, python3-dev, python3-pip (названия пакетов приведены для Linux Ubuntu), а так же установить Multicorn 2 (https://github.com/pgsql-io/multicorn2.git).

- Установить FDW GoogelSheetsApi согласно инструкции на странице (https://github.com/lincolnturner/gspreadsheet_fdw.git).
- Подключиться к БД Навигатора любым удобным способом под суперпользователем (postgres), выполнить следующие команды:

```
CREATE EXTENSION multicorn;
CREATE SERVER navigator_GoogleSpreadsheetFDW FOREIGN DATA WRAPPER multicorn options ( wrapper
'gspreadsheet_fdw.GspreadsheetFdw' );
GRANT EXECUTE ON FUNCTION pg_read_file(text) TO as_admin;
GRANT EXECUTE ON FUNCTION pg_ls_dir(text) TO as_admin;
GRANT EXECUTE ON FUNCTION pg_read_binary_file(text) TO as_admin;
GRANT pg_read_server_files TO as_admin;
GRANT EXECUTE ON FUNCTION pg_ls_dir(text,bool,bool) TO as_admin;
GRANT USAGE ON FOREIGN DATA WRAPPER multicorn TO as_admin;
GRANT ALL ON FOREIGN SERVER navigator_GoogleSpreadsheetFDW TO as_admin;
```

- Подготовить json-файл с учетными данными:
  - Создать сервисный аккаунт Google;
  - Зайти под этим аккаунтом в УЗ Google, включить в настройках Google API;
  - Сгенерировать и скачать JSON с ключом сервисной УЗ;
  - Задать права для сервисной УЗ в необходимом документе-источнике.
- В Навигаторе в вкладке «Система» - «Подключения» добавить подключение типа GoogleSheets и заполнить необходимые поля:




- В поле «Наименование подключения» указывается произвольное название подключения;
- В поле «Путь до Credentials JSON-файла» указывается абсолютный путь до JSON, сгенерированного на шаге 5. Для корректной работы JSON должен находиться на сервере БД PostgreSQL и быть доступным системной УЗ postgres;
- В поле «Идентификатор облачного документа» указывается API key (44-хсимвольная строка из ссылки на документ в браузере сразу после последовательности /spreadsheets/d/);
- В поле «Номер листа» необходимо указать ID листа, который можно получить при помощи следующего скрипта:

```
import gspread

from oauth2client.service_account import ServiceAccountCredentials

scopes = ['https://spreadsheets.google.com/feeds']

path = input("Введите путь до json-файла с настройками входа в УЗ: ")

table = input("Введите идентификатор таблицы: ")

credentials = ServiceAccountCredentials.from_json_keyfile_name(path, scopes)

client =
```

```
gspread.authorize(credentials)

                                        sh =
client.open_by_key(table)

                                        worksheet_list =
sh.worksheets()

                                        for i in worksheet_list:

            print ("Название страницы: " +
str(i).removeprefix("<Worksheet ").replace("
id:","\t\t\tИдентификатор страницы:").removesuffix(">"))
```

- В поле «Наименование столбцов в документе» указывается наименование столбцов, записанных через табуляцию или через точку с запятой;
- В поле «Предметная область» выбирается необходимая предметная область, к которой будет привязано подключение.

## 3.5 Настройка RLS при подключении к внешним БД PostgreSQL

В Навигаторе имеется возможность использования встроенного механизма RLS PostgreSQL при подключении к удаленным БД. Для RLS используются только источники подключения. Данные при этом не сохраняются в Навигаторе, а передаются в напрямую из сторонней БД в виджеты Навигатора.

Для настройки использования RLS необходимо выполнить следующие шаги:

Перейти в АРМ: Администрирование;
В меню Система - Настройки - Опции реализации добавить параметр enable-foreign-rls со значением true;
Убедится, что переключатель disable-external-sources установлен в положении **false**;
В меню Система - Подключения создать подключение с типом PostgreSQLwRLS;
В меню Система - Настройки - Настройки для внешних таблиц (RLS) необходимо добавить строку и заполнить следующие поля:

| | nid | islocalmapping | smappingtable | sparamrls | smappingrls | srolerls | ssystemrole | ntimeout |
|---|---|---|---|---|---|---|---|---|
| 1 | 777 | false | "Test_schema"."mapping_table" | slogin | slogin | srole | navigator | 60 |

- nid - номер соданного подключения;
- islocalmapping – Расположение таблицы сопоставления ролей и пользователей (false – во внешней базе или true – в базе Навигатора);
- smappingtable – Имя таблицы сопоставления пользователей;
- sparamrls – Имя столбца таблицы rm.tuser БД Навигатора, по которому будет производиться сопоставление;

- smappingrls – Имя столбца из таблицы сопоставления пользователей, по которому будет сопоставление параметра пользователя и роли;
- srolerls – Имя столбца из таблицы сопоставления пользователя, по которому будет определяться роль пользователя;
- systemrole – Наименование роли, под которой будет осуществляться доступ к таблице с сопоставлениями пользователей или обращение к таблицам, если роль не была определена;
- ntimeout – Время до таймаута работы запроса на удалённом сервере (в секундах);

На скриншоте:

Таблица сопоставления пользователей находится на внешнем сервере.

- nid - **777**
- islocalmapping – **false** – во внешней базе
- smappingtable – **"Test_schema"."mapping_table"**
- sparamrls – **slogin**
- smappingrls – **slogin**
- srolerls – **srole**
- systemrole – **navigator**
- ntimeout – **60**

# 4 Использование сторонних планировщиков задач

При установке Навигатора на системы, для которых не существует сборки штатного планировщика pg_cron, рекомендуется использовать любой другой планировщик, который может запускать внешние команды по расписанию.

Общая маршрутизирующая задача `CALL comm.watchdog();` должна запускаться в одном экземпляре каждую минуту на основном сервере или каждом сервере при установке сред на разные сервера СУБД.

Для работы очереди задач необходимо запускать следующие процедуры:

| Команда запуска | Минимальное (рекомендуемое) количество заданий в планировщике | Расписание | Запуск на сервере СУБД основной среды или при одной среде установки | Запуск на остальных серверах при установке нескольких сред | Описание |
|---|---|---|---|---|---|
| CALL queue.router(); | 1 (только на основном сервере) | каждую минуту | ДА | НЕТ | Очередь задач. Маршрутизатор |
| CALL queue.worker(0); | 1 | каждую минуту | ДА | ДА | Очередь задач. Исполнительный модуль № 0 - выделенный для технических задач |
| CALL queue.worker(N); | 1 (8) (N-номер по порядку от 1 и далее) | каждую минуту | ДА | ДА | Очередь задач. Исполнительный модуль |
| CALL queue.c_worker(N); | 1 (2) (N-номер по порядку от 1 и далее) | каждую минуту | ДА | НЕТ | Очередь задач. Обработчик условных переходов |
| CALL queue.jobrun(null); | 1 | каждую минуту | ДА | НЕТ | Очередь задач. Планировщик. Модуль Вычисления |
| CALL queue.jobrunner(); | 1 | каждую минуту | ДА | НЕТ | Очередь задач. Планировщик. Исполняемый модуль |

# 5 Резервное копирование

Резервное копирование БД Навигатора может осуществляться тремя разными способами:

- Созданием дампа БД
- Копированием файлов БД
- Сторонним ПО

Создание дампа БД предпочтительно в случае небольшого размера БД и не требует остановки служб PostgreSQL, за исключением служб планировщика pg_cron.

Копирование файлов БД применяется при необходимости сделать копию с кластера или при большом объеме данных.

Рекомендуем совмещать эти методы для обеспечения непрерывной работоспособности БД. Типичный сценарий для планирования расписания резервного копирования – ежедневное\еженедельное создание дампа БД и еженедельное\ежемесячное копирование файлов БД.

## 5.1 Создание дампа БД

Создание дампа БД рекомендуется использовать в качестве основного способа резервного копирования, так как при создании дампа не требуется остановка БД.

Дамп БД создается из-под суперпользователя. Для создания дампа нужно выполнить команду:

```
pg_dumpall-h ip_адрес_сервера -U postgres -W | gzip > ИМЯ_АРХИВА_С_БД.gz
```

За подробностями по использованию команд `pg_dumpall` и `pg_dump` обращайтесь к руководству по PostgreSQL.

## 5.2 Создание файловой копии БД

В отличие от дампа БД, файловую копию БД нужно создавать при полностью остановленном сервере БД. Для того, чтобы правильно остановить сервер, необходимо:

- отключить выполняемые процессы `pg_cron`:

```
SELECT cron.alter_job (<TASK_ID>, null, null, null, null, false);
```

- остановить службу БД;
- убедится, что служба PostgreSQL перешла в состояние «остановлена»;
- выполнить команду на создание бэкапа:

```
sudo tar -cf backup.tar Путь_к_папке_data
```

Внимание! Восстановление из файловой копии, созданной при работающей БД, будет невозможно!

Также к файловым копиям можно отнести создание полной копии или снэпшота виртуальной машины (в случае размещения БД на ВМ). В таком случае смотрите документацию производителя программы виртуализации.

## 5.3 Восстановление из резервной копии

Восстановить БД из дампа можно при помощи следующей команды:

```
gunzip -c ИМЯ_АРХИВА_С_БД.gz | psql -f postgres
```

После чего нужно проверить в БД наличие табличных пространств Навигатора.

Для восстановления из файловой копии необходимо:

- Остановить целевую БД;
- Удалить содержимое каталога
  ```
  Путь_к_папке_data;
  ```
- Распаковать файлы из архива с бэкапом в целевые каталоги;
- В папке `…/postgresql/15/main/` проверить наличие файла `recovery.conf` (создать его в случае необходимости). Этот файл должен содержать строку

```
restore command = 'cp /tmp/[clientName]/wal/%f %p
```

- Проверить, что владельцем файла `recovery.conf` является пользователь postgres, и что права на файл выставлены в конфигурации 600
- Запустить БД стандартным образом, после чего начнется процесс восстановления БД
- По окончании процесса восстановления файл `recovery.conf` будет переименован в `recovery.done`, после чего БД будет готова к работе.

За более подробной информацией обращайтесь к документации по вашей версии PostgreSQL.

# 6 Оптимизация работы БД

Существует несколько неочевидных настроек БД, предназначенных для оптимизации работы базы в зависимости от нагрузки и производительности сервера СУБД. Эти настройки находятся в файле **postgresql . conf**. Рекомендации ниже приведены из расчета на слабую, среднюю и высокую нагрузку (10, 100 и 1000 пользователей соответственно).

| **Параметр:** | **Рекомендуемые значения:** | | |
|---|---|---|---|
| | 10 пользователей: | 100 пользователей: | 1000 пользователей: |
| max_connections | 50 | 120 | 500 |
| max_prepared_transactions | 50 | 120 | 500 |
| autovacuum_max_workers | 8 | | |
| autovacuum_naptime | 10s | | |
| autovacuum_vacuum_cost_limit | 4000 | | |
| autovacuum_vacuum_cost_delay | 2ms | | |
| vacuum_cost_page_hit | 1 | | |
| vacuum_cost_page_miss | 5 | | |
| vacuum_cost_page_dirty | 15 | | |
| autovacuum_vacuum_scale_factor | 0.001 | | |
| autovacuum_analyze_scale_factor | 0.001 | | |
| shared_buffers | 25 - 30% RAM | | |
| temp_buffers | 64Mb | 128Mb | 256Mb |
| work_mem | 64Mb | 128Mb | 256Mb |
| maintenance_work_mem | 1Gb | | |
| autovacuum_work_mem | -1 | | |
| effective_cache_size | RAM-shared_buffers (и оставить 5-10% RAM на ОС) | | |

| | | |
|---|---|---|
| random_page_cost | 1.2 | |
| track_functions | pl | |
| max_wal_size | 4Gb | 8Gb |
| min_wal_size | 1Gb | 2Gb |
| wal_buffers | 16Mb | |
| max_worker_processes | 32 | 64 |
| max_parallel_workers_per_gather | 4 | |
| max_parallel_workers | 16 | 32 |
| max_parallel_maintenance_workers | 4 | |
| jit | off | |
| synchronous_commit | off | |
| wal_sync_method | fdatasync | |
| shared_preload_libraries | дополнить значением 'pg_stat_statements' | |

# 7 Процедура вакуума БД

Процедура вакуума предназначена для очистки БД от накопившихся временных данных. Перед процедурой вакуума крайне рекомендуется проводить резервное копирование.

Существует две версии процедуры вакуума – обычная и полная. Отличие в том, что при полном вакууме требуется гораздо большее время на очистку таблиц и исключительный доступ к таблице, тогда как обычный вакуум позволяет частичную работу с таблицей (см. руководство по PostgreSQL) и выполняется гораздо быстрее, однако таблицы чистятся менее эффективно.

Для запуска процедуры вакуума необходимо выполнить от имени суперпользователя (postgres) команду `VACUUM ANALYZE` без указания таблиц. Однако, при необходимости, можно запускать процедуру для конкретной таблицы, указав её имя в качестве аргумента.

Следует помнить, что, хотя при процедуре вакуума таблицы чистятся, однако место, занимаемое БД на диске, не изменяется. Если вам необходимо получить свободное место на диске за счет очистки БД, нужно запускать процедуру полного вакуума.

Выполнение процедуры полного вакуума отличается от обычного тем, что при полном вакууме требуется исключительный доступ к БД, соответственно, все остальные сессии должны быть сброшены. Так как практически такое же требование предъявляется к процедуре файловой резервной копии БД, мы рекомендуем совмещать эти два действия – сначала проводить полный вакуум, затем делать файловую копию БД.

Для запуска процедуры полного вакуума необходимо ввести команду `VACUUM FULL` от имени суперпользвателя.

Процедуру вакуума рекомендуем запускать ежедневно, а полного вакуума – раз в неделю при большом количестве обновленных значений в таблицах.

# 8 Оптимизация WildFly и Java

Работу web-приложений можно оптимизировать путем изменения настроек в файле `путь_к_WildFly/bin/standalone.conf`

Основные параметры, отвечающие за работу Java-машины:

`-XmS` и `-XmX` – Отвечают за размер памяти, выделенный для Java heap.

`-XX:MaxMetaspaceSize` – Отвечает за выделение памяти вне пределов heap.

По умолчанию эти значения заданы, как абсолютные величины в мегабайтах (см. Руководство по установке, п. 1.2.1 раздела 4.1), однако параметры -XmS и -XmX можно задавать в процентном соотношении к используемой оперативной памяти. Для этого предназначен параметр и `-XX:MaxRAMPercentage`. Значение этого параметра необходимо указывать числом в процентах от всего объёма RAM (пример: `-XX:MaxRAMPercentage=50.0`). Следует учитывать, что этот параметр игнорируется, если заданы параметры –XmS и –XmX.

Для предотвращения ошибок, связанных с переполнением памяти при одновременной работе с БД большого количества пользователей в Навигаторе задан таймаут подключения приложения к БД. Его значение можно изменить в зависимости от нагрузки на БД.

Для этого в файле `путь_к_WildFly/standalone/configuration/navigator-proxyserver-config.yml` изменить параметр `defaultTimeout`. По умолчанию его значение – 60 секунд.

# 9 Настройка web-приложения в режиме iFrame

Если требуется отобразить какой-либо дэшборд в режиме iFrame так, чтобы скрыть навигационное меню, а также включить по-умолчанию цветовую тему, то необходимо через URL задать определенные параметры.

Параметры URL: `#iframe:theme-type=light&hide-navigation=true`

Описание:

`#iframe` – включить режим работы iFrame

`theme-type` – включить заданную цветовую тему. Возможные варианты: galaxy, dark, light

`hide-navigation` – скрыть заголовки, меню и опции профиля пользователя

Пример: https://navigator.my-company.ru/gdash/100500/200500#iframe:theme-type=light&hide-navigation=true

Здесь

- https://navigator.my-company.ru/gdash/100500/200500 – это ссылка на дэшборд и при необходимости конкретный экран
- #iframe:theme-type=light&hide-navigation=true – включить iFrame со светлой цветовой схемой и скрыть навигационное меню

Особенности:

Режим iFrame работает только при запуске web-приложение из iframe страницы. Если открыть по прямой ссылке, данная команда не применяется.

# 10 Особенности настройки мобильных приложений

При установке мобильных приложений (как для iOS, так и для Android), при первом запуске необходимо указать адрес сервера, к которому происходит подключение. При первом подключении происходит запрос типа авторизации, и дальнейшие действия пользователя зависят от выбранного типа авторизации.

Изменить сервер для подключения можно следующим образом:

Android – очистить хранилище приложения в системных настройках Android, перезапустить приложение;

iOS – В системных настройках iOS для приложения Навигатор включить опцию «Удалить пароли и сертификаты при следующем входе».

# 11 Организация мониторинга системы

Для контроля работоспособности, необходимо проверять следующие параметры системы:

## 11.1 Работоспособность приложения

### 11.1.1 Доступность

Проверка выполняется командой https://navigator.my-company.ru/inavigator2-backend/admin/healthCheck . В момент проверки также происходит тестирование доступа к базе данных.

В случае успеха запрос возвращает пустой JSON с кодом HTTP-200

```
{"root":""}
```

В случае проблем возвращается ошибка с кодом HTTP-500

```
{
    "message": "Произошла системная ошибка. Попробуйте еще раз позднее или обратитесь в
техническую поддержку по адресу []",
    "timestamp": "2024-06-18 19:47:57",
    "messageID": "2075",
    "errorID": 50001,
    "requestID": "bf7b9872-7e29-4343-adc0-6f7bd40cc30f"
}
```

После чего в логах WildFly по `requestID` можно найти более детальное описание ошибки.

Техническое описание ошибки не выводится в API по соображениям безопасности для предотвращения раскрытия деталей реализации системы.

Данная команда спроектирована таким образом, что создаёт минимальную нагрузку на систему, что позволяет использовать её в сетевом балансировщике для проверки доступности узла в кластере. Например для HAProxy это можно сделать таким образом:

```
backend webservers
    option httpchk
    http-check send meth GET uri /inavigator2-backend/admin/healthCheck hdr Content-
Type application/json body "{\"root\":\"\"}"
    server server1 192.168.50.2:80 check
    server server2 192.168.50.3:80 check
    server server3 192.168.50.4:80 check
```

### 11.1.2 Проверка срока действия лицензии Навигатора

Проверка выполняется командой https://navigator.my-company.ru/inavigator2-backend/check-license . Её достаточно выполнять раз в сутки. Результат команды возвращает JSON, в котором необходимо проверять поле `daysLeft` с условием, что значение параметра не менее 20 дней. Пример JSON:

```
{
    "expiryDate": "2024-12-31",
    "developersUsed": 4,
```

```
    "developersMax": 200,
    "isValid": true,
    "usersMax": 18000,
    "backendBuildNumber": "D-03.123.02_03-RELEASE",
    "daysLeft": 194,
    "usersUsed": 4,
    "dbBuildNumber": "03.123.02"
}
```

### 11.1.3 Проверка срока действия серверного сертификата

Данный сертификат используется для установления SSL сессии между пользователями системы и сервером. После истечения срока действия сертификата, новые подключения будут невозможны, в результате система станет неработоспособна. Проверка выполняется командой keytool на сервере WildFly. Необходимо заблаговременно, не менее чем за 15 дней, инициировать процесс обновления сертификата.

```
keytool -list -v -alias server -keystore путь_к_WildFly/standalone/configuration/
application.jks -storepass 12345678  |  grep until:
```

Результат команды

```
Valid from: Tue Sep 26 01:15:24 MSK 2023 until: Wed Sep 25 01:20:24 MSK 2024
```

### 11.1.4 Проверка используемой памяти Java Runtime

При запуске службы WildFly для java-машины выделяется заданный объём оперативной памяти. Существует две области памяти:

- Java heap space - используется для хранения объектов, с которыми работает приложение. Задаётся параметром `-Xmx`.
- Java non-heap space - используется для хранения кода приложения. Задается различными параметрами `-XX:<Parameter>` (например `-XX:MaxMetaspaceSize`).

Текущее используемое место должно быть в пределах максимального заданного объёма. В противном случае java-машина прерывает свою работу, а сервис WildFly необходимо будет перезапустить. При этом, работа пользователей также будет прервана.

Получить текущее значения использованной памяти можно командой:

```
java -jar путь_к_WildFly/bin/jboss-cli-client.jar --connect --controller="localhost:99
90" --user="wfadmin" --password="wfadmin" --commands="/core-service=platform-mbean/
type=memory:read-resource(include-runtime=true)"
```

В качестве вывода будет JSON, где нужно параметр `used` сравнить с параметром `max` отдельно для каждой области памяти heap и non-heap.

```
{
    "outcome" => "success",
    "result" => {
        "heap-memory-usage" => {
            "init" => 268435456L,
```

```
            "used" => 1476490400L,
            "committed" => 2147483648L,
            "max" => 2147483648L
        },
        "non-heap-memory-usage" => {
            "init" => 2555904L,
            "used" => 440112824L,
            "committed" => 469827584L,
            "max" => -1L
        },
        "object-name" => "java.lang:type=Memory",
        "object-pending-finalization-count" => 0,
        "verbose" => true
    }
}
```

## 11.2 Работоспособность и метрики СУБД

- **Количество подключений.** Должно быть на 20% меньше, чем общее допустимое количество подключений, заданное в файле `пусть_к_postgres\postgres.conf` в параметре `max_connections`. Получить текущее количество подключений можно с помощью запроса `SELECT count(*) FROM pg_stat_activity`
- **Состояние процесса репликации кластера.** Для кластерной конфигурации СУБД необходимо постоянно проверять размер накопленных WAL-файлов на мастер-ноде. Внезапный рост объема говорит о проблемах с процессом репликации (например, отказ слейв-ноды) и потенциально к разрушению кластера. Уточнить параметры настроек кластера можно через команду `patronictl show-config`
- **Объём базы navigator.** При нормальной работе не должно быть внезапного изменения объёма, если не происходит загрузка данных из внешнего источника. Получить текущий объём базы можно командой `select pg_database_size('navigator')`

## 11.3 Метрики сервера по данным операционной системы

На всех серверах, где развёрнуты службы WildFly и Postgres необходимо контролировать следующие основные характеристики:

- **Дисковое пространство.** Критически важно контролировать свободное место на диске где развёрнута СУБД, так как в случае его переполнения служба прерывает работу и перезапустить её будет невозможно. В качестве рекомендации необходимо создать на диске 2-3 пустых файла размером в 5% от общего объема; это позволит оперативно освободить место для перезапуска СУБД и устранения проблемы. Уточнить свободное место можно командой `df -H`. Для службы WildFly требуется контролировать свободное дисковое место в папке с логами `путь_к_wildfly/standalone/logs`.
- **Объем свободной оперативной памяти.** При переполнении оперативной памяти процесс Java-runtime и/или СУБД будут прерваны, что может повлечь потерю данных и/или временный отказ в работе системы. Получить объём текущей используемой памяти можно командой `free`.
- **Объём сетевого траффика.** Существует ситуация, когда на дэшборд выводится значительный объём данных при одновременном увеличении числа пользовательских запросов. Это может привести к проблемам с пропускной способностью сетевого интерфейса, что скажется на быстродействии системы. Рекомендуется отслеживать такую ситуацию, например при помощи команды `iftop`