

БФТ.ПЛАТФОРМА

Инструкция по скачиванию, установке экземпляра программного обеспечения и запуску Системы «БФТ.ПЛАТФОРМА» на RedOS

Листов 31

© 2022 ООО «БФТ»

СОДЕРЖАНИЕ

1. Скачивание дистрибутива.....	3
2. Установка системы «БФТ.ПЛАТФОРМА» на РЕД ОС	4
2.1.Установка шрифтов	5
2.2.Установка OpenJRE	6
2.3.Установка PostgreSQL	7
2.4.Установка и настройка Tomcat.....	10
3. Запуск системы «БФТ.ПЛАТФОРМА».....	15
4. Просмотр данных примера	16
Приложение 1	24

1. Скачивание дистрибутива

Для получения дистрибутива¹ необходимо пройти по ссылке, ввести пароль и скачать файл («app.war») и файл базы данных с примером (с расширением .dump) из облачного хранилища.

Ссылка: <https://cloud.mail.ru/public/2ZYk/5uXAcoeTS>

¹ Пример настройки системы на базе «БФТ.ПЛАТФОРМА»

2. Установка системы «БФТ.ПЛАТФОРМА» на РЕД ОС

Порядок установки:

1. Установка шрифтов.
2. Установка OpenJRE;
3. Установка PostgreSQL;
4. Создание базы приложения;
5. Установка и настройка Tomcat;
6. Настройка портов брандмауэра.

2.1. Установка шрифтов

Для установки шрифтов необходимо выполнить:

```
sudo yum -y install ./Fonts/msttcore-fonts-2.0-2.noarch.rpm
```

Примечание:

- Без установки шрифта формирование отчета в PDF осуществляется некорректно.
- Необходимо указать полный путь, где находится распакованный дистрибутив из раздела **Ошибка! Источник ссылки не найден..**
REF _Ref49176940 \h * MERGEFORMAT **Ошибка! Источник ссылки не найден..**

2.2. Установка OpenJRE

Для установки OpenJRE (из дистрибутива) необходимо выполнить следующий запрос:

```
yum install java-11-openjdk
```

2.3. Установка PostgreSQL

1. Устанавливаем PostgreSQL 12 версии

```
sudo yum -y install postgresql-server postgresql-contrib
```

2. Инициализируем базу

```
sudo postgresql-setup initdb
```

Разрешаем автозапуск службы postgres

```
sudo systemctl enable postgresql
```

3. Вносим конфигурацию доступа для будущей базы

3.1 сохраняем копию оригинального файла настроек

```
sudo mv /var/lib/pgsql/data/pg_hba.conf /var/lib/pgsql/data/pg_hba.conf_original
```

3.2 Вносим изменения

```
echo "# TYPE DATABASE USER ADDRESS METHOD" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# local is for Unix domain socket connections only" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo " local all all peer" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# IPv4 local connections:" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# host all all 127.0.0.1/32 ident" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# IPv6 local connections:" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# host all all ::1/128 ident" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# Allow replication connections from localhost, by a user with the" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "# replication privilege." | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "local replication all peer" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "host replication all 127.0.0.1/32 ident" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "host replication all ::1/128 ident" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "host db_ice_starter_140 u_starter 127.0.0.1/32 md5" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "host db_ice_starter_140 u_starter ::1/128 md5" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
echo "host all all 0.0.0.0 reject" | sudo tee -a /var/lib/pgsql/data/pg_hba.conf
```

Или используя любой удобный редактор приводим файл к виду:

```
nano /var/lib/pgsql/data/pg_hba.conf
```

```
# TYPE DATABASE USER ADDRESS METHOD
# local is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
# host all all 127.0.0.1/32 ident
# IPv6 local connections:
# host all all ::1/128 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 ident
host replication all ::1/128 ident
host db_ice_starter_140 u_starter 127.0.0.1/32 md5
host db_ice_starter_140 u_starter ::1/128 md5
host all all 0.0.0.0/0 reject
```

Изменим владельца файла

```
sudo chown postgres:postgres /var/lib/pgsql/data/pg_hba.conf
```

Примечание: Данная конфигурация не претендует на полную безопасность сервера баз данных и применяется, только если в этом есть необходимость для неопытного системного администратора, но все же предоставляет должную надежность для работы стенда.

4. Запускаем сервис

```
sudo systemctl start postgresql
```

5. Создание пользователя и базы данных

5.1 Создаем пользователя базы данных

```
sudo -u postgres psql -c "CREATE USER u_starter WITH LOGIN SUPERUSER
PASSWORD 'Pa$$WordDb';"
```

5.2 Создаем базу данных

```
sudo -u postgres psql -c "CREATE DATABASE db_starter OWNER u_starter;;"
```

5.3 Применяем роль

```
sudo -u postgres psql -c "GRANT postgres TO u_starter;"
```

5.4 Создаем технического пользователя ice_user без возможности входа

```
sudo -u postgres psql -c "CREATE USER ice_user WITH NOLOGIN;"
```

5.5 Применяем права для пользователя ice_user

```
sudo -u postgres psql -c "GRANT ice_user TO u_starter;"
```

Для развертывания нового экземпляра данных действий достаточно.

Для разворачивания копии данных необходимо выполнить восстановление дампа (запросит пароль):

```
pg_restore -h localhost -U u_starter -F c -d db_ice_starter_140  
./BaseDump/db_ice_starter_140.tar.gz
```

Примечание: Необходимо указать полный путь, где находится распакованный дистрибутив из раздела 1. Скачивание дистрибутива.

2.4. Установка и настройка Tomcat

1. Создаем папку для работы приложения

```
sudo mkdir -p /opt/_Tomcat/starter-8080
sudo tar xvf ./Tomcat/apache-tomcat-9.0*tar.gz -C /opt/_Tomcat/starter-8080 --strip-components=1
```

Примечание: Необходимо указать полный путь, где находится распакованный дистрибутив из раздела 1. Скачивание дистрибутива.

2. Создаем пользователя и группу для Tomcat

```
sudo groupadd tomcat
sudo useradd -M -s /bin/nologin -g tomcat -d /opt/_Tomcat tomcat
```

3. Устанавливаем права на созданную папку

```
sudo chown -R tomcat:tomcat /opt/_Tomcat
sudo find /opt/_Tomcat/* -type f -exec chmod 660 {} \;
sudo find /opt/_Tomcat/* -type d -exec chmod 770 {} \;
sudo find /opt/_Tomcat/* -type f -name *.sh -exec chmod 770 {} \;
```

4. Создаём каталоги для журналов:

```
sudo mkdir -p /var/log/tomcat/starter-8080/archiv
sudo rmdir /opt/_Tomcat/starter-8080/logs
sudo ln -s /var/log/tomcat /logs
sudo ln -s /var/log/tomcat/starter-8080 /opt/_Tomcat/starter-8080/logs
sudo chmod -R 770 /var/log/tomcat
sudo chown -R tomcat:tomcat /var/log/tomcat
sudo restorecon -Rv /var/log/tomcat
```

5. В файл <ТМС>/conf/context.xml перед закрывающим тегом </Context> добавим строку:

```
sudo vi /opt/_Tomcat/starter-8080/conf/context.xml
-----
...
<Resources cachingAllowed="true" cacheMaxSize="100000" cacheTtl="2000" />
</Context>
```

Команды:

1. `sudo vi /opt/_Tomcat/starter-8080/conf/context.xml`
2. Нажать «i»
3. Вставить скопированный текст через правую кнопку мыши
4. Нажать «Esc»
5. Сохранить файл путем нажатия «:wq»

6. Создадим файлы systemd-юнитов для запуска Tomcat-ов в качестве сервиса:

```
sudo vi /etc/systemd/system/tomcat-starter-8080.service
```

```
-----Начало скрипта -----
# Systemd unit file for Tomcat - Application1
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target

[Service]
Type=forking

###-- Каталог установки JDK
### OpenJDK - JRE
# Environment='JAVA_HOME=/usr/lib/jvm/jre'

Environment='JAVA_HOME=/usr/lib/jvm/jre-11

#Environment='JAVA_HOME=/usr/lib/jvm/java-1.8.0'
### OracleJDK - JRE
# Environment='JAVA_HOME=/usr/java/latest/jre'

Environment='CATALINA_PID=/opt/_Tomcat/starter-8080/temp/tomcat.pid'
Environment='CATALINA_HOME=/opt/_Tomcat/starter-8080'
Environment='CATALINA_BASE=/opt/_Tomcat/starter-8080'
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

WorkingDirectory=/opt/_Tomcat/starter-8080
ExecStart=/opt/_Tomcat/starter-8080/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

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

Команды:

1. `sudo vi /etc/systemd/system/tomcat-starter-8080.service`
2. Нажать «i»
3. Вставить скопированный текст через правую кнопку мыши
4. Нажать «Esc»
5. Сохранить файл путем нажатия «:wq»

7. Для запуска приложения необходимо дописать настройки приложения в файле

```
sudo vi /opt/_Tomcat/starter-8080/conf/catalina.properties
```

в конец файла добавить

```
-----текст файла catalina.properties -----
#unix: -Djline.terminal=org.springframework.shell.core.IdeTerminal
#windows: -Djline.WindowsTerminal.directConsole=false
Djline.terminal=jline.UnsupportedTerminal
dm.fieldSetPackages=com.bftcom.ice.warehouse.common,com.bftcom.configurator.common,com.bftcom.reports.common,\
    com.bftcom.bpm.common,com.bftcom.ice.common,com.bftcom.icewidgets.common
ice.servicesPackages=com.bftcom.ice.warehouse.common.service,com.bftcom.configurator.common.services,com.bftcom.ice.common.service
ice.rls.enabled = true
ice.script.jvmPackages= com.bftcom.ice.common.maps.*,\
    com.bftcom.ice.server.util.*,\
    com.bftcom.ice.common.general.*,\
    com.bftcom.ice.common.service.notification.*,\
    com.bftcom.ice.common.service.notification.NotificationApi.*,\
    com.bftcom.ice.common.utils.*,\
    org.camunda.bpm.engine.impl.persistence.entity.*,\
    com.bftcom.bpm.server.util.*,\
    com.bftcom.bpm.*

spring.profiles.active=postgresql
spring.datasource.url=jdbc:postgresql://localhost:5432/db_ice_starter_140
spring.datasource.username=u_starter
spring.datasource.password=Pa$$WordDb
#spring.devtools.restart.trigger-file = application.properties
spring.devtools.restart.enabled = false
spring.devtools.livereload.enabled = false
server.compression.enabled=true
server.compression.min-response-size=2048
server.compression.mime-types=application/pdf,application/json,application/xml,text/html,application/javascript,text/css,text/xml,text/plain
spring.servlet.multipart.max-file-size=32MB
spring.servlet.multipart.max-request-size=32MB
server.servlet.contextPath=/app
# Charset of HTTP requests and responses. Added to the "Content-Type" header if not set
```

```
explicitly.  
spring.http.encoding.charset=UTF-8  
# Enable http encoding support.  
spring.http.encoding.enabled=true  
# Force the encoding to the configured charset on HTTP requests and responses.  
spring.http.encoding.force=true  
spring.jackson.serialization.write-dates-as-timestamps=true  
spring.quartz.job-store-type=jdbc  
spring.quartz.jdbc.initialize-schema=never  
spring.quartz.properties.org.quartz.jobStore.tablePrefix=quartz.qrtz_  
spring.quartz.properties.org.quartz.jobStore.driverDelegateClass           =  
org.quartz.impl.jdbcjobstore.PostgreSQLDelegate  
spring.shell.interactive.enabled=false  
log.access.enabled=false  
log.access.httpRequests.maxPayloadLength=1000  
log.access.httpRequests.includePayload=true  
camunda.bpm.application.delete-upon-undeploy=true  
  
-----Конец файла -----
```

Команды:

1. `sudo vi /opt/_Tomcat/starter-8080/conf/catalina.properties`
2. Нажать «i»
3. Вставить скопированный текст через правую кнопку мыши
4. Нажать «Esc»
5. Сохранить файл путем нажатия «:wq»

Полное описание настроек `application.properties` приведено в Приложении.

8. Перечитаем сервисы

```
sudo systemctl daemon-reload
```

Разрешаем автозапуск Tomcat

```
sudo systemctl enable tomcat-starter-8080.service
```

9. Настройка портов брандмауэра

Для возможности подключения к серверу с других машин в сети на данной машине необходимо разрешить порту приложения (порт приложения указан в файле `server.xml` в поддиректории Tomcat'a `conf` – в параметре `port` тега `Connector`) принимать запросы извне.

Разрешаем порты 8080 Tomcat, 5432 postgres.

```
sudo vi /etc/sysconfig/iptables
```

```
# Tomcat
```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT
```

```
# Postgresql 10.6
```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 5432 -j ACCEPT
```

Команды:

1. `sudo vi /etc/sysconfig/iptables`
2. Нажать «i»
3. Вставить скопированный текст через правую кнопку мыши
4. Нажать «Esc»
5. Сохранить файл путем нажатия «:wq»

10. Запуск сервиса

```
sudo systemctl start tomcat-starter-8080
```

```
cp ./Application/app.war /opt/_Tomcat/starter-8080/webapps/
```

```
sudo chown tomcat:tomcat /opt/_Tomcat/starter-8080/webapps/app.war
```

Примечание: Необходимо указать полный путь вместо `./Application/`, где находится распакованный дистрибутив из раздела 1. Скачивание дистрибутива

3. Запуск системы «БФТ.ПЛАТФОРМА»

Работа в платформе доступна только для зарегистрированных пользователей.

Для перехода к окну авторизации в строке адреса браузера вводится адрес сервера системы. Откроется окно авторизации пользователя:

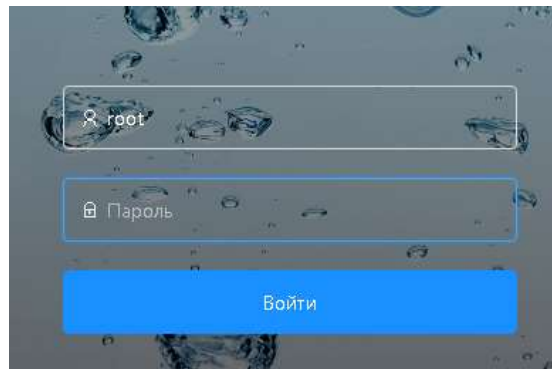


Рисунок 1 – Окно входа в систему

Для входа в систему необходимо указать следующую информацию:

- **Логин:** «**root**» - имя пользователя.
- **Пароль:** «**root**» - пароль пользователя.

Далее нажимается кнопка **Войти**.

В случае ввода верной информации (указаны зарегистрированный в системе пользователь и правильный пароль) осуществится вход в главное окно системы. При вводе неверного **Пользователя** или **Пароля** на экране появится сообщение об ошибке.

4. Просмотр данных примера

Функции платформы можно рассмотреть на примере процесса сбора данных по заболеваемости коронавирусом с муниципальных районов Тюменской области.

Требования:

- необходимо организовать сбор данных со стороны субъекта по каждому муниципальному району на выбранную дату в составе показателей:
 - заболело коронавирусом;
 - умерло от коронавируса;
 - выздоровело после заболевания коронавирусом.
- данные должны проходить процедуру согласования на уровне муниципального района и субъекта.
- должна быть возможность в процессе получить отчет по собранным показателям с возможностью сохранения в различных форматах (doc, xls, pdf);
- должна быть возможность настроить аналитические представления собранных данных в виде графиков и таблиц в разрезе дат, муниципальных районов, показателей «Заболели», «Умерли», «Выздоровели».

Сущности примера:

- справочник «Муниципальные районы»;
- справочник «Субъекты»;
- bpm-процессы сбора данных о количестве заболевших коронавирусом:
 - главный процесс «Формирование сводного отчета о количестве заболевших коронавирусом», в т.ч. объект приложения с контекстом процесса;

- вложенный процесс «Сбор данных с муниципальных районов по заболеваемости коронавирусом», в т.ч. объект приложения с контекстом процесса.
- отчет о количестве заболевших коронавирусом (таблица с диаграммой);
- объект хранилища для аналитических представлений (виджетов) «Результаты по коронавирусу»;
- настроенные аналитические представления на рабочей панели «Данные по коронавирусу».

Справочники и другие объекты с данными можно посмотреть по пункту меню «Справочники»:

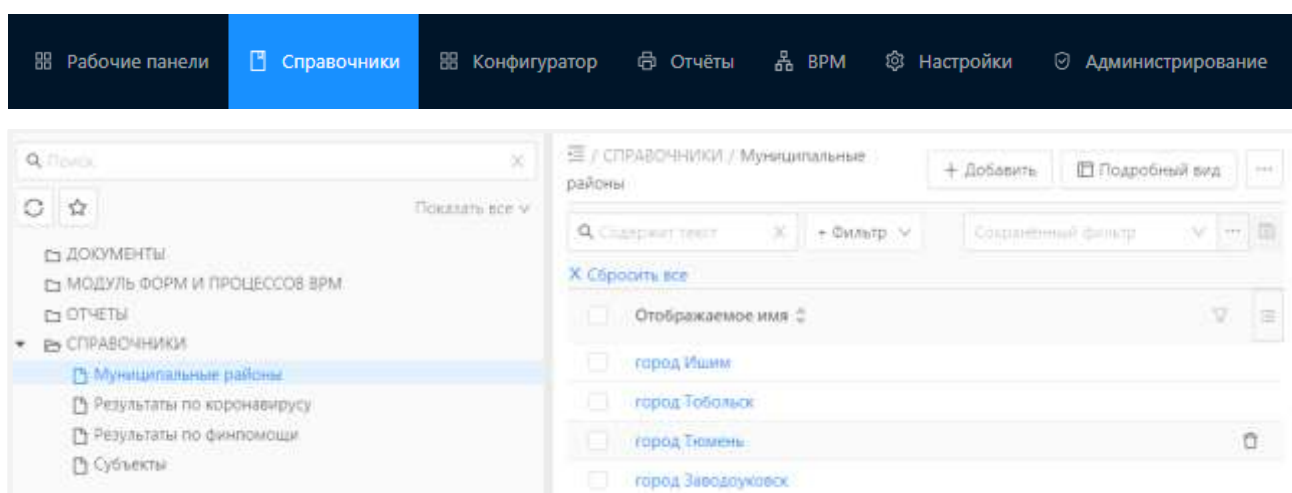
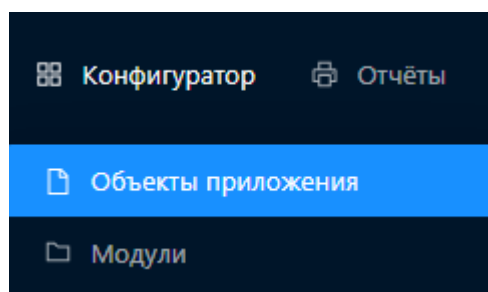


Рисунок 2 – Справочник «Муниципальные районы» (вид для пользователя)

Конфигурации (настройку атрибутов, форм, валидации) можно посмотреть по пункту меню «Конфигуратор → Объекты приложения»:



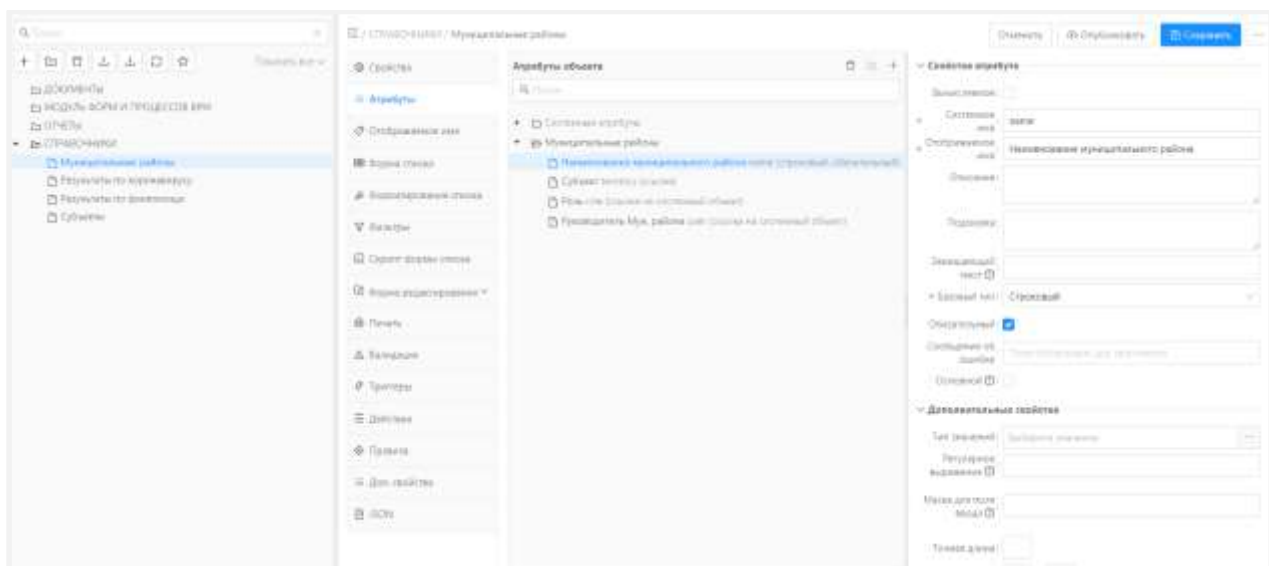
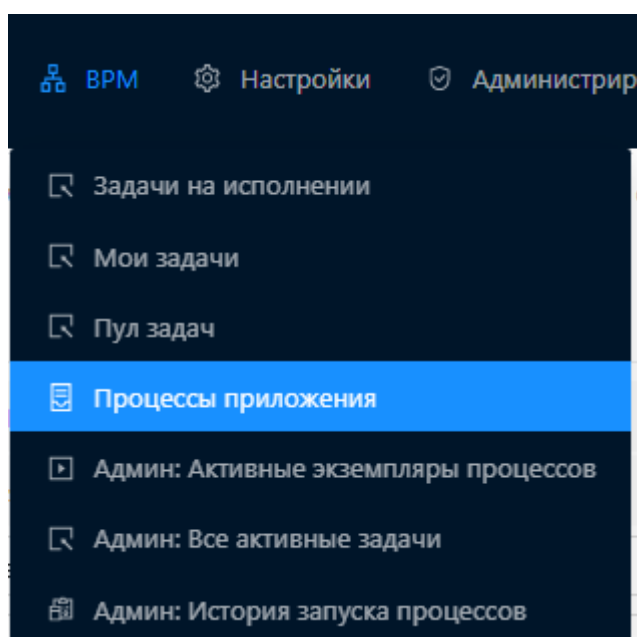


Рисунок 3 – Справочник «Муниципальные районы» (настройка конфигурации: Конфигуратор/Объекты приложения/Справочники)

Процессы ВРМ можно посмотреть по пункту меню «ВРМ/Процессы приложения»:



В области навигации слева открыть пункт «Отчет о количестве заболевших коронавирусом», который содержит главный и вложенный процессы.

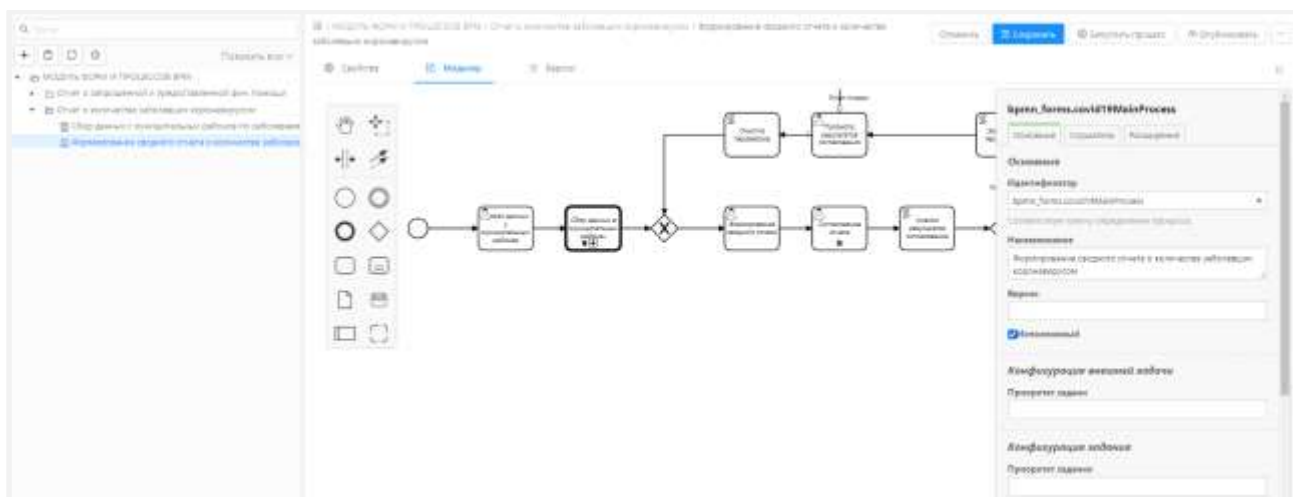


Рисунок 4 – Главный процесс «Формирование сводного отчета о количестве заболевших коронавирусом»

Переход к контексту процесса (настраивается с помощью Конфигуратора) осуществляется на вкладке «Свойства» по ссылке «Настроить контекст процесса»:

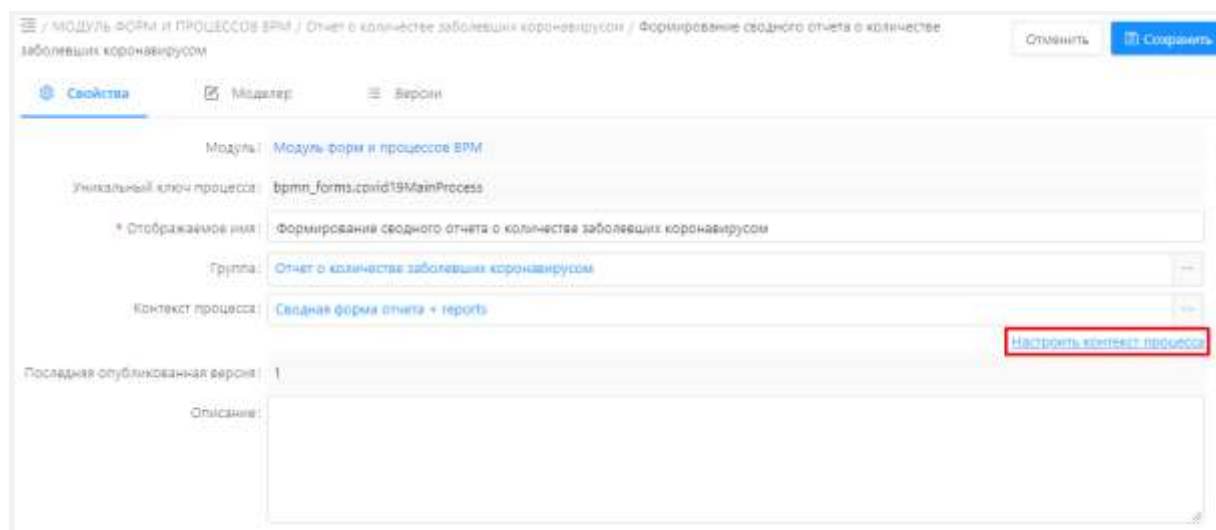


Рисунок 5 – Переход к контексту процесса

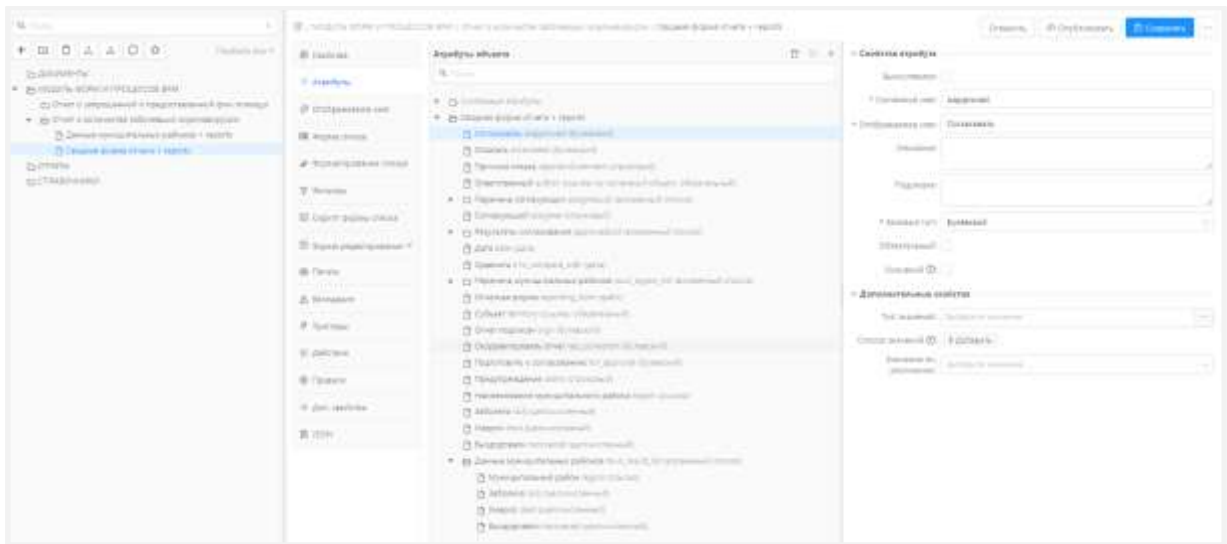


Рисунок 6 – Контекст главного процесса «Формирование сводного отчета о количестве заболевших коронавирусом»

Аналогичным образом можно посмотреть модель и контекст вложенного процесса «Сбор данных с муниципальных районов по заболеваемости коронавирусом».

Отчет по заболеваемости коронавирусом можно посмотреть в меню Справочники.



Рисунок 7 – Отчет по заболеваемости коронавирусом в меню Справочники.

Нажать кнопку «Выполнить». Сформированный отчет можно посмотреть в меню Отчеты/Журнал выполнения отчетов.

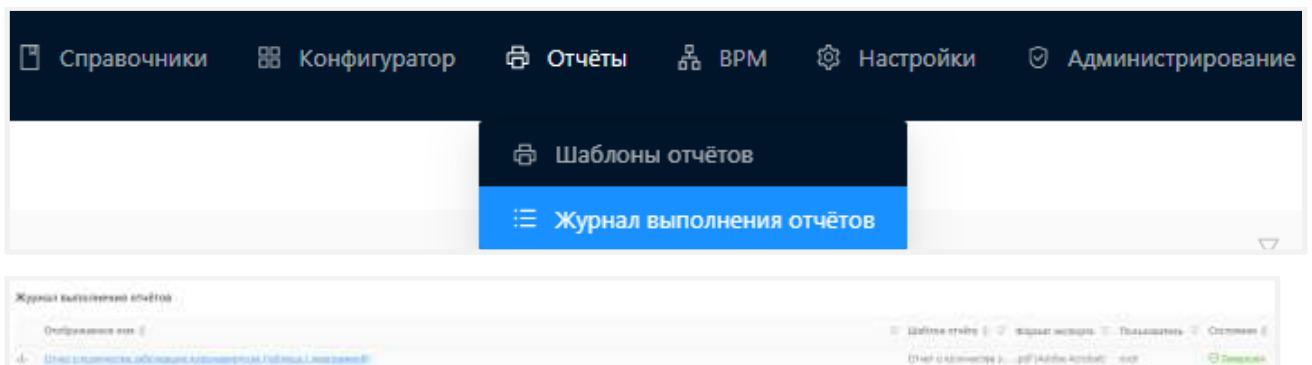
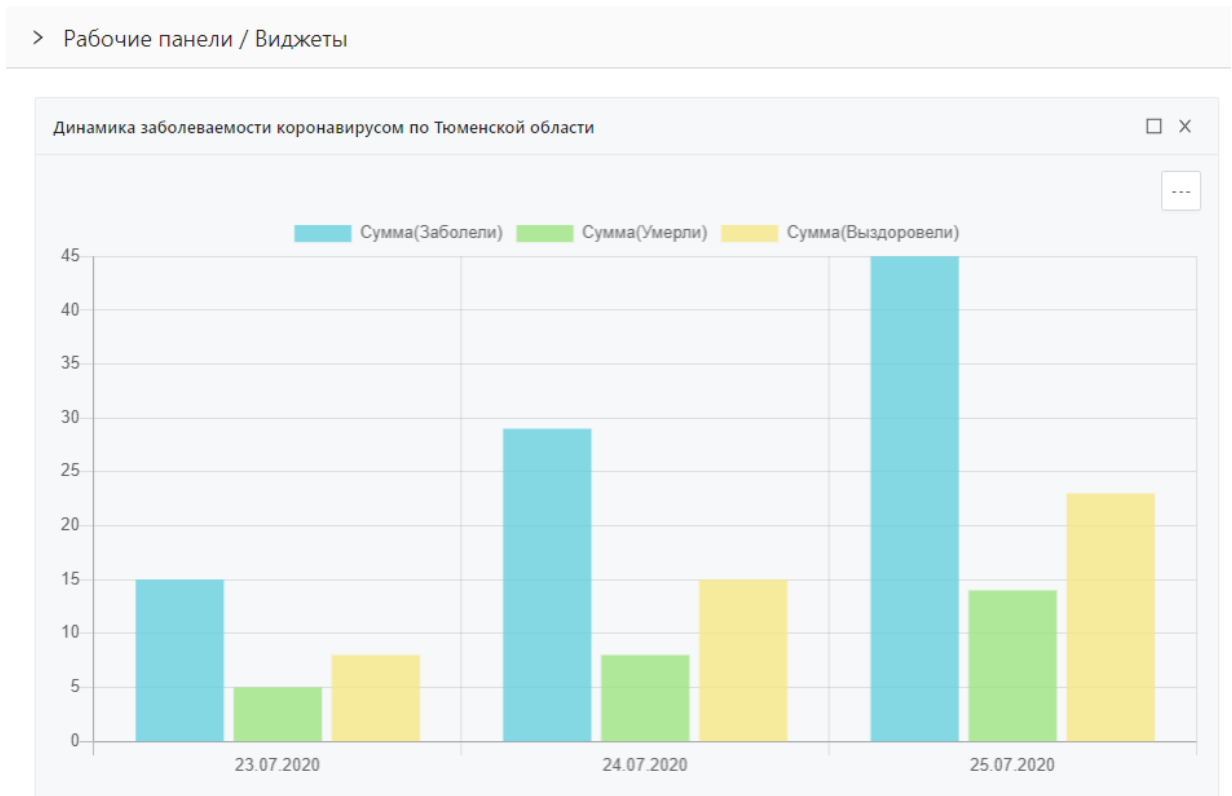




Рисунок 8 – Сформированный отчет

Аналитические представления можно посмотреть в меню «Рабочие панели».



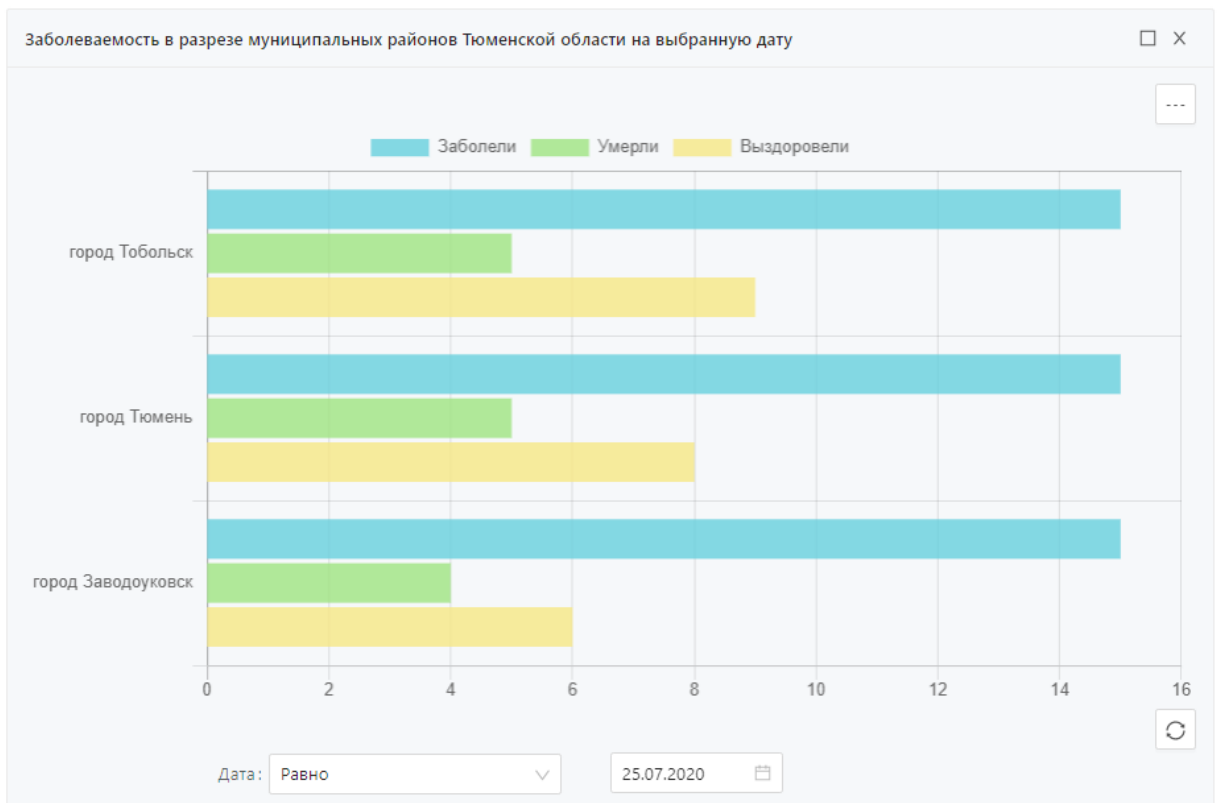


Рисунок 9 – Просмотра аналитического представления в меню «Рабочие панели»

Настройка представлений включает в себя настройку источника данных, настройки виджета и отображения на рабочей панели (меню Администрирование).

Редактирование: Группировка по датам по коронавирусу

Имя: Группировка по датам по коронавирусу

Тип источника: Запрос по объекту приложения

Таблица представления для группировки данных для графика: Результаты по коронавирусу

Сгруппировать по: Дата

Агрегируемые поля

Агрегатная функция: Поле объекта

Сумма: Заболели

Сумма: Умерли

Сумма: Выздоровели

Параметры запроса

Отображаемое имя: Имя параметра

Дата: date

Базовый тип: date

Отменить Сохранить

Рисунок 10 – Источник данных

Редактирование: Динамика заболеваемости коронавирусом по Тюменской области

Окнонные параметры

- Название: Динамика заболеваемости коронавирусом по Тюменской области
- Тип: График или диаграмма
- Доступ для ролей: Супер-пользователь
- Активный:
- Ширина карточек: 3

Дополнительные параметры

- Источник данных: Группировка по дате по коронавирусу
- Тип диаграмм: Столбчатая диаграмма
- Значение по оси X: Дата
- Значения по оси Y: Сумма(Заболевли) * Сумма(Умерли) * Сумма(Выздоровели)
- Показывать всплывающие подсказки при наведении:
- Показывать легенду:
- Расположение легенды: Выброта значения
- Сгруппировать данные в один столбец:
- Палитра: Пастельная
- Поле, значение которого будет передано в качестве глобального параметра:
- Список повторов: Выброта значения

Отменить Сохранить

Рисунок 11 – Настройка виджета

Редактирование: Данные по коронавирусу

Название: Данные по коронавирусу

Отображать в меню "Рабочие панели":

Приказ по учетной записи:

Пользователь: root

Использовать по умолчанию:

Виджеты

Виджет: Динамика заболеваемости коронавирусом по Тюменской области

Заболеваемость в разрезе муниципальных районов Тюменской области на выбранную дату

Дочерние рабочие панели

Название: Нет данных

Отменить Сохранить

Рисунок 12 – Настройка рабочей панели

Приложение 1

Разъяснение к колонкам таблицы:

Обязательно? - приложение не будет работать или будет работать некорректно, если свойство, указанное как обязательное, отсутствует в application.properties.

Значение по умолчанию - значение, которое будет использоваться для свойства, если не задано явно в application.properties.

Свойство	Обязательно?	Значение по умолчанию	Описание	Пример значения
<u>Camunda (движок для BPM)</u>				
camunda.bpm.application.delete-on-undeploy	Да, если подключен BPM-модуль	false	Настройка для BPM модуля. Если установлено в true, то при удалении процесса приложения, будут удалены все активные экземпляры процесса, а также вся история запусков	true
БФТ. Платформа				
dm.fieldSetPackages	Да		Перечень пакетов, в которых проводится поиск датамапов (филдсетов)	com.bftcom.starter.common,com.bftcom.configurator.common,com.bftcom.reports.common,com.bftcom.bpm.common,com.bftcom.ice.common,com.bftcom.mdm.configurator.common,com.bftcom.icewidgets.common
ice.servicesPackages	Да		Перечень пакетов, в которых проводится поиск сервисов, к которым можно будет обратиться с фронта (например, из задания планировщика)	com.bftcom.starter.common.service,com.bftcom.configurator.common.services,com.bftcom.ice.common.service

ice.rls.enabled	Нет	true	Включен ли режим Row Level Security. Если разграничение доступа на уровне строк не требуется, свойство рекомендуется явно выставлять в false	true
ice.script.jvm Packages	Да		Перечисляются пакеты, которые будут доступны в скриптах конфигурации	com.bftcom.ice.common.maps.*, com.bftcom.ice.server.util.*, com.bftcom.ice.common.general.*, com.bftcom.ice.common.service.notification.*, com.bftcom.ice.common.service.notification.NotificationApi.*, com.bftcom.ice.common.utils.*, org.camunda.bpm.engine.impl.persistence.entity.*, com.bftcom.bpm.server.util.*, com.bftcom.bpm.*
ice.dbSync.enabled	Нет	true	Выполнять ли скрипты на БД при запуске приложения	false
ice.consistency.enabled	Нет	true	Проверять ли ссылки при выполнении удаления объекта. Если включено, увеличивается время для выполнения удаления, но обеспечивается целостность данных - нельзя удалить объект, если на него есть ссылки (речь о данных в json-поле)	false
jodconverter . Использует LibreOffice или OpenOffice для преобразования файлов в различные форматы (html, doc, docx, xls, xlsx, pdf и др.). Применяется для просмотра файлов в приложении.				
jodconverter.local.enabled	Нет	true	Разрешено ли преобразование документов с помощью локально установленного на сервере LibreOffice или OpenOffice.org.	true

			Если нет установленного LibreOffice или OpenOffice, свойство нужно установить в false	
Логирование и сжатие HTTP-запросов и ответов				
log.access.enabled			Включено ли логирование HTTP-запросов и ответов	false
log.access.httpRequests.maxPayloadLength			Максимальный размер тела HTTP-запроса, который будет сохранен в лог (в байтах), при включенном логировании	1000
log.access.httpRequests.includePayload			Будет ли логироваться тело HTTP-запроса	true
server.compression.enabled	Нет	false	Включено ли сжатие ответа сервера. Свойство поддерживается серверами приложений Tomcat, Jetty, Undertow. Если включено, будет использоваться сжатие gzip, в заголовке ответа также будет указано accept-encoding = gzip	true
server.compression.min-response-size	Нет	2KB	Минимальный размер ответа, при котором будет использовано сжатие	2048
server.compression.mime-types	Нет	text/html, text/xml, text/plain, text/css, text/javascript, application/javascript, application/json,	Типы содержимого, который нужно сжимать при ответе	application/pdf, application/json, application/xml, text/html, application/javascript, text/css,text/xml,text/plain

		application/xml		
Прочие свойства				
timeMachine.enabled	Нет	false	Вести журнал изменений	true
server.servlet.contextPath	Да		Контекстный путь приложения (путь до приложения в URI)	/app
spring.autoconfigure.exclude	Нет	null	Классы, которые должны быть исключены из автоконфигурации. Свойство можно использовать для отключения отдельных возможностей, например <code>spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.quartz.QuartzAutoConfiguration</code> отключает планировщик заданий в целом	org.springframework.boot.autoconfigure.quartz.QuartzAutoConfiguration
spring.devtools.restart.enabled				false
spring.devtools.livereload.enabled	Нет	true	Разрешено ли автоматическое обновление браузера при изменении ресурса приложения. Для продуктовых сред обновление выключено	false
spring.devtools.restart.trigger-file	Нет			application.properties
Кеширование данных				
spring.cache.jcache.provider	Нет		Используемая библиотека для кеширования данных.	org.ehcache.jsr107.Ehcache

r			<p>Подробнее о провайдере для кеширования <code>org.ehcache.jsr107.EhcacheCachingProvider</code> - см. https://www.ehcache.org/</p> <p>Свойство должно быть определено только если в <code>classpath</code> указано более одной реализации JSR-107, иначе <code>spring</code> сам определит провайдера кеширования</p>	CachingProvider
spring.cache.jcache.config	Нет	null	Путь до файла с настройками кеширования указанного провайдера	classpath:ehcache.xml
Настройки источника данных				
spring.profiles.active	Да		Активный профиль. Определяет поставщика данных в приложении. Возможные значения: <code>postgresql</code> , <code>postgresql95</code> , <code>oracle</code>	postgresql
spring.datasource.url	Да		Путь к базе данных	jdbc:postgresql://localhost:5432/platform
spring.datasource.username	Да		Имя пользователя для соединения с БД	postgres
spring.datasource.password	Да		Пароль для соединения с БД	postgres
Настройки сериализации Java-объектов				
spring.json.default-property-inclusion	Нет	always	Определяет каким образом будут сериализованы свойства Java-объекта. Значение "non_null" говорит о том, что при сериализации будут учтены только свойства с ненулевыми значениями (подробнее о возможных	non_null

			значениях см. https://fasterxml.github.io/jackson-annotations/javadoc/2.7/com/fasterxml/jackson/annotation/JsonInclude.Include.html)	
spring.jackson.serialization.write-dates-as-timestamps	Нет	false	Если true, то значения типа дата-время в Java-объектах будут сериализованы как число; если false - при сериализации даты-времени будет использоваться другой вид (например, текстовое представление)	true
<u>Кодировка HTTP запросов и ответов</u>				
spring.http.encoding.charset	Нет	UTF-8	Кодовая страница HTTP запросов и ответов. Добавляет в заголовок "Content-Type", если он не задан явно	UTF-8
spring.http.encoding.enabled	Нет	true	Включена ли кодировка HTTP запросов и ответов	true
spring.http.encoding.force	Да		Включена ли принудительная кодировка HTTP запросов и ответов в соответствии с указанной кодовой страницей.	true
Допустимые размеры запросов				
spring.servlet.multipart.max-file-size	Нет	1MB	Максимально допустимый размер для каждого передаваемого в запросе файла. Значение -1 означает, что размер файла не ограничен	32MB

spring.servlet.multipart.max-request-size	Нет	10MB	Максимально допустимый размер файловых данных в одном запросе	32MB
Консоль (командная строка приложения, https://projects.spring.io/spring-shell/)				
spring.shell.interactive.enabled	Нет	true	Запрещает или разрешает интерактивную работу консоли приложения	false
<u>Планировщик заданий</u>				
spring.quartz.job-store-type	Нет	memory	Настройка планировщика заданий. Указывает где будет храниться информация (задания, триггеры, календари). jdbc - хранение в реляционной БД	jdbc
spring.quartz.jdbc.initialize-schema	Нет		Нужно ли при старте инициализировать базу данных с помощью стандартных сценариев, предоставляемых библиотекой Quartz. При значении always при каждом запуске будет выполняться скрипт, удаляющий все существующие в БД таблицы планировщика и все триггеры	never
spring.quartz.properties.org.quartz.jobStore.tablePrefix	Нет	QRTZ_	Префикс для таблиц планировщика в БД	quartz.qrtz_
spring.quartz.properties.org.quartz.jobStore.driverDelegateClass	Да		Делегат драйвера для СУБД, которая используется для хранения данных планировщика заданий	org.quartz.impl.jdbcjobstore.PostgreSQLDelegate