Навигатор BI

Версия 41

23.09.2026



Руководство по установке

Москва
2026

Навигатор BI: Руководство по установке

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 Общие сведения 4

2 Схемы развертывания 5

2.1 Базовый (минимальный) вариант развертывания 5

2.2 Максимальный вариант развертывания с использованием всех модулей системы 5

3 Системные требования 6

4 Установка 9

4.1 Типовая установка 9

4.2 Особенности установки 24

4.2.1 Установка на ОС Red Hat Enterprise Linux 24

4.2.2 Установка на ОС AstraLinux 24

4.2.3 Распределенная установка 24

4.2.4 Установка в режиме консоли 24

4.2.5 Ручная настройка SSL 25

4.2.6 Настройка планировщика pg_cron 25

4.3 Установка клиентских приложений iOS\Android: 26

5 Обновление 28

6 Откат обновления 29

7 Проверка работоспособности 30

8 Частые вопросы по установке 31

1 Общие сведения

Приложение реализовано на платформе J2EE и имеет модульную архитектуру. Такая архитектура позволяет легко масштабировать приложение под необходимые нужды в случае распределенной установки, а также позволяет реализовать отказоустойчивость в случае установки на кластер. База данных должна быть развёрнута на СУБД Postgres. Клиентом может выступать либо web-браузер, либо мобильное приложение (Android, iOS).

Минимальные системные требования рассчитаны на промышленную эксплуатацию в среде с минимальным набором установленных модулей и с низкой нагрузкой (5-10 конкурентных пользователей).

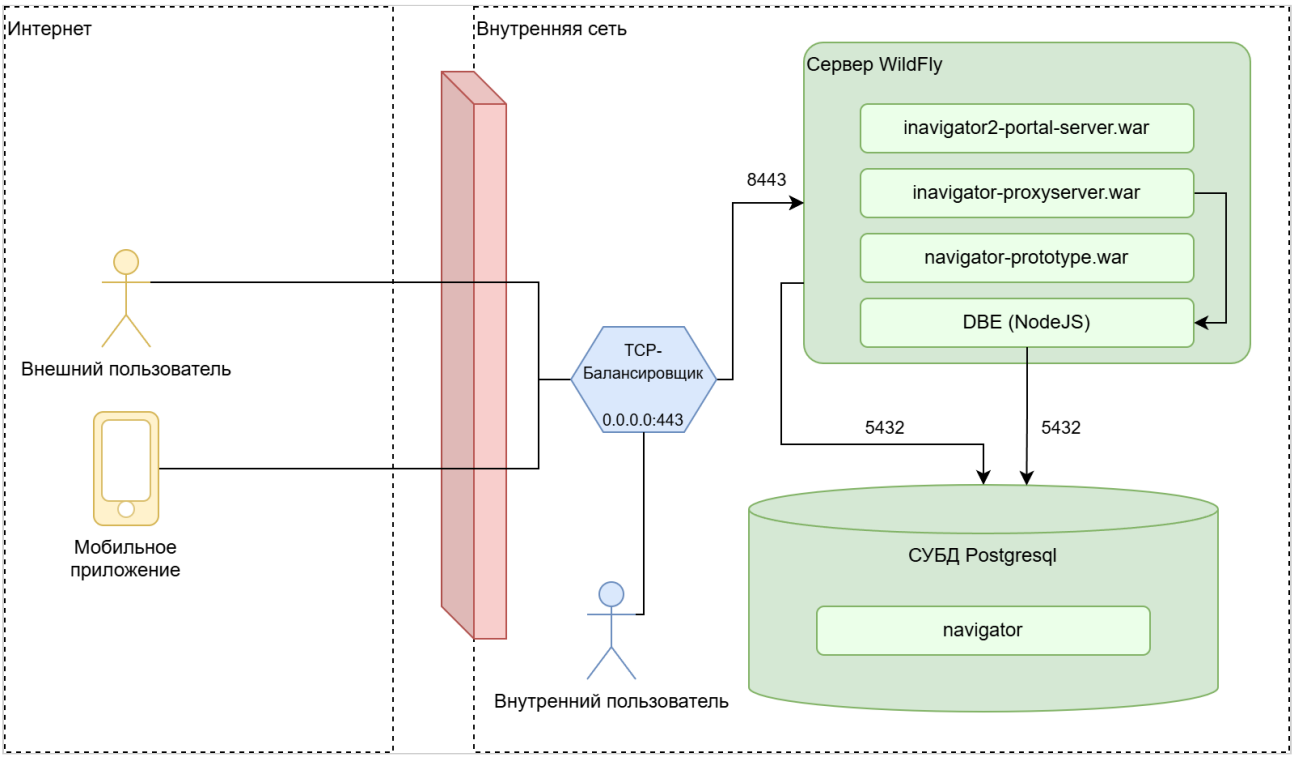
В качестве типовой установки в руководстве подразумевается установка системы из графического инсталлятора в конфигурации standalone (все компоненты на одном сервере) на ОС Ubuntu 24, с установленной Java 17 (openjdk), PostgreSQL 15, WildFly 32.0.1.Full и NodeJS 21.

В системе предусмотрена проверка лицензионного ключа, без которого она не запустится.

Лицензионный ключ представляет из себя файл, который необходимо разместить рядом с установленным Java-приложением (см. п. 1.10 раздела 4.1).

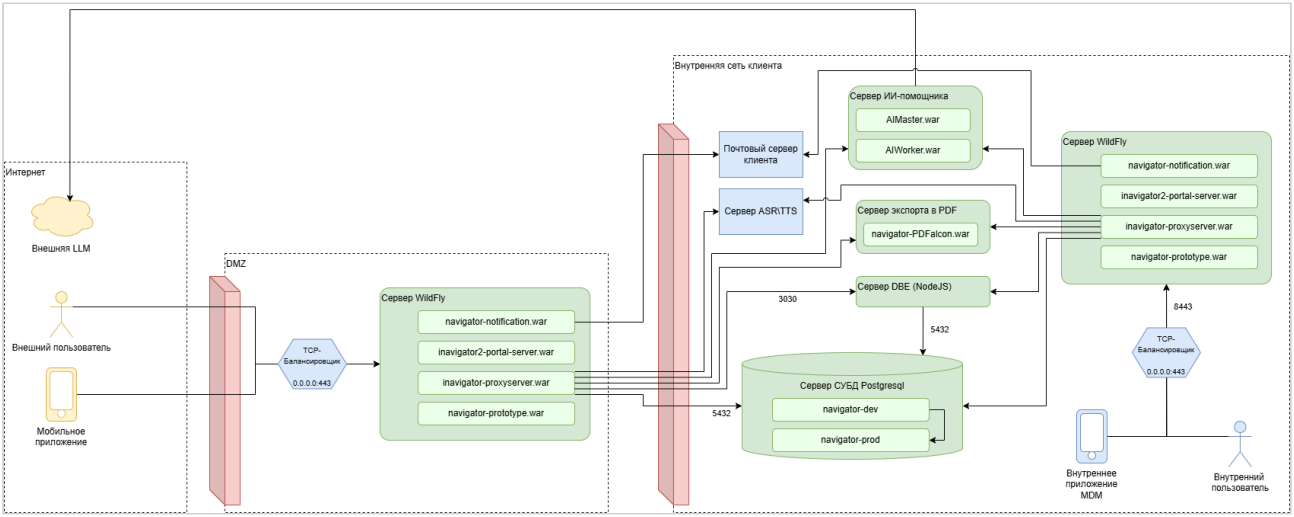
2 Схемы развертывания

2.1 Базовый (минимальный) вариант развертывания



При типовом базовом варианте установки компоненты системы разворачиваются на единственном сервере. Минимальный набор компонентов для развертывания представлен на диаграмме выше.

2.2 Максимальный вариант развертывания с использованием всех модулей системы



При максимальном варианте развертывания каждый логический компонент системы устанавливается на отдельный сервер, обеспечивается наличие двух контуров с отдельным доступом для пользователей внутреннего и внешнего контура.

3 Системные требования

Системные требования для установки серверной части в варианте standalone	
Java Virtual Machine:	Версия 17, x64 Oracle JDK, OpenJDK, AdoptOpenJDK Важно! Прочие версии кроме 17 не поддерживаются
Сервер приложений:	JBoss WildFly версии 32.0.1 Full
Сервер Dashboard Engine	NodeJS 20 и выше
СУБД:	PostgreSQL 15-18, pg_cron
Операционная система:	Red Hat Enterprise Linux 7.9-8.2, Ubuntu 20.04 и выше, AstraLinux SE 1.7
Процессор	X86_64 совместимый процессор, 16 ядер
Объём ОЗУ, Гигабайт	32
Объём ПЗУ, Гигабайт	Минимальный – 10 Рекомендуемый с учётом витрин данных – 500
Объём памяти для Java-машины (Heap space), Гигабайт	2
Объём metaspace на сервере WildFly, Гигабайт	0,5
Системные требования при распределенной установке	
Сервер приложений:	
Java Virtual Machine:	Версия 17, x64 Oracle JDK, OpenJDK, AdoptOpenJDK Важно! Прочие версии кроме 17 не поддерживаются
Сервер приложений:	JBoss WildFly версии 32.0.1 Full
Операционная система:	Red Hat Enterprise Linux 7.9-8.2, Ubuntu 20.04 и выше, AstraLinux SE 1.7
Процессор	X86_64 совместимый процессор, 2 ядра

Системные требования при распределенной установке	
Объём ОЗУ, Гигабайт:	4
Объём ПЗУ, Гигабайт:	10
Объём памяти для Java-машины (Heap space), Гигабайт:	2
Объём metaspace на сервере WildFly, Гигабайт:	0,5
Сервер Dashboard Engine:	
Среда запуска приложений:	NodeJS 20+
Операционная система:	Red Hat Enterprise Linux 7.9-8.2, Ubuntu 20.04 и выше, AstraLinux SE 1.7
Процессор	X86_64 совместимый процессор, 4 ядра
Объём ОЗУ, Гигабайт:	8
Объём ПЗУ, Гигабайт:	20
Сервер СУБД:	
СУБД:	PostgreSQL 15-18, pg_cron
Операционная система:	Red Hat Enterprise Linux 7.9-8.2, Ubuntu 20.04 и выше, AstraLinux SE 1.7
Процессор	X86_64 совместимый процессор, 4 ядра
Объём ОЗУ, Гигабайт:	8
Объём ПЗУ, Гигабайт:	Минимальный – 10 Рекомендуемый с учётом витрин данных – 500 Гб
Системные требования клиентской части	
Приложение iOS:	
Типы устройств:	iPad, iPad mini, iPad Pro, iPhone, iPhone Max

Системные требования клиентской части	
Версия iOS:	16.6 и выше
Приложение Android:	
Типы устройств:	Телефон, планшет
Версия Android:	11 и выше
Web-приложение:	
Имя и версия браузера:	Гарантирована работоспособность на: Google Chrome (версия 130 и выше) Yandex Browser (версия 24 и выше) Safari (версия 18 и выше) Частично поддерживаемые: Firefox (версия 133 и выше) Microsoft Edge (версия 131 и выше) Не поддерживаемые: Internet Explorer
Системные требования инсталлятора	
Java Virtual Machine:	Версия 17, x64 Важно! Прочие версии кроме 17 не поддерживаются

Внимание! Системные требования компонентов PDFalcon и ИИ-помощник указаны в соответствующих руководствах.

4 Установка

4.1 Типовая установка

Под типовой установкой подразумевается установка в Standalone-режиме по схеме «все компоненты на одном сервере» с использованием в качестве основной ОС Ubuntu. Описание других вариантов установки, а также установку на другие ОС, см. в разделе 4.2 «Особенности установки».

1. Перед установкой в обязательном порядке необходимо выполнить следующие действия:

1.1. Установить Java JRE или JDK;

1.2. Установить и настроить сервер WildFly:

1.2.1. Отредактировать файл `путь_к_WildFly/bin/standalone.conf`

- В строке `set JAVA_OPTS` выставить значение `-Xmx=2048M` или больше
- В строке `set JAVA_OPTS` выставить значение `-XX:MaxMetaspaceSize=512M` или больше;

1.2.2. Подключить сервер WildFly к необходимому сетевому интерфейсу, для этого в файле `путь_к_WildFly/standalone/configuration/standalone.xml`:

- в секции `<server> - <interfaces> - <public>`, или в административной консоли WildFly в разделе `Configuration - Interfaces - public - Inet Address` задать необходимый внешний IP адрес вместо 127.0.0.1 либо, если необходимо, чтобы WildFLy слушал все адреса - задать адрес 0.0.0.0;
- В строке

  `<socket-binding name="management-http" interface="management" port="${jboss.management.http.port:9990}"/>`

- Удалить параметр `interface=”management”`, чтобы строка приняла вид

  `<socket-binding name="management-http" port="${jboss.management.http.port:9990}"/>`

1.3. Установить сервер БД Postgres;

1.4. Создать каталоги для табличных пространств (каталоги могут находиться по любому пути, пути ниже даны для примера):

- `mkdir /путь_к_Postgres/18/main/pg_tblspc/navigator_ts`
- `mkdir /путь_к_Postgres/18/main/pg_tblspc/rb_ts`
- `mkdir /путь_к_Postgres/18/main/pg_tblspc/kpi_ts`
- `mkdir /путь_к_Postgres/18/main/pg_tblspc/comm_ts`
- `mkdir /путь_к_Postgres/18/main/pg_tblspc/data_ts`
- `mkdir /путь_к_Postgres/18/main/pg_tblspc/imp_ts`

Внимание! Важно помнить, что права на папку с табличными пространствами в ОС должен иметь только пользователь postgres, всем остальным учётным записям доступ должен быть запрещён. Это ограничение СУБД PostgreSQL, без выполнения которого СУБД не запустится.

1.5. Задать пароль суперпользователя СУБД `postgres`, выполнив команды:

```
sudo passwd postgres (Ввести пароль для системного пользователя postgres)
sudo su postgres
psql
ALTER USER postgres WITH PASSWORD ‘ваш_пароль’;
\quit
```

1.6. Задать права для пользователя postgres на созданные в п. 1.4 каталоги табличных пространств:

```
chmod -R 744 /путь_к_Postgres/18/main/pg_tblspc/
chown -R postgres: /путь_к_Postgres/18/main/pg_tblspc/
```

1.7. Разрешить внешние сетевые доступы:

Отредактировать файл `/var/lib/pgsql/pg_hba.conf`, добавив в секции Local и IPV4 следующие строки:

```
# TYPE      DATABASE        USER            ADDRESS                 METHOD
# LOCAL
local       all             postgres                                trust
# IPV4
host        navigator       navi_app        all                     scram-sha-256
host        navigator       navi_run        127.0.0.1/32            trust
```

В случае необходимости в колонке ADDRESS вместо all указывается IP-адрес сервера, с которого планируется осуществлять доступ к БД.

Внимание! После установки строку

```
local       all             postgres                                trust
```

в целях безопасности необходимо будет удалить либо заменить тип аутентификация для пользователя postgres на другой.

Для пользователя navi_run допускается использовать тип аутентификации peer, в таком случае, необходимо создание локальной учетной записи navi_run, а также отредактировать файл pg_hba: перенести запись в начало файла в секцию LOCAL, изменить TYPE на local и оставить значение ADDRESS пустым:

```
# TYPE      DATABASE        USER            ADDRESS                 METHOD
# LOCAL
local       all             postgres                                peer
local       navigator       navi_run                                peer
# IPV4
host        navigator       navi_app        all                     scram-sha-256
```

Отредактировать файл `/var/lib/pgsql/postgresql.conf`, раскомментировать строку:

```
listen_address='*'
```

1.8. Перезагрузить сервис БД PostgreSQL, убедится, что сервис запустился корректно.

```
systemctl restart postgresql
systemctl status postgresql
```

1.9. Для запуска инсталлятора следует задать разрешение на выполнение для скриптов посредством выполнения команды **chmod**:

```
chmod a+x /Путь к дистрибутиву/*.sh
```

1.10. Перед установкой необходимо разместить файл лицензионного ключа на сервере в папке `путь_к_WildFly/standalone/configuration`, после чего командой

```
chown wildfly: license.key
```

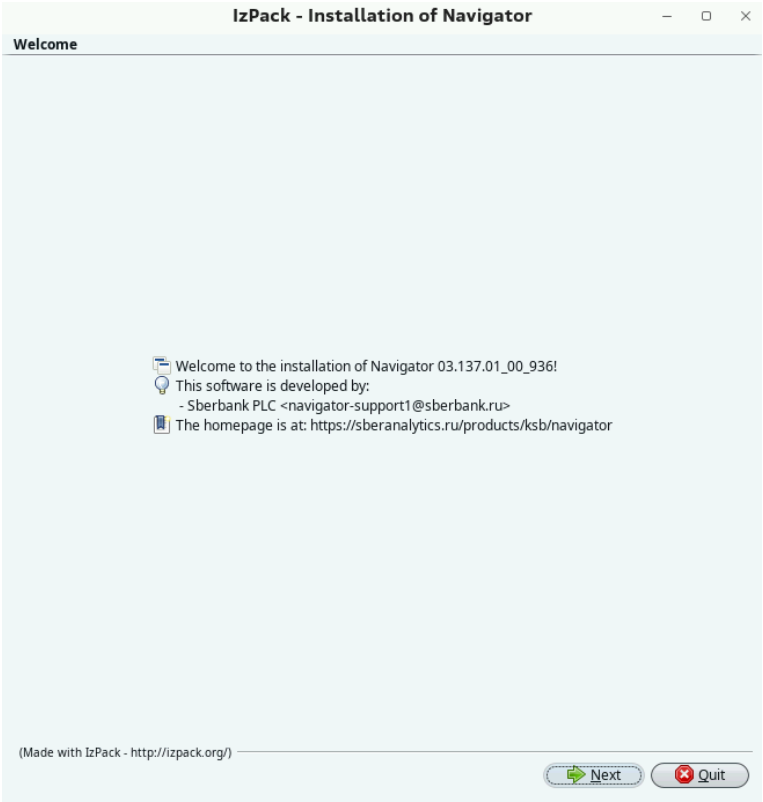
задать права на файл для пользователя **wildfly**.

2. В зависимости от операционной системы и наличия графической оболочки запустить из корня дистрибутива:

- `install.sh` – для Linux;
- `install.cmd` – для Windows (версии до релиза 03.132 включительно);

3. Развёртывание системы выполняется посредством мастера установки. Процедура выполнения скриптов установки запускается на последнем шаге мастера.

Приглашение к установке в графическом режиме выглядит следующим образом:



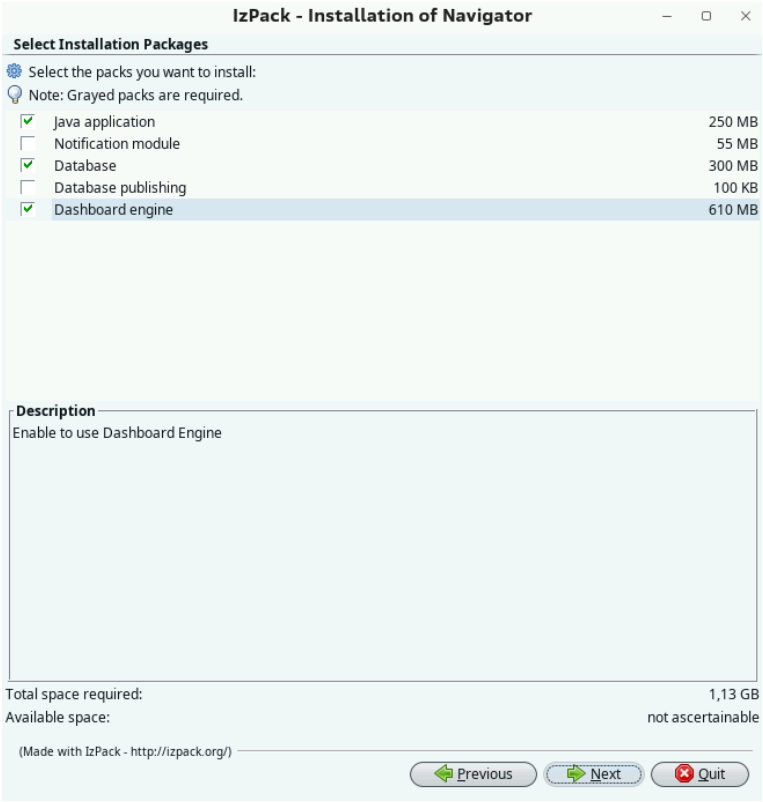
4. Ознакомиться и согласиться с лицензионным соглашением.



5. Выбрать «Install» для первичной установки.



В случае типовой установки следует оставить все выбранные по умолчанию компоненты:



При выполнении развёртывания отказоустойчивого кластера, либо в случае распределенной установки, выбирается тот или иной компонент в зависимости от роли узла кластера или сервера установки (см. подраздел 4.2.4).

Внимание! Если будет использоваться публикация между БД сред, в обязательном порядке необходимо включать чекбокс «Database publishing».

6. Установка приложения производится через WildFly CLI. Для этого требуется задать URL-адрес и логин-пароль администратора web-консоли WildFly.



Если установка выполняется локально (`localhost:9990` в строке «**Console address**»), то будет предложено выбрать папку установки WildFly. Это необходимо для копирования туда конфигурационных файлов приложения. Если установка выполняется на удалённый сервер, то конфигурационные файлы будут сгенерированы в папке инсталлятора с названием "WildFly_standalone_configuration", после чего их нужно будет скопировать на целевой сервер Wildfly в папку `путь_к_WildFly/standalone/configuration`

Внимание! Обновление выполняется только на локальном сервере.

В полях «**Admin username**» и «**Admin password**» нужно указать соответственно логин и пароль администратора WildFly, заданные при установке WildFly.

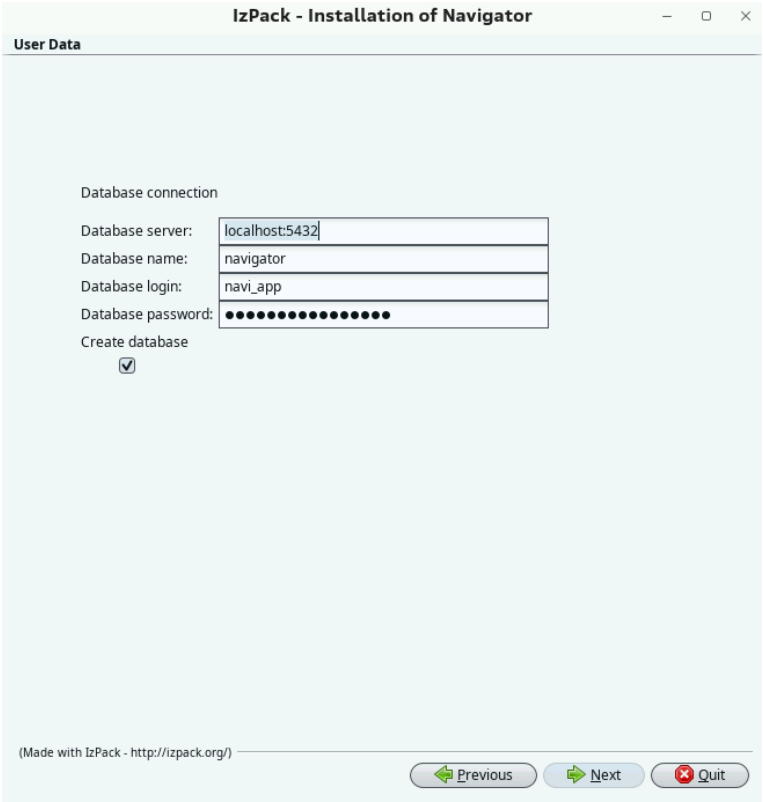
В поле «**Public url after install**» нужно задать тот URL, который будет вводить пользователь в адресной строке браузера, чтобы зайти в систему. Это может быть как имя сервера, так и DNS Alias. Важно задать правильный URL, иначе работа в браузере будет невозможна. Установка по умолчанию использует порт 8443, при необходимости использования стандартного 443 порта требуется перед WIldFly установить промежуточный прокси-сервер (например, nGinx). При необходимости, параметр «**Public url after install**» можно будет скорректировать после установки путем правки конфигурационного файла.

Чекбокс «**Enable Mutual TLS**» необходим для настройки аутентификации по SSL. При стандартной установке с использованием аутентификации по логину-паролю использовать его нет необходимости.

При включении чекбокса в процессе установки формируются файлы application.keystore и truststore.keystore, а также вносятся специфические настройки в standalone.xml. В файле application.keystore формируется SSL-сертификат для сервера с именем, указанным в поле «**Public**

url after install», формируется и сохраняется в хранилище сертификатов WildFly сертификат пользователя Admin, а также в папке установки формируется сертификат navigator-admin.p12, который необходимо установить на клиентском устройстве, с которого будет осуществляться первичный вход в систему.

7. На этом этапе производится задание параметров для создания БД PostgreSQL, выполняемого посредством утилиты flyway (входит в состав дистрибутива Навигатора).



«**Database server**» - Имя/IP адрес и порт сервера БД (по умолчанию – localhost:5432)

«Database name» – название БД (существующей или создаваемой, если стоит галка в чекбоксе «Create Database»).

«Database login» – учётная запись БД, под которой будет работать Java-приложение (по умолчанию – navi_app). Необходимые права на уровне СУБД: владелец базы данных. Если учетная запись отсутствует, она будет создана в ходе установки.

«**Database password**» – задается пароль пользователя БД

«**Create Database**» Создать БД в ходе установки (по умолчанию включено).

Внимание! Крайне не рекомендуется использовать привилегированную учётную запись postgres для работы Java-приложения.

Внимание! Следующее окно появляется только при установленной галке в чекбоксе «Create Database»:



«**Superuser login**» – требуется указать привилегированную учётную запись для создания базы и регистрации необходимых расширений из стандартного комплекта PostgreSQL (pgcrypto, tablefunc) в создаваемой базе Навигатора (по умолчанию - postgres). Эта учетная запись не используется для регулярной работы системы и нужна только при первичной инсталляции.

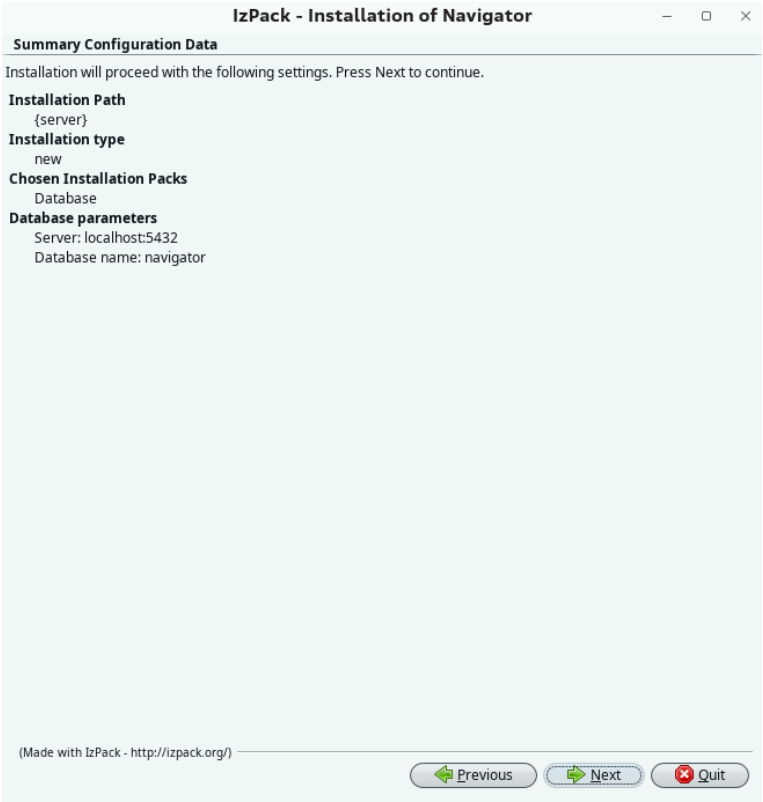
« **Run pg_cron under account:** » – учётная запись БД, под которой будут создаваться задания расширения pg_cron (по умолчанию – navi_run). Необходимые права на уровне СУБД: владелец базы данных. Если учетная запись отсутствует, она будет создана в ходе установки.

Внимание! Если БД предварительно создана и галка «**Create Database**» на предыдущем шаге отключена – необходимо самостоятельно установить расширения pgcrypto, tablefunc.

«**Create users and tablespaces**» – автоматически создать табличные пространства, роли и пользователя «**Database login**». Если это предварительно уже сделано вручную, следует оставить переключатель выключенным.

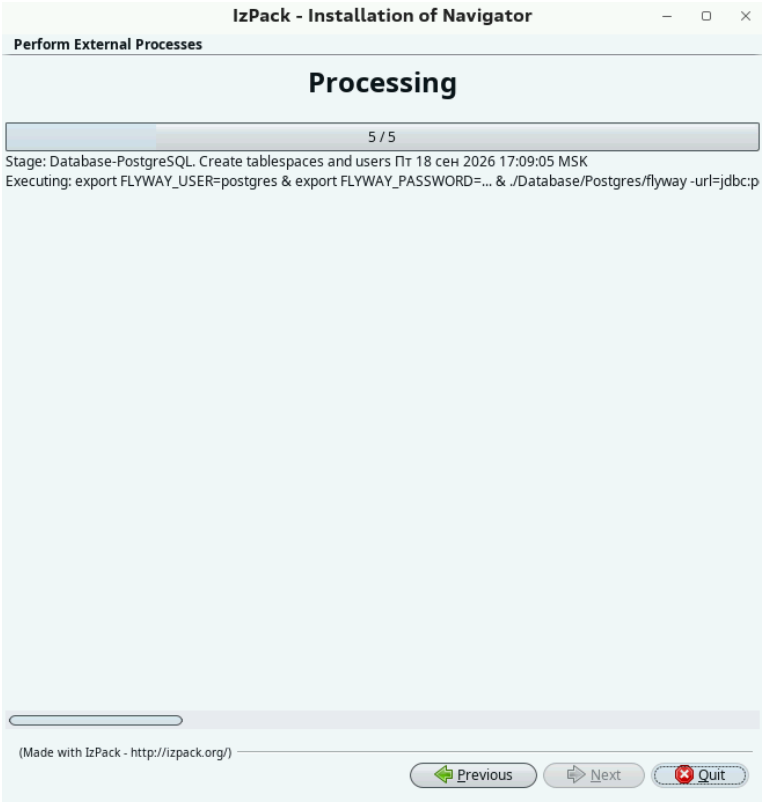
« **Postgres tablespaces directory** » – путь к заранее созданным папкам табличных пространств (см. п. 1.5 раздела 4.1 Руководства).

8. Перед стартом процесса отображается резюме:

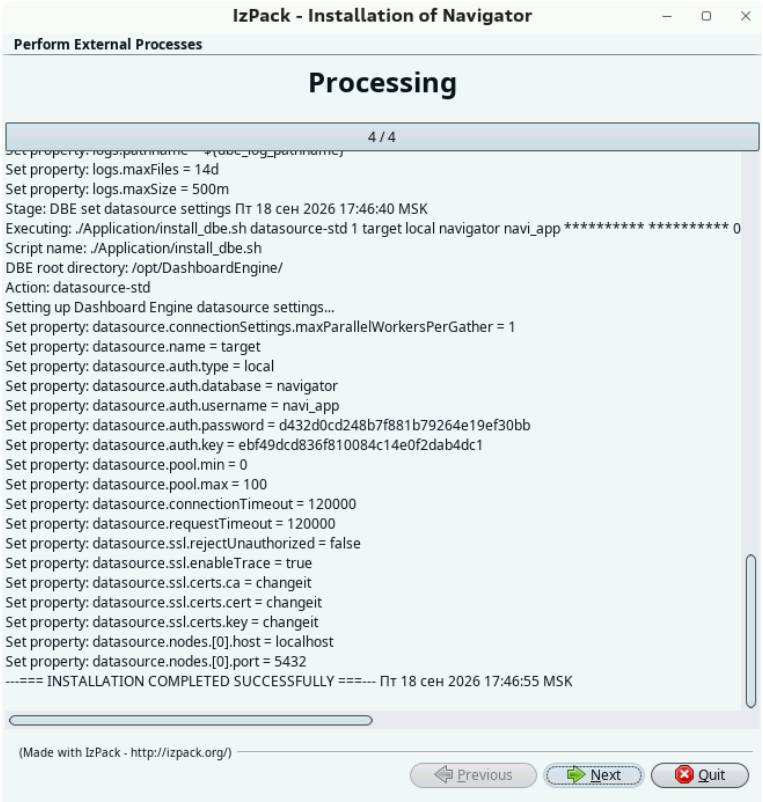


При нажатии кнопки «**Далее**» запустится процесс установки.

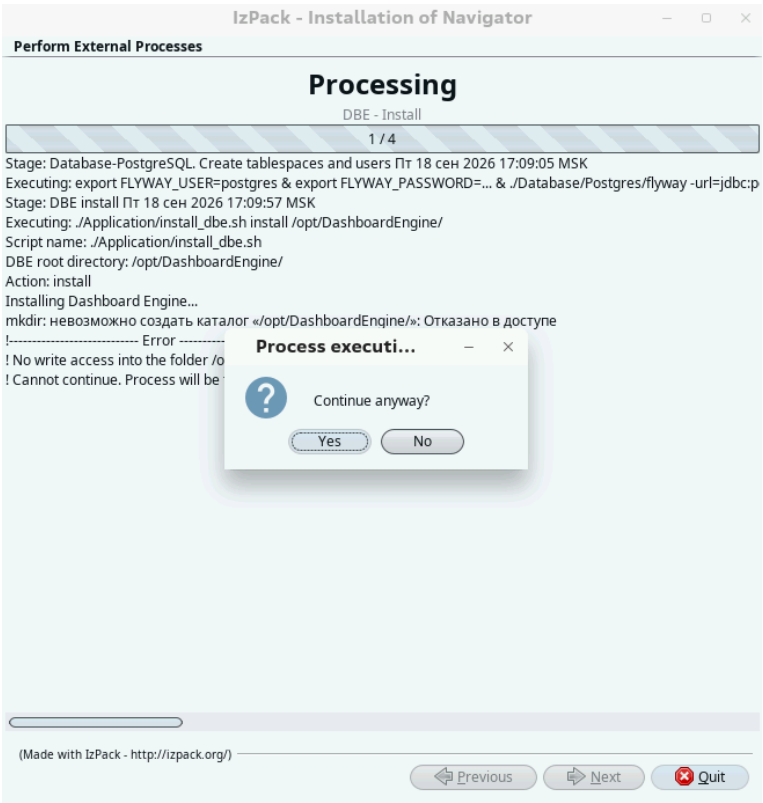
9. Процесс установки выглядит следующим образом:



Окно успешного завершения установки:



10. При возникновении ошибок в процессе установки появляется следующее сообщение:



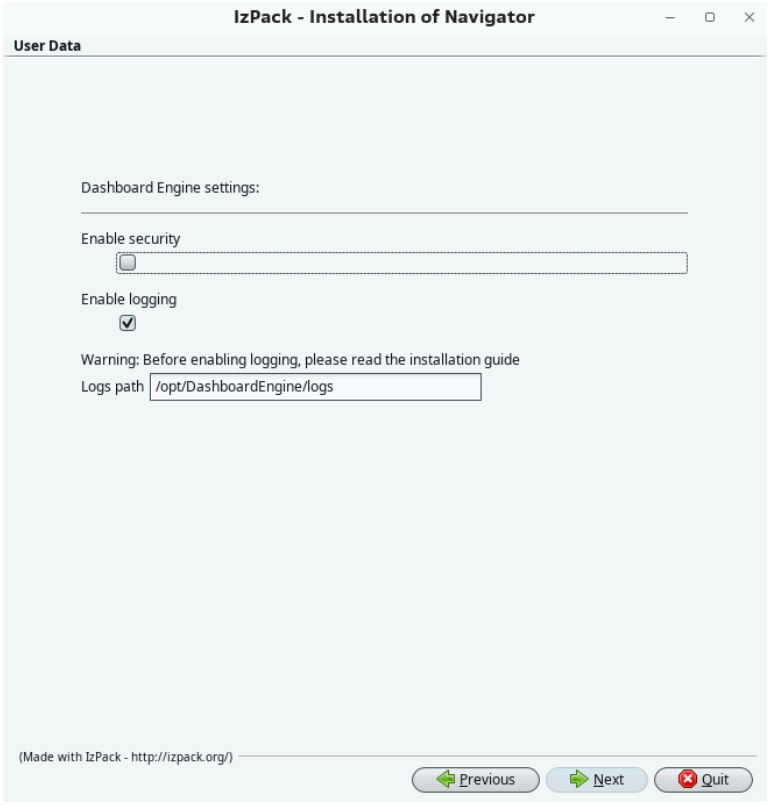
Например, если был указан неправильный пароль подключения к БД, то можно нажать «No», вернуться на предыдущий экран и скорректировать параметры.

11. **Внимание!** Журнал установки сохраняется в корне дистрибутива Навигатора (файл вида `Install_VXX.XXX.XX_YYYYYYYYYYYYYY_Z.log`, где VXX.XXX.X – версия программного изделия, YYY… - дата и время установки, Z – служебный идентификатор). Необходимо передать этот лог разработчику Навигатора в случае возникновения ошибок.

12. Если для установки был выбран компонент Dashboard Engine (DBE), то в процессе установки будет необходимо задать дополнительные параметры:



В полях «**Port**» и «**Host**» нужно указать соответственно порт и ip-адрес, на которых будет размещен компонент DBE.



«**Enable Security**» - по умолчанию выключено, позволяет при установке включить аутентификацию по сертификатам.

Внимание! При включении опции необходимо вручную в файле `путь_к_WildFly/standalone/configuration/DBEConfig.yml` прописать необходимые параметры аутентификации:

DBEConfig.yml

```
# Раздел безопасности
security:
  # Флаг включения режима https и валидации сертификатов
  # Если этот флаг установлен в true, то сервер будет использовать HTTPS и проверять подлинность сертификатов.
  enable: false

  # Флаг для включения отладки tls соединения
  enableTlsTrace: true

  # По умолчанию true
  # Возможность управлять проверкой сертификата, возможно отключить установив в false
  # Тогда остается https но без проверки клиентского сертификата
  checkCert: true

  # Данные сертификатов для защиты доступа к api
  # Вбить необходимые атрибуты которые должны совпадать с сертификатом
  # Структура 'атрибут' : 'значение' || ['значение1', 'значение2']
  # Использование утилиты для чтения атрибутов из сертификата: npm run tls readTlsCert {путь к сертификату}
  allowedDn:
    - O: MyDN
```

```
      OU: 00CA
      CN: CXXX_DEV
    - C: RU
      ST: MSC
      O: MyTestCompany
      OU: TestUnit
      CN: PN
      emailAddress: some@body.ru

  # C=RU, ST=Moscow, L=Moscow, O=MyDN, OU=00CA, CN=SOMEBODY-DEV-VMnodeusr

  # Защиты доступа к метрикам GET /metrics
  # Массив dn-ов, если хотя бы один подходит то авторизация успешна
  allowedDnStat:
    - O: MyDN
      OU: 00CA
      CN: CXXX_DEV
    - C: RU
      ST: MSC
      O: MyTestCompany
      OU: TestUnit
      CN: PN
      emailAddress: some@body.ru

  # В этом разделе указываются пути до файлов сторонних сертификатов,
  # которые будут использованы для проверки подлинности.
  # Возможен один файл с цепочкой сертификатов.
  # Формат сертификата PEM.
  allowedCa:
    - ./somerootca-chain.pem

  # Раздел для настройки собственных сертификатов сервера https
  # В этом разделе указываются пути до файлов ключа и сертификата
  # опциональное поле
  ssl:
    key: ./dev/server.key
    cert: ./dev/server.pem
```

«**Enable Logging**» - Включение логирования (по умолчанию включено).

«**Log path**» - путь для сохранения логов.

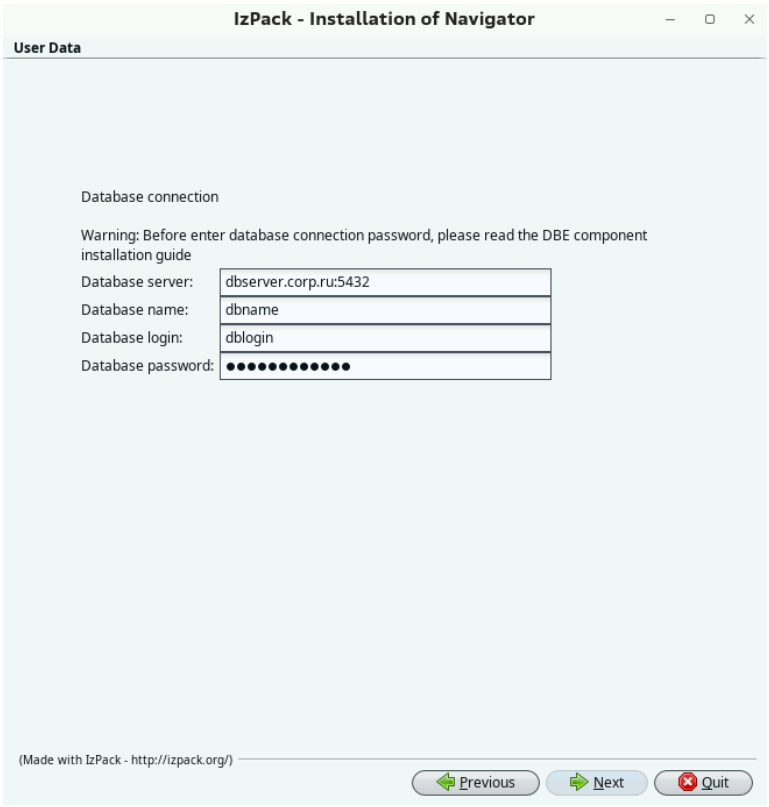
Уровень детализации логирования можно задать в файле `путь_к_WildFly/standalone/configuration/DBEConfig.yml`

DBEConfig.yml

```
# Настройка логов
logs:
  # Включение записи логов в файл
  isWriteMode: true
  # Уровни логирования
  # error: 0,
  # warn: 1,
  # info: 2,
```

```
  # http: 3,
  # verbose: 4,
  # debug: 5,
  # silly: 6
  # Логирование происходит по нисходящей, если warn - то логируется warn и error
  level: debug
  # Настройка абсолютного пути сохранения логов.
  # Дефолтный путь к папке лежит для приложения dashboard-engine в dist/apps/dashboard-
engine
  pathname: "/opt/DashboardEngine/logs"
  # Максимальное кол-во файлов. Может быть также кол-вом дней если есть суффикс d.
  # Сохранение лог файлов происходит с ротацией - по дням или с превышением счетка файлов.
  maxFiles: "14d"
  # Максимальный размер лог файла, при превышении создается новый с суфиксом счетчика.
  maxSize: "500m"
```

В случае, если установка производиться на отдельный сервер (на шаге, описанном в п. 5 раздела 4.1 Руководства не выбраны никакие другие компоненты, кроме DBE), на следующем шаге будет предложено заполнить параметры подключения к БД. Параметры аналогичны приведенным в п. 7 раздела 4.1 Руководства.



Внимание! В процессе установки предусмотрен контроль наличия установленной версии **NodeJS 20+** на сервере. В случае, если ПО **NodeJS** не будет найдено, будет выведено следующее сообщение об ошибке:



Внимание! Компонент устанавливается по жестко заданному пути `/opt/DashboardEngine`. На данную папку должны быть выданы права пользователю, осуществляющему установку. В случае невозможности записи в каталог, будет выведено следующее сообщение об ошибке:



4.2 Особенности установки

4.2.1 Установка на ОС Red Hat Enterprise Linux

Установка на ОС семейства Red Hat как с использованием графической, так и консольной версии инсталлятора, практически не отличается от установки на ОС Debian-семейства Linux. Основные отличия:

- использование иного пакетного менеджера (dnf\yum вместо apt);
- отличающиеся репозитории и пути установки пакетов;
- отличающиеся имена пакетов (pg_cron15 вместо postgresql-15-cron).

4.2.2 Установка на ОС AstraLinux

Установка на ОС AstraLinux отличается тем, что в редакциях «Воронеж» и «Смоленск» вместо СУБД PostgreSQL можно использовать СУБД Tantor соответствующих версий.

4.2.3 Распределенная установка

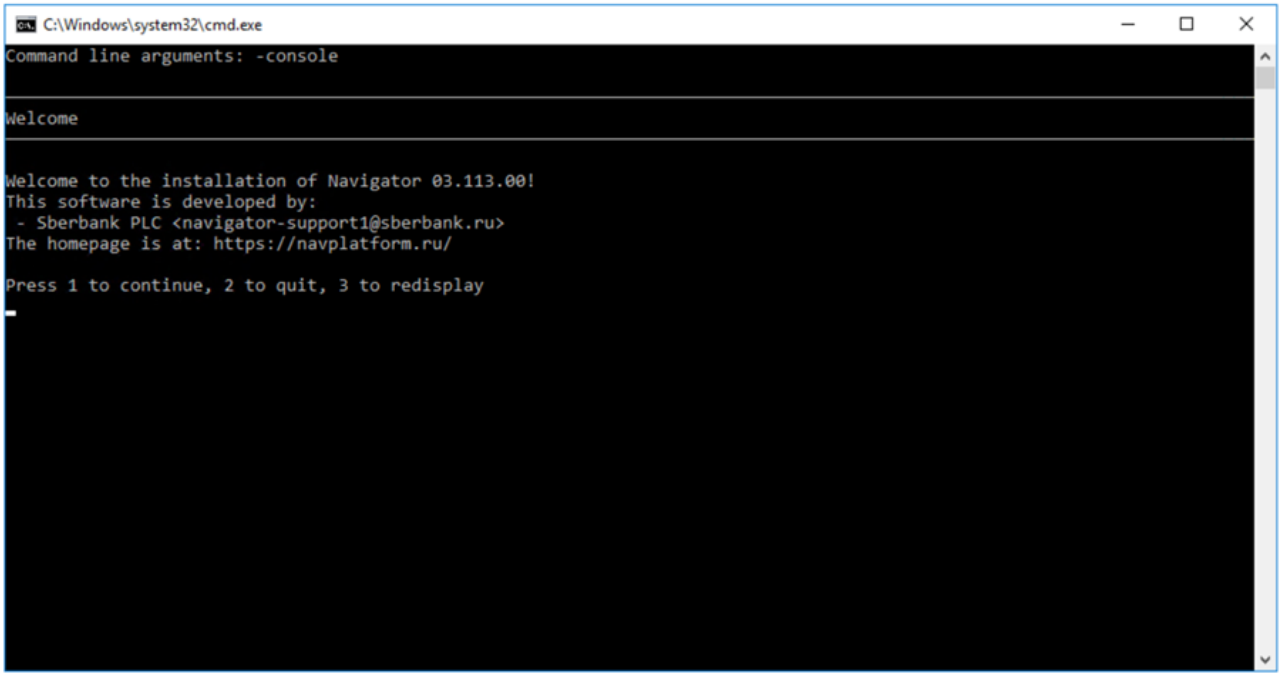
При распределенной установке следует учитывать необходимость обеспечения устойчивого сетевого взаимодействия между узлами системы (сервер БД, сервер приложений WildFly).

Для этого необходимо убедится в доступности портов 5432, 5433, 8443, 9990.

При распределенной установке следует в процессе установки (п. 4 раздела 4.1.) выбрать необходимый для данного сервера компонент (Java application в случае установки на сервер приложений WildFly; Database в случае установки на сервер БД).

4.2.4 Установка в режиме консоли

Спецификой процесса установки является способ ввода ответов пользователя на запросы инсталлятора.



Для выбора вариантов ответа на запрос используются цифры от 1 до 0, при необходимости ответ может вводится текстом с клавиатуры, в таком случае клавиша Enter подтверждает ввод.

Внимание! Крайне рекомендуется не пропускать вопросы инсталлятора, рекомендуется давать ответ на все запросы, даже на те, где предложенный вариант совпадает с необходимым.

4.2.5 Ручная настройка SSL

Внимание! Ручная настройка SSL необходима в случае использования авторизации по сертификату, если при первичной установке не был отмечен чекбокс «Setup additional properties for SSL».

1. Предварительно подготовить два хранилища сертификатов: сертификаты сервера WildFly (`application.jks`) и список доверенных удостоверяющих центров (`truststore.jks`).

   Хранилища можно создать при помощи входящей в состав Java утилиты KeyTool.

   Хранилище `application.jks` должно содержать сертификат и приватный ключ для доменного имени вашего сервера. Например, navigator-domain.my-company.ru. Здесь важно задать alias для сертификата со значением «server», т.к. он связывается по этому имени в настройке далее в п.3.

   Хранилище `truststore.jks` должно содержать один или более публичных сертификатов от удостоверяющих центров, которые выдают личные сертификаты пользователям. Непосредственно личные сертификаты добавлять сюда не требуется.

2. Разместить эти хранилища в папке `путь_к_WildFly/standalone/configuration`

3. Отредактировать файл `путь_к_WildFly/standalone/configuration/standalone.xml` для работы с сертификатами (см. руководство сервера приложений WildFly).

4.2.6 Настройка планировщика pg_cron

Установить планировщик заданий pg_cron;

Произвести необходимые настройки для корректной работы планировщика задач cron, для чего отредактировать в файле `/var/lib/pgsql/postgresql.conf` строки:

```
shared_preload_libraries = 'pg_cron'
cron.database_name = 'navigator'
```

После чего перезагрузить сервис PostgreSQL командой systemctl restart postgresql.

Для регистрации расширений после установки Навигатора выполнить следующие команды:

```
psql -h адрес_сервера_БД -d navigator -U postgres
CREATE EXTENSION pg_cron;
```

Выйти из psql и зайти под пользователем navi_run:

```
psql -h адрес_сервера_БД -d navigator -U navi_run
```

После чего запустить процедуру создания обработчиков внутрисистемных заданий Навигатора:

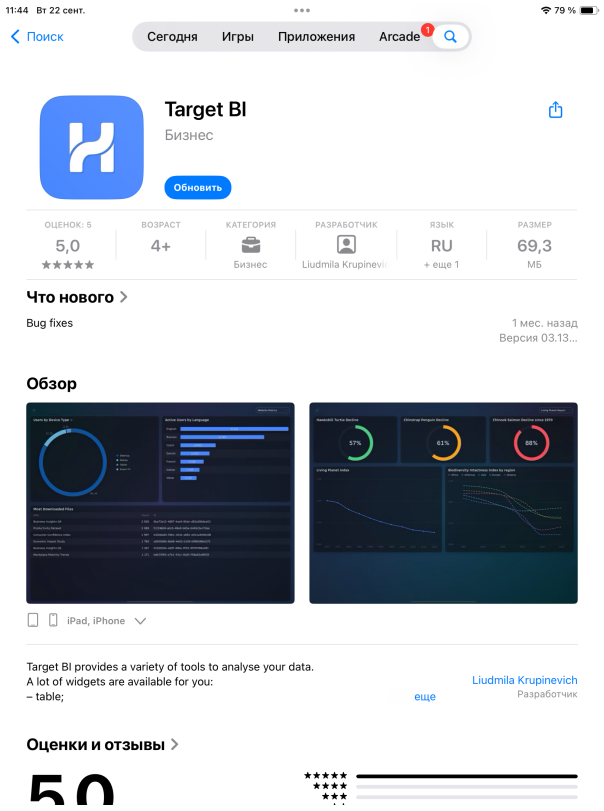
```
SELECT cron.schedule('* * * * *', $$CALL comm.watchdog();$$);
```

Внимание! В случае, если для пользователя navi_run в файле pg_hba.conf указан тип аутентификации peer, необходимо в таблице cron.job в столбце nodename для созданной на предыдущем шаге задачи задать пустое значение (не NULL):

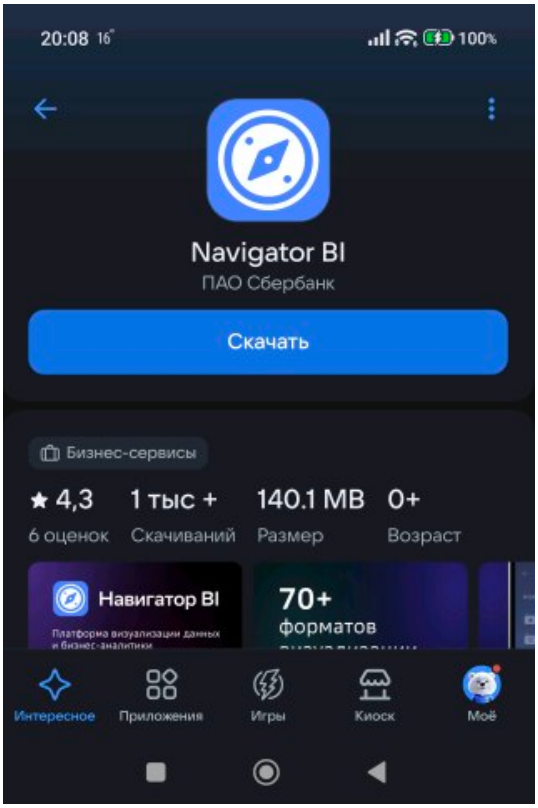
```
UPDATE cron.job SET nodename='';
```

4.3 Установка клиентских приложений iOS\Android:

1. Клиентские приложения для мобильных устройств доступны в соответствующих магазинах приложений:

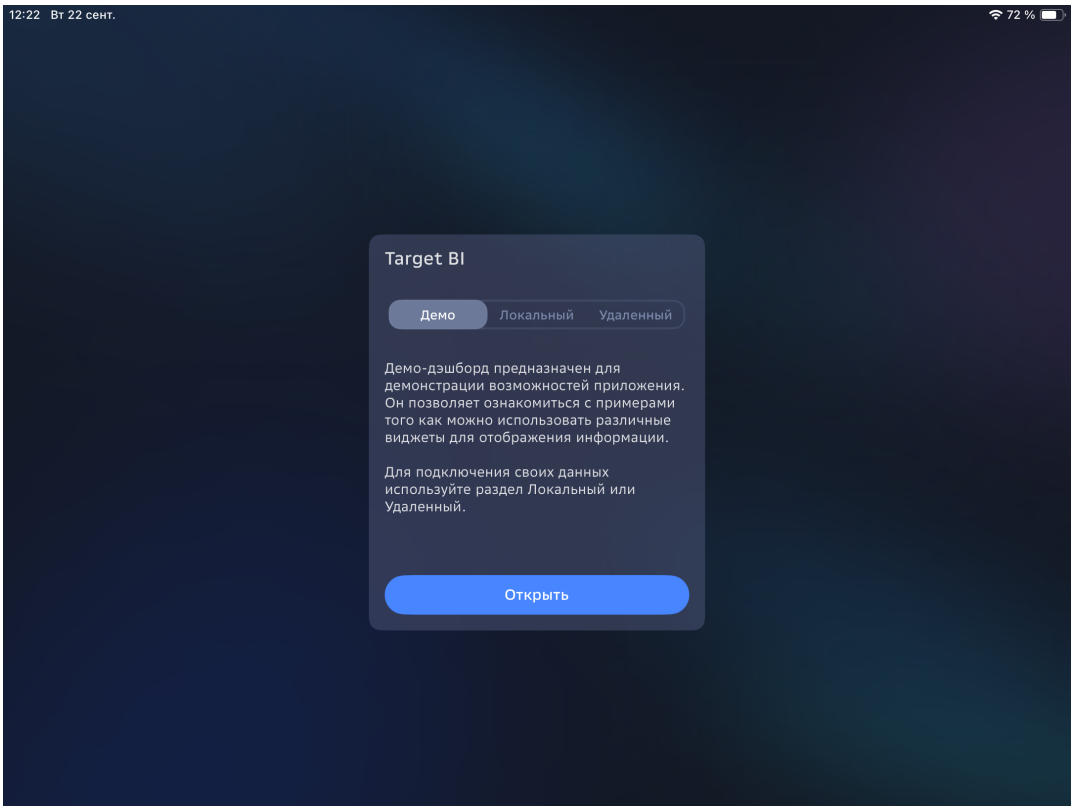


TargetBI в AppStore для iOS



Navigator BI в RuStore для Android.

2. После скачивания и запуска приложения необходимо будет выбрать тип работы:



5 Обновление

Внимание! Перед выполнением обновления необходимо в обязательном порядке создать резервную копию БД (Процедура описана в Руководстве администратора).

Внимание! Перед установкой обновления необходимо отключить выполнение заданий в планировщике (pg_cron или другом используемом планировщике).

Процесс обновления приложения не затрагивает пользовательские данные и настройки. В процессе обновления осуществляется перезапись приложений и системных таблиц в БД Навигатор.

При обновлении необходимо выполнить следующие шаги:

1. Создать резервную копию БД и сервера приложений любыми доступными средствами.

2. Выбрать «**Upgrade »** на этапе выбора типа установки. Выбрать необходимые компоненты для обновления.

3. Ввести параметры соединения с сервером приложений WildFly (в случае обновления приложений), параметры соединения с сервером СУБД (в случае обновления БД)

4. Запустить процесс обновления БД

5. Обновить приложения на клиентских устройствах (iOS, Android).

6 Откат обновления

Откат установленного обновления подразумевает возврат к первоначальной версии программы, установленной до начала обновления. Откат выполняется посредством восстановления программы из резервной копии, заранее созданной в п. 1 раздела 5.

7 Проверка работоспособности

1. Установить сертификат пользователя (при необходимости, если при установке было выбрано включение Mutual TLS):

- Если сертификат был сгенерирован в ходе установки (см. п. 6 раздела 4.1), то он расположен в корне дистрибутива (файл navigator-admin.p12**)**.

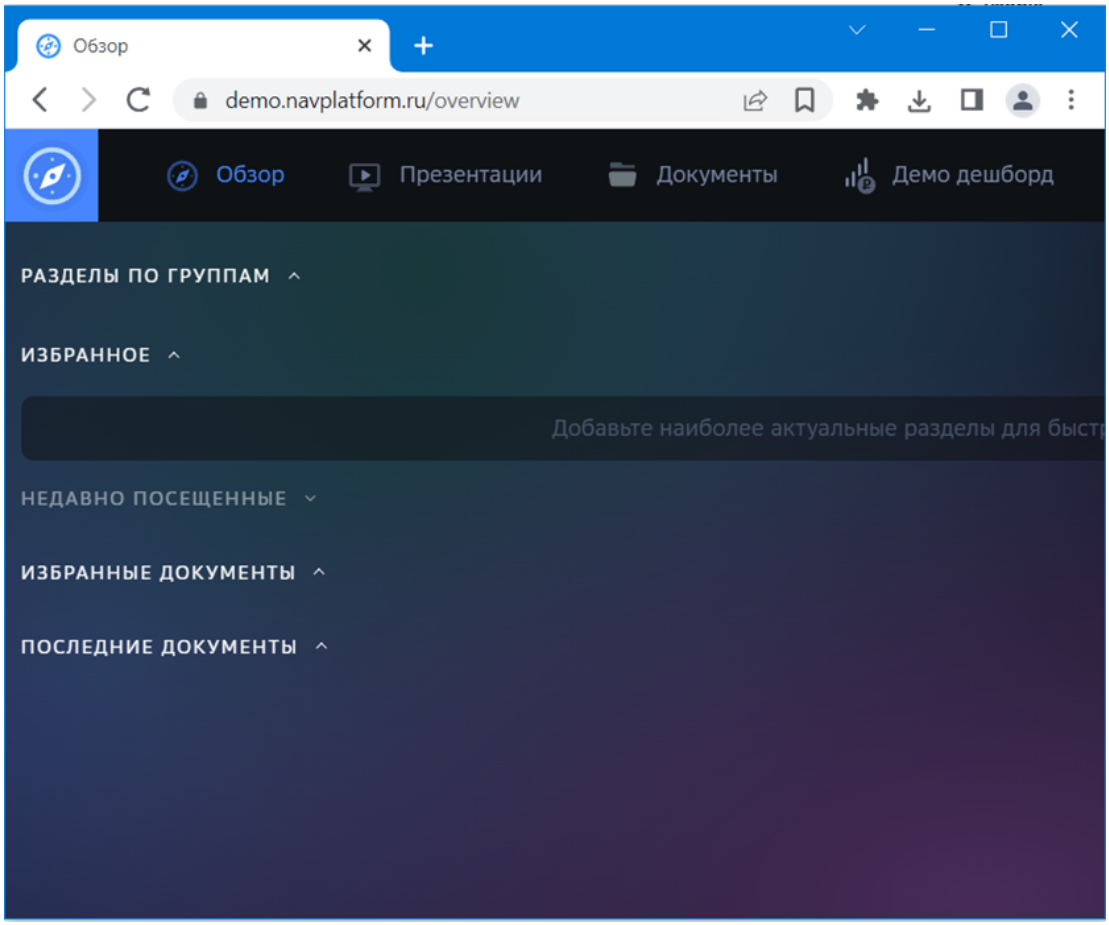
  Пароль сертификата: **12345678**

- Если используется корпоративный удостоверяющий центр, то необходимо подключаться с сертификатом, в котором имеется атрибут

  EMAILADDRESS=admin@navplatform.ru

2. Перейти в браузере на станицу, указанную при установке в поле «**Public url after install**» (см. п. 6 раздела 4.1)

3. Должен появиться стартовый экран Навигатора



4. В случае возникновения проблем журнал приложения можно найти в папке `путь_к_WildFly/standalone/log/` в файле `navigator-portal-server-debug-current.log`

8 Частые вопросы по установке

1. Не запускается скрипт установки:

- Убедиться, что для текущего пользователя заданы права на исполнение файлов скрипта (выполнена команда `chmod a+x /*.sh`);
- Убедиться, что установлены JRE, JDK (выполнить команду `java -version`);
- Запустить установщик вручную командой `javaw -jar installer.jar` для графической версии, `java -jar installer.jar -console` для консольной версии.

2. В процессе установки возникает ошибка «No such file or directory»:

- Убедиться, что для текущего пользователя заданы права на исполнение файлов скрипта (выполнена команда `chmod a+x /*.sh`);
- Запустить установщик от sudo.

3. В процессе установки возникает ошибка « `Каталог …имя_каталога_ts не существует` »:

- Убедиться, что созданы каталоги для размещения табличных пространств;
- Убедится, что путь, указанный в процессе установки (см. п. 7 раздела 4.1 Руководства по установке) совпадает с фактическим местонахождением каталогов;
- Убедиться, что у пользователя postgres есть права на каталоги (выполнив команду `ls -l`);
- Запустить установщик от sudo.

4. В процессе запуска в браузере отображается пустой экран:

- Убедитесь, что в браузере указан тот адрес\IP, который был указан в процессе установки (см. п. 6 раздела 4.1 Руководства по установке);
- При необходимости, изменить адрес\IP (параметр `reactAppApiUrl` в файле `путь_к_WildFly/standalone/configuration/navigator-frontend-props.yml`).

5. В браузере отображается ошибка «Ошибка доступа: не удалось установить соединение с сервером»:

- Проверить настройки параметра `listen_address='*'` в файле `/etc/postgresql/postgresql.conf`
- Проверить, запущен ли сервис PostgreSQL:

```
systemctl status postgresql.service
```

6. В браузере при запуске системы появляется сообщение «Лицензионный ключ не найден»:

- Убедиться, что файл лицензионного ключа скопирован в папку `путь_к_WildFly/standalone/configuration`;
- Убедиться, что файл называется `license.key` и имеет не нулевой размер.