ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«БЮДЖЕТНЫЕ И ФИНАНСОВЫЕ ТЕХНОЛОГИИ»

# СИСТЕМА «ЕДИНАЯ СИСТЕМА УПРАВЛЕНИЯ НОРМАТИВНО-СПРАВОЧНОЙ ИНФОРМАЦИЕЙ»
# «БФТ.ЕНСИ»

**Версия 1.11**

# РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

На 76 листах

Москва, 2025

# АННОТАЦИЯ

В данном программном документе приведено руководство системного программиста по установке, настройке и применению Система «Единая система управления нормативно-справочной информацией» «БФТ.ЕНСИ» версия 1.11.

В разделе «Общие сведения о программе» указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы, а также требования к персоналу.

В разделе «Структура программы» приведены сведения о структуре программы, ее составных частях, о связях между составными частями и о связях с другими программами.

В разделе «Настройка программы» приведено описание действий по настройке программы на условия конкретного применения (настройка на состав технических и программных средств, выбор функций и др.).

В разделе «Проверка программы» приведено описание способов проверки, позволяющих дать общее заключение о работоспособности программы (контрольные примеры, методы прогона, результаты).

Документ содержит описание действий системного программиста.

Оформление программного документа «Руководство программиста» выполнено по требованиям ЕСПД (ГОСТ 19.101-77[1], ГОСТ 19.103- 77[2], ГОСТ 19.104-78[3], ГОСТ 19.105-78[4], ГОСТ 19.106-78[5], ГОСТ 19.504-79[6], ГОСТ 19.604-78[7]).

[1] ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов
[2] ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов
[3] ГОСТ 19.104-78* ЕСПД. Основные надписи
[4] ГОСТ 19.105-78* ЕСПД. Общие требования к программным документам
[5] ГОСТ 19.106-78* ЕСПД. Общие требования к программным документам, выполненным печатным способом
[6] ГОСТ 19.504-79* ЕСПД. Руководство программиста. Требования к содержанию и оформлению
[7] ГОСТ 19.604-78* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

**СОДЕРЖАНИЕ**

# 1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Настоящий документ определяет порядок установки, настройки и администрирования Система «Единая система управления нормативно-справочной информацией» «БФТ.ЕНСИ» версия 1.11.

Документ разработан согласно требованиям следующих нормативных документов:

– ГОСТ 19.503-79 Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению;

– ГОСТ Р 59853–2021 «Информационные технологии (ИТ). Комплекс стандартов на автоматизированные системы. Термины и определения».

## 1.1. Назначение программы

Система «Единая система управления нормативно-справочной информацией» «БФТ.ЕНСИ» версия 1.11 (далее – Система) предназначена для централизованного управления нормативно-справочной и реестровой информацией, в том числе с использованием внешних систем, и ее предоставления во внешние системы.

## 1.2. Функции программы

Работа с программой «БФТ.ЕНСИ» осуществляется в двух режимах:

– Настройка объектов приложения, процессов, ролевых моделей или других сущностей, или проектирование системы.

Работу с программой в этом режиме осуществляет оператор с навыками системного аналитика-проектировщика (далее - Администратор). К таким операторам относятся учетные записи с ролями: «Администратор», «Аналитик метаданных».

– Применение настроенной, готовой, системы с целью её использования в том или ином роде деятельности компании.

Работу с программой в данном режиме, кроме администратора, может осуществлять оператор, не владеющий специальными навыками по проектированию системы (далее - Пользователь). К таким операторам относятся учетные записи с ролями: «Пользователь НСИ», «Дата-стюард», «Согласующий/утверждающий».

Система обеспечивает выполнение следующих функций:

- Формирование структуры объектов НСИ;
- Ведение данных объектов НСИ;
- Обеспечение версионности объектов НСИ;
- Ведение Заявок на изменение НСИ;
- Ведение Запросов на изменение, Запросов на добавление записей;
- Обеспечение юридической значимости Заявок на изменение НСИ (опционально);
- Дедупликация данных НСИ;
- Формирование правил валидации;
- Настройка правил проверки качества данных;
- Ведение эталонных записей реестров;
- Распространение НСИ;
- Ведение Паспорта НСИ;
- Формирование отчетов;
- Конфигурирование Системы;
- Администрирование Системы;
- Настройка бизнес-процессов в Системе;
- Настройка виджетов для рабочего стола;
- Аудит действий пользователей;
- Предоставление общероссийских классификаторов;
- Загрузка данных из внешних источников НСИ;
- Поиск данных в объектах НСИ (опционально).

## 1.3. Минимальный состав технических средств

Для обеспечения работоспособности «БФТ.ЕНСИ» требуются технические средства, указанные в таблице ниже.

Таблица 1 – Минимальный состав технических средств «БФТ.ЕНСИ»

| № | Название сервера | CPU (шт) | RAM (Gb) | HDD (Gb) |
|---|---|---|---|---|
| 1 | Сервер приложения | 8 | 16 | 100 |
| 2 | Сервер СУБД | 8 | 16 | 100 |

Для хранения данных на Сервере СУБД должен быть предусмотрен выделенный диск под хранимые данные.

## 1.4. Минимальный состав программных средств

Перечень необходимого системного программного обеспечения, обеспечивающего корректную работу «БФТ.ЕНСИ», представлен в таблице ниже.

Таблица 2 – Перечень необходимого системного программного обеспечения «БФТ.ЕНСИ»

| № | Тип сервера | Примечание | Программное окружение |
|---|---|---|---|
| 1 | Сервер приложения | Операционная система | RedOS 7.3.x / 8.0<br>Astra Linux SE 1.7 / 1.8<br>AltLinux 8 СП / 10 |
| | | JDK (JRE) | OpenJDK 17 / AxiomJDK 17 |
| | | Tomcat | Apache Tomcat 9.0 (не ниже 9.0.86) |
| | | Шрифты | https://srv-nexus-3.bftcom.com/repository/devops/fonts/ms_tt_core.tar.gz |
| 2 | Сервер баз данных приложения | Операционная система в соответствии с требованиями к СУБД | RedOS 7.3.x / 8.0<br>Astra Linux SE 1.7 / 1.8<br>AltLinux 8 СП / 10 |
| | | СУБД PostgreSQL | PostgreSQL 15+, Pangolin 5.1.0+ |
| 3 | Совместимый криптопровайдер | | КриптоПро CSP 4.0 |

## 1.5. Требования к персоналу (системному программисту)

Системный программист должен иметь минимум среднее техническое образование.

В перечень задач, выполняемых системным программистом, должны входить:

– Задача поддержания работоспособности технических средств;

– Задача установки (инсталляции) и поддержания работоспособности «БФТ.ЕНСИ» и необходимых для работы «БФТ.ЕНСИ» системных средств;

– Задача диагностики и определение причин неисправности «БФТ.ЕНСИ» и необходимых для работы «БФТ.ЕНСИ» системных средств.

# 2. СТРУКТУРА ПРОГРАММЫ

## 2.1. Сведения о структуре программы

«БФТ.ЕНСИ» является совокупностью модулей, состоящих из функциональных блоков, которые включают в себя группы функций. Модули подразделяются на основные и дополнительные. Набор модулей в составе программы определяется поставленной задачей.

Модули, входящие в состав «БФТ.ЕНСИ» приведены в таблице ниже.

Таблица 3 – Модули, входящие в состав «БФТ.ЕНСИ»

| №п/п | Системное имя модуля | Название модуля | Тип |
|---|---|---|---|
| 1 | ICE.Core | Ядровой модуль | Основной |
| 2 | ICE.Configurator | Модуль конфигурирования | Основной |
| 3 | ICE.BPM | Модуль бизнес-процессов | Основной |
| 4 | ICE.Reports | Модуль отчетов | Основной |
| 5 | ICE.Reports (Stimulsoft) | Модуль отчетов на основе генератора Stimulsoft | Дополнительный |
| 6 | ICE.Widgets (WIP) | Модуль ядра функциональности виджетов и рабочих панелей | Основной |
| 7 | ICE.Task.Executor | Модуль выполнения фоновых задач | Дополнительный |
| 8 | ICE.FullTextSearch | Модуль полнотекстового поиска | Дополнительный |
| 9 | ICE.Notification | Модуль уведомлений | Дополнительный |
| 10 | СЭП | Сервис электронной подписи | Дополнительный |
| 11 | ICE.Initializr | Инициализатор приложений | Дополнительный продукт |
| 12 | Ag-table | Модуль Ag-table | Дополнительный |
| 13 | AI.Assistant | AI Ассистент | Дополнительный |
| 14 | Attachment Storage | Модуль хранения вложений | Основной |
| 15 | DataFlow | Обеспечение потоков данных | Основной |

Компонентная схема «БФТ.ЕНСИ», с включенными в ее состав модулями, приведена на рисунке ниже.

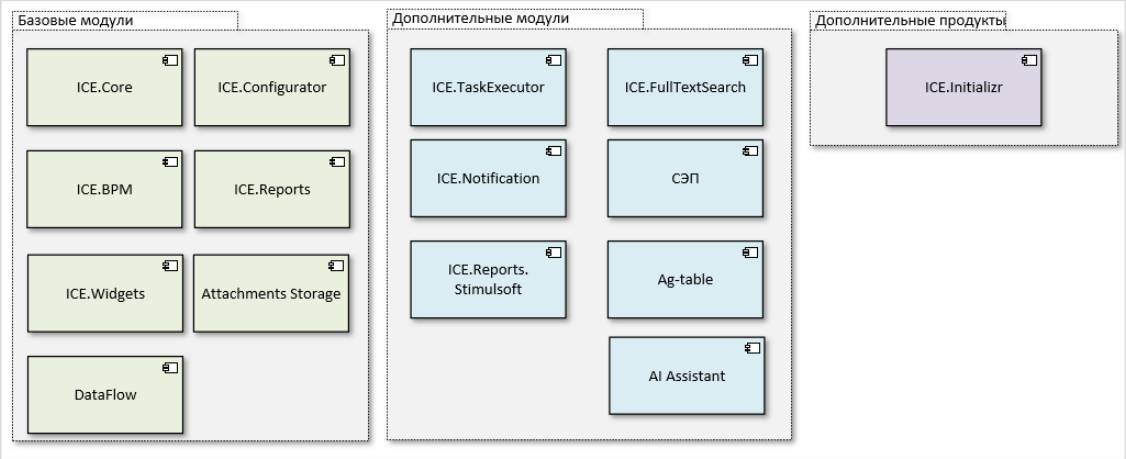


Рисунок 1 – Компонентная схема «БФТ.ЕНСИ»

## 2.2. Сведения о составных частях программы

В состав каждого модуля «БФТ.ЕНСИ» входят библиотеки. Набор библиотек модулей приведен на рисунке ниже.

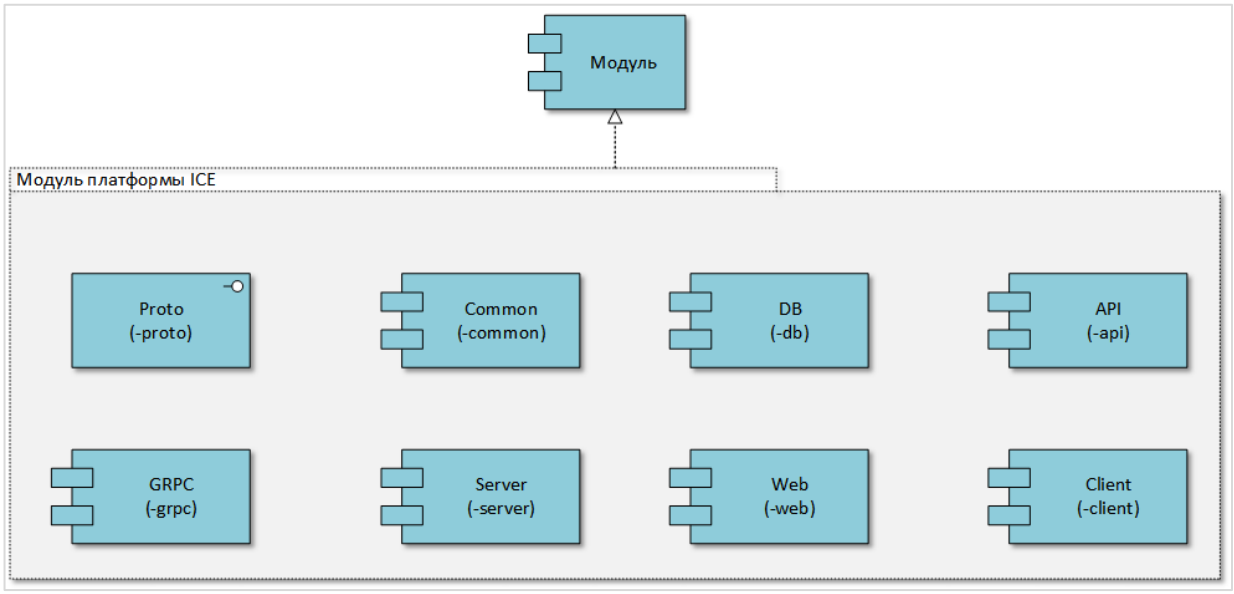


Рисунок 2 – Набор библиотек модуля

В Таблице 4 представлено краткое назначение библиотек, приведенных на Рисунке 2.

Таблица 4 – Назначение библиотек модуля

| № п/п | Библиотека | Назначение | Вид |
|---|---|---|---|
| 1 | Server | Предназначен для реализации серверной бизнес-логики и служебных сервисов | Основной |
| 2 | Web | Предназначен для создания графических интерфейсов | Основной |
| 3 | Common | Предназначен для хранения совместно используемых ресурсов блоков Server и Web | Основной |
| 4 | DB | Предназначен для хранения скриптов для работы с базой данных | Основной |
| 5 | Client | Предназначен для хранения набора программных продуктов, обеспечивающих доступ к серверной части и сервисам | Дополнительный |
| 6 | API | Предназначен для создания программных интерфейсов | Дополнительный |
| 7 | Proto | Предназначен для разделения GRPC спецификаций и сгенерированных на их основе модулей | Дополнительный |
| 8 | GRPC | Предназначен для хранения классов, сгенерированных на основе proto, и утилитных классов для GRPC инфраструктуры | Дополнительный |

## 2.3. Сведения о связях между составными частями программы

Схема, поясняющая связи между составными частями (модулями) «БФТ.ЕНСИ» представлена на Рисунке 3.

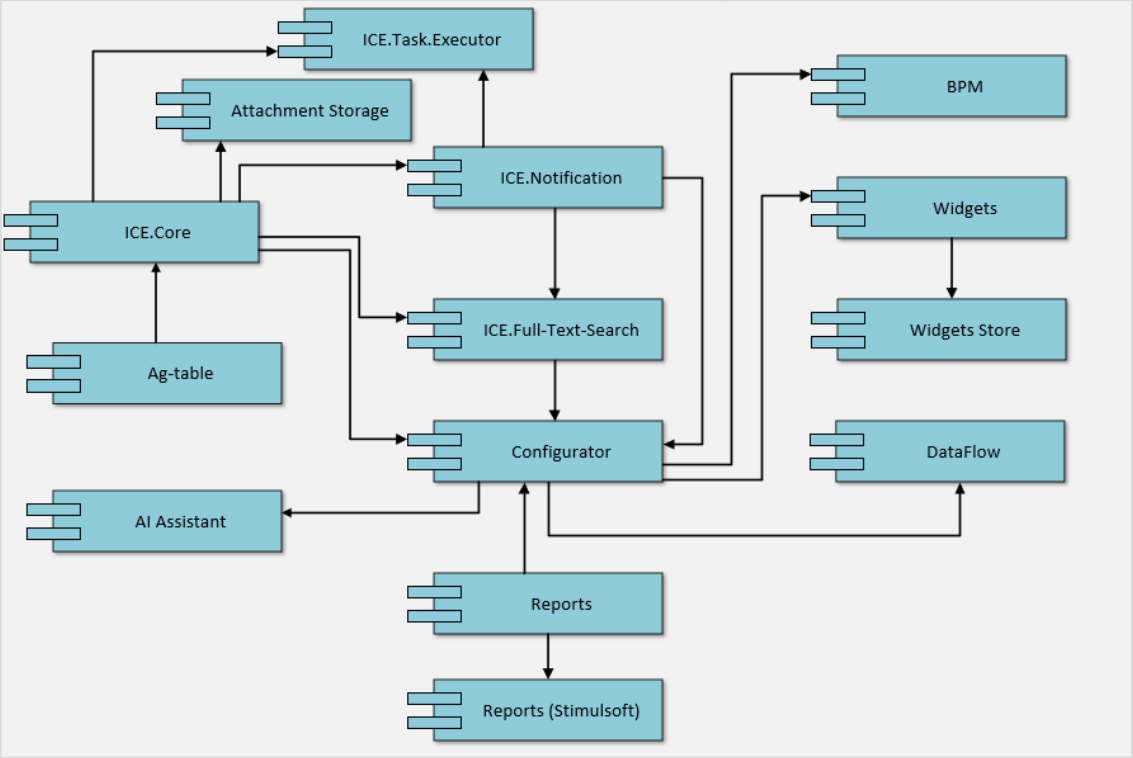


Рисунок 3 – Связи между составными частями «БФТ.ЕНСИ»

## 2.4. Сведения о связях с другими программами

Перечень программ и сервисов, взаимодействующих с «БФТ.ЕНСИ», приведен в Таблице 5.

Взаимодействие с другими программами и сервисами осуществляется синхронным и асинхронным методом через API. При передаче данных могут использоваться брокеры сообщений (kafka, rabbit). Выгружаемые данные передаются в формате json или виде структурированных файлов расширения csv, xml, xsd.

Таблица 5 – Характеристики взаимосвязей «БФТ.ЕНСИ» с другими программами и сервисами

| **Сервис/другая программа** | **Характер взаимодействия** |
|---|---|
| ГИС ЕСИА | Взаимодействие осуществляется для получения возможности аутентификации пользователя через систему ЕСИА. |
| МДМ | Взаимодействие осуществляется посредством штатного интеграционного механизма в программе через REST API и брокер сообщений ActiveMQ, использующий JMS протокол. |

| Сервис/другая программа | Характер взаимодействия |
|---|---|
| | REST API используется для запроса доступных для интеграции справочников (реестров) и для передачи из МДМ в программу схем справочников (реестров).<br>Первоначальная загрузка данных справочника осуществляется через очередь сообщений посредством выполнения серверного задания, прикрепленного к переходу «Загрузить начальное решение» статусной модели процесса интеграции сущности. После загрузки начального решения справочник будет наполнен актуальными данными из МДМ на дату сервера, на которую осуществляется загрузка. Последующая загрузка данных из МДМ возможна 2 способами: синхронизация и перезагрузка данных.<br>При синхронизации набор загружаемых данных справочника из МДМ определяется от даты последней синхронизации до даты сервера, в которую осуществляется загрузка. Таким образом, в программу поступают только актуальные данные справочника, которые были добавлены или изменены с помощью заявок в МДМ с даты последнего выполнения загрузки начального решения или синхронизации.<br>При перезагрузке данных осуществляется полное обновление пакета данных не зависимо от даты сервера и даты выполнения последней синхронизации. Поступающие в программу данные справочника из МДМ сравниваются с текущими и при нахождении несоответствий происходит обновление данных справочника в программе. |
| Аванпост | Взаимодействие осуществляется через REST API.<br>Аванпост FAM (Federated Access Manager) система, позволяющая проводить прозрачную и многофакторную аутентификацию в мобильных и веб-приложениях, приложениях с толстым клиентом, SaaS-сервисах и терминальных решениях, как на базе стандартных протоколов (Kerberos, SAML, OAuth), так и с помощью метода перехвата окон аутентификации информационных систем.<br>В программе реализована возможность аутентификации через Аванпост FAM с использованием протокола OAuth 2.0.<br>Аванпост IDM система, позволяющая централизованно управлять учетными записями и правами доступа пользователей в различных информационных системах компании. |

# 3. НАСТРОЙКА ПРОГРАММЫ

## 3.1. Настройка и состав технических средств

Состав технических средств описан в пункте 1.3 настоящего документа.

Дополнительных требований к настройке технических средств, кроме требований, предъявляемых применяемым системным программным обеспечением, не предъявляется.

## 3.2. Настройка и состав программных средств

Инструкция по скачиванию, установке экземпляра программного обеспечения и запуску программы «БФТ.ЕНСИ» на RedOS размещена в Приложении 1.

### 3.2.1. Запуск «БФТ.ЕНСИ»

Работа в «БФТ.ЕНСИ» доступна только для зарегистрированных пользователей.

Для перехода к окну авторизации в строке адреса браузера вводится адрес сервера «БФТ.ЕНСИ», при этом открывается форма для авторизации пользователя (Рисунок 4). В данной форме необходимо ввести учетные данные (логин и пароль) и нажать кнопку «Войти».

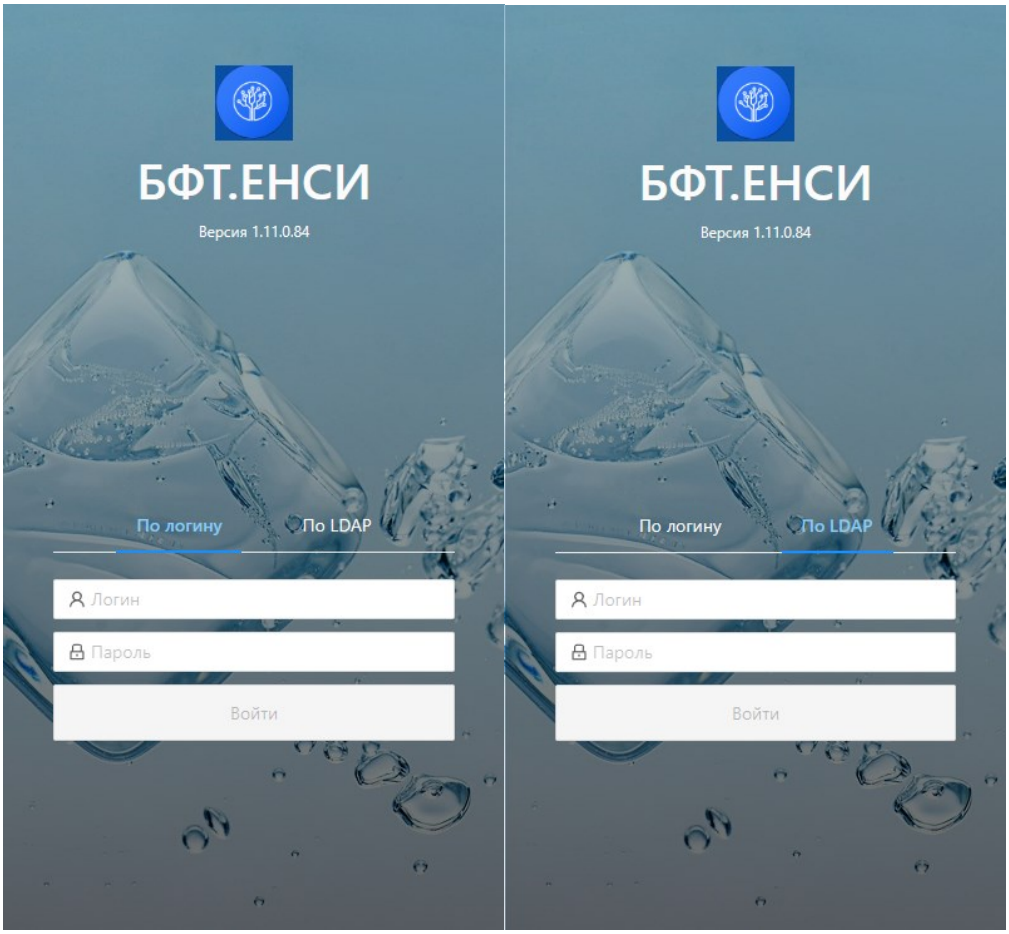


Рисунок 4 – Форма для авторизации пользователя в «БФТ.ЕНСИ»

Если у пользователя установлено средство криптографической защиты информации (например, «КриптоПро ЭЦП Browser Plug-in») и к учетной записи привязан сертификат, то возможна авторизация по данному сертификату. Если учетная запись пользователя связана с учетной записью LDAP, возможна авторизация по LDAP.

После успешной авторизации откроется страница с доступными для учетной записи пользователя разделами в Системе (Рисунок 5).

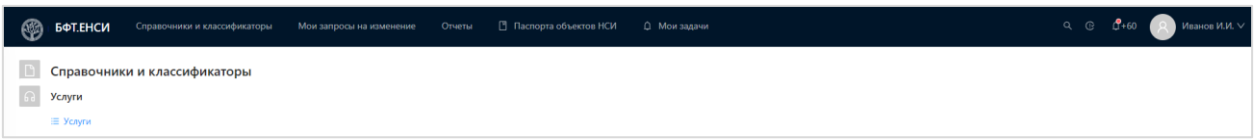


Рисунок 5 – Главное окно БФТ.ЕНСИ

При вводе неверных данных на экране появится информационное сообщение: «Неправильное имя пользователя или пароль».

# 4. ПРОВЕРКА ПРОГРАММЫ

## 4.1. Описание способов проверки

После установки, обновления, перезапуска, восстановления данных рекомендуется выполнить контрольный пример, описание которого представлено в Таблице 6.

Таблица 6 – Контрольный пример

| № п/п | Операции и действия | Ожидаемый результат |
|---|---|---|
| 1 | В адресной строке интернет веб-браузера ввести адрес сервера системы.<br>Указать следующую информацию:<br>– Логин;<br>– Пароль.<br>Нажать кнопку «Войти». | В результате откроется окно авторизации пользователя. |
| 2 | В случае ввода верной информации (указаны зарегистрированный в системе пользователь и правильный пароль) осуществится вход в главное окно Системы. | В результате откроется главное окно Системы. |
| 3 | Перейти в раздел «Конфигуратор-> Объекты приложения», выбрать объект приложения. | В результате откроется списочная форма объекта приложения. |
| 4 | В списочной форме открыть запись объекта приложения. | В результате откроется форма просмотра записи объекта приложения. |

# 5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Дополнительные возможности отсутствуют.

# 6. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

В ходе настройки и функционирования «БФТ.ЕНСИ» могут выводиться сообщения, требующие от системного программиста выполнения определенных действий. Описание этих действий приводится в сообщении от «БФТ.ЕНСИ».

Также, сообщения могут выводиться в журнале работы сервера Apache Tomcat.

ПРИЛОЖЕНИЕ 1

# ИНСТРУКЦИЯ ПО СКАЧИВАНИЮ, УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ЗАПУСКУ ПРОГРАММЫ «БФТ.ЕНСИ» НА REDOS

## 1.1. Общая информация

Инструкция описывает установку «БФТ.ЕНСИ» на операционную систему версии RedOS 7.3.x.

Параметры конфигурации памяти -Xms4G -Xmx6G (минимальные) для сервиса tomcat-starter-8080.service указаны для примера. Их необходимо изменять в зависимости от ресурсов, выделенных для данного приложения.

## 1.2. Скачивание дистрибутива

Дистрибутив программы «БФТ.ЕНСИ» может быть предоставлен по запросу. Контактная информация размещена на сайте https://bftcom.com/.

Apache Tomcat можно скачать по ссылке https://archive.apache.org/dist/tomcat/tomcat-9/ , взяв версию Tomcat 9.0.100 и выше:
https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.100/bin/apache-tomcat-9.0.100.tar.gz

## 1.3. Установка программы «БФТ.ЕНСИ» на РЕД ОС

Порядок установки:

- Установка шрифтов;
- Установка OpenJDK;
- Установка PostgreSQL;
- Создание базы приложения;
- Установка и настройка Tomcat;
- Настройка портов брандмауэра;
- Установка JodConverter;

– Установка и настройка СЭП.

### 1.3.1. Установка OpenJDK

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

***dnf install java-17-openjdk***

### 1.3.2. Установка PostgreSQL

#### 1.3.2.1. Установка PostgreSQL 15 версии и выше.

Для установки *postgresql15* выполните команду:

***sudo dnf install postgresql15-server***

#### 1.3.2.2. Инициализация базы

Произвести инициализацию базы данных postgresql:

***sudo postgresql-15-setup initdb***

#### 1.3.2.3. Разрешение автозапуска службы postgres

***sudo systemctl enable postgresql-15.service***

#### 1.3.2.4. Внесение в конфигурацию доступа для будущей базы

##### 1.3.2.4.1. Сохранение копии оригинального файла настроек

***sudo mv /var/lib/pgsql/15/data/pg_hba.conf /var/lib/pgsql/15/data/pg_hba.conf_original***

**1.3.2.4.2. Внесение изменений**

***echo "# TYPE  DATABASE        USER      ADDRESS   METHOD" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "# local is for Unix domain socket connections only" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "  local   all             all                                     peer" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "# IPv4 local connections:" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "#  host    all             all             127.0.0.1/32            ident" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "# IPv6 local connections:" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "#  host    all             all             ::1/128                 ident" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "# Allow replication connections from localhost, by a user with the" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "# replication privilege." | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "local   replication     all                                     peer" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "host    replication     all             127.0.0.1/32            ident" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "host    replication     all             ::1/128                 ident" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "host    db_starter  u_starter   127.0.0.1/32   scram-sha-256" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "host    db_starter  u_starter   ::1/128     scram-sha-256" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

***echo "host    all             all         0.0.0.0/0               scram-sha-256" | sudo tee -a /var/lib/pgsql/15/data/pg_hba.conf***

Или, используя любой удобный редактор, приводим файл к виду:

***nano /var/lib/pgsql/15/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_starter u_starter 127.0.0.1/32 scram-sha-256***

***host db_starter u_starter ::1/128 scram-sha-256***

***host all all 0.0.0.0/0 scram-sha-256***

Изменим владельца файла:

***sudo chown postgres:postgres /var/lib/pgsql/15/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.

#### 1.3.2.4.3. Запуск сервиса

***sudo systemctl start postgresql-15.service***

проверить, что служба запущена (В статусе должно отображаться active (running)):

***sudo systemctl status postgresql-15.service***

### 1.3.3. Создание пользователя и базы данных

#### 1.3.3.1. Создание пользователя базы данных

***sudo -u postgres psql -c "CREATE USER u_mdm25 WITH LOGIN BYPASSRLS NOSUPERUSER CREATEROLE***

***\password u_mdm25 #(ввести пароль например - PASSWORD 'P@s$w0rd'")***

#### 1.3.3.2. Создание базы данных

***sudo -u postgres psql -c "CREATE DATABASE db_mdm25 OWNER u_mdm25; "***

***sudo -u postgres psql -c "COMMENT ON DATABASE db_mdm25 IS 'MDM25 / ENSI';"***

***\c db_mdm25***

#### 1.3.3.3. Создание схемы и выдача на неё права пользователю

***sudo -u postgres psql -d db_mdm25 -c "CREATE SCHEMA bft;"***

***sudo -u postgres psql -d db_mdm25 -c "ALTER SCHEMA bft OWNER TO u_mdm25;"***

***sudo -u postgres psql -d db_mdm25 -c "GRANT ALL PRIVILEGES ON SCHEMA bft TO u_mdm25;"***

***sudo -u postgres psql -d db_mdm25 -c "GRANT ALL PRIVILEGES ON DATABASE db_mdm25 TO u_mdm25;"***

Для развертывания нового экземпляра данных действий достаточно.

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

***pg_restore -h localhost -U u_mdm25 -F c -d db_mdm25 ./BaseDump/db_mdm25.tar.gz***

Примечание: Необходимо указать полный путь, где находится копия данных.

### 1.3.4. Установка и настройка Tomcat

#### 1.3.4.1. Создание папки для работы приложения

***sudo mkdir -p /opt/_Tomcat/ENSI-8080***

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

Примечание: Необходимо указать полный путь, где находится распакованный дистрибутив из раздела 1.2 настоящего документа.

#### 1.3.4.2. Создание пользователя и группы для Tomcat

***sudo groupadd tomcat***
***sudo useradd -M -d /opt/_Tomcat -g tomcat --system tomcat***

#### 1.3.4.3. Установка прав 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 {} \;***

#### 1.3.4.4. Создание каталогов для журналов и технологического каталога

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

Создаём технологические каталоги:
***sudo mkdir -p <TMC>/.ice***
***sudo chown tomcat:tomcat <TMC>/.ice***
***sudo chmod 750 <TMC>/.ice***
***sudo mkdir /opt/_Tomcat/ENSI-8080/temp***
***sudo chown tomcat:tomcat /opt/_Tomcat/ENSI-8080/temp***
***sudo chmod 750 /opt/_Tomcat/ENSI-8080/temp***

Создаём файл /spring-shell.log (его необходимость зависит от параметров, указанных в <TMC>/conf/catalina.properties)
***sudo touch /spring-shell.log***

***sudo chown tomcat:tomcat /spring-shell.log***

***sudo chmod 750 /spring-shell.log***

Создаём скрипт для архивации журналов (архивирует все журналы за определённую дату и сохраняет архивы в каталоге <TMC>/logs/archiv в течении недели)

***sudo mkdir -p /root/_Scripts/***

***sudo touch /root/_Scripts/arch_log_date-tomcat.sh***

приводим его к виду:

```
#!/bin/bash
mkdir -p $1/archiv
find $1/*$(date --date '-1 day' +%Y-%m-%d)*.* -exec tar -r -f $1/archiv/$(date --date '-1 day' +%Y%m%d).tar --remove-files {} \; && find $1/archiv/*.tar -exec gzip {} \;
&& find $1/archiv/*.tar.gz -mtime +5 -delete

chown $2:$3 $1/archiv/*.tar.gz
chmod 640 $1/archiv/*.tar.gz
```

Для logrotate создаём файл-конфигурацию ротации основного файла журнала Tomcat-a (при активной работе с приложениями в Tomcat он увеличивается очень быстро) /etc/logrotate.d/tomcat следующего содержания:

```
/var/log/tomcat/ENSI-8080/catalina.out
{
        rotate 10
        size 200M
        compress
        notifempty
        missingok
        copytruncate
        su tomcat tomcat
}
```

В файл конфигурации "планировщика" **/etc/crontab** добавляем две строки:

```
0  */12 *  *  * root    logrotate --force /etc/logrotate.d/tomcat > /dev/null 2>&1
# для ENSI-8080
```

*0 1 * * * root /root/_Scripts/arch_log_date-tomcat.sh /var/log/tomcat/ENSI-8080 tomcat tomcat > /dev/null 2>&1*

### 1.3.4.5. Добавление необходимых параметров в конфигурационные файлы Tomcat.

Для этого в файле конфигурации сервера Tomcat/opt/_Tomcat/ENSI-8080/conf/server.xml добавляем дополнительные необходимые параметры **maxThreads, compresssion, compressibleMimeType**:

```
<Connector port="8080" protocol="HTTP/1.1"
   connectionTimeout="20000"
   compressibleMimeType="text/html,text/xml,text/plain,text/css,text/javascript,application/javascript,application/json,application/xml"
   compression="on" compressionMinSize="8192" useSendfile="false"
   maxThreads="3800"
   maxSwallowSize="105906176"
   maxParameterCount="1000"
   redirectPort="8443" />
```

При необходимости настройки работы по **HTTPS** необходимо добавить дополнительные блок параметров:

```
### после
<Connector port="8080"....
### добавляем
### для ENSI-8080
<Connector port="8443" protocol="HTTP/1.1" scheme="https" secure="true"
   SSLEnabled="true" sslProtocol="TLSv1.2" clientAuth="false"
   SSLVerifyClient="optional"
   connectionTimeout="20000"
   compressibleMimeType="text/html,text/xml,text/plain,text/css,text/javascript,application/javascript,application/json,application/xml"
   compression="on" compressionMinSize="8192" useSendfile="false"
   maxThreads="3800"
   maxParameterCount="1000"
   maxSwallowSize="105906176"
# при использовании отдельных файлов сертификатов
   SSLCertificateFile="/opt/_Tomcat/ENSI-8080/conf/ssl/tls.crt"
   SSLCertificateKeyFile="/opt/_Tomcat/ENSI-8080/conf/ssl/tls.key"
# при использовании криптоконтейнеров
```

*keystoreFile="/opt/_Tomcat/ENSI-8080/conf/ssl/keystore.p12"*
*keyAlias="application1" keystorePass="**********" keystoreType="PKCS12"*
*/*

Если необходимо, создадим самоподписанные ключи:

***### для ENSI-8080***

***# при использовании криптоконтейнеров***

***sudo mkdir -p /opt/_Tomcat/ ENSI -8080/conf/ssl***

***sudo keytool -genkeypair -keyalg RSA -keysize 2048 -storetype PKCS12 -keystore keystore.p12 -validity 3650 -alias application1 -file /opt/_Tomcat/ ENSI -8080/conf/ssl/keystore.p12***

***# при использовании отдельных файлов сертификатов***

***openssl req -new -x509 -keyout /opt/_Tomcat/ ENSI -8080/conf/ssl/tls.key -out /opt/_Tomcat/ ENSI -8080/conf/ssl/tls.crt -days 7300 -nodes***

В файл **<TMC>/conf/context.xml** перед закрывающим тегом **</Context>** добавим строку с параметрами кешироания:

...

***<Resources***
***cachingAllowed="true" cacheMaxSize="100000" cacheTtl="2000" />***
***</Context>***

#### 1.3.4.6. Создание файла systemd-юнита для запуска Tomcat в качестве сервиса

***sudo vi /etc/systemd/system/tomcat-ensi-8080.service***

***----------Начало скрипта ----------***

***# Systemd unit file for Tomcat - ENSI***

***[Unit]***

***Description=Apache Tomcat Web Application Container***

***After=syslog.target network.target***

***[Service]***

***Type=forking***

***User=tomcat***

***Group=tomcat***

*UMask=0007*

*Environment='JAVA_HOME=/usr/lib/jvm/jdk-17'*

*Environment='CATALINA_PID=/opt/_Tomcat/ENSI-8080/temp/tomcat.pid'*

*Environment='CATALINA_HOME=/opt/_Tomcat/ENSI-8080'*

*Environment='CATALINA_BASE=/opt/_Tomcat/ENSI-8080'*

*Environment='JAVA_OPTS=-server -XX:+UseParallelGC -XX:MaxGCPauseMillis=1000 -Dfile.encoding=UTF-8 -Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -Djava.io.tmpdir=/opt/_Tomcat/ENSI-8080/temp'*

*# для продуктивных стендов*

*Environment='CATALINA_OPTS=-Xms2G -Xmx4G -XX:+DisableAttachMechanism'*

*# для тестовых стендов (включён JMX для удалённой отладки и возможность снятия дампа памяти)*

*# Environment='CATALINA_OPTS=-Xms2G -Xmx4G -Djava.rmi.server.hostname=<IP_или_имя_сервера> -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=0.0.0.0:8000 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9000 -Dcom.sun.management.jmxremote.rmi.port=9000 -Dcom.sun.management.jmxremote.local.only=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false'*

*WorkingDirectory=/opt/_Tomcat/ENSI-8080*

*ExecStart=/opt/_Tomcat/ENSI-8080/bin/startup.sh*

*ExecStop=/bin/kill -15 $MAINPID*

*StandardOutput=null*

*# возможный вариант*

*# StandardOutput=append:/dev/null*

*StandardError=journal*

*# возможный вариант*

*# StandardError=syslog*

*RestartSec=10*

*Restart=always*

*[Install]*

*WantedBy=multi-user.target*

***----------Конец скрипта----------***

#### 1.3.4.7. Перечитывание сервисов

***sudo systemctl daemon-reload***

#### 1.3.4.8. Разрешение автозапуска Tomcat

***sudo systemctl enable tomcat-ENSI-8080.service***

#### 1.3.4.9. Параметры для компонентов

Указанный файл <TMC>/conf/catalina.properties

***### Домашний каталог конкретного экземпляра установленной копии Tomcat (на примере Application1)***
***ice.projectRoot = \opt_Tomcat\ENSI-8080***

***### Профили (уточняются в зависимости от проекта)***
***spring.profiles.active = postgresql,combinator***

***### Подключение к СУБД (с версии 1.8 указывается имя схемы)***
***spring.datasource.url = jdbc:postgresql://<имя_сервера_СУБД>:<порт_сервера_СУБД>/<имя_базы>?currentSchema=<имя_схемы>***
***spring.datasource.username = *******
***spring.datasource.password = **********
## с версии 1.8, указывается та же роль, что и в spring.datasource.username***
***ice.db.roleName = *******
## для всех установок***
***dm.defaultScheme = <имя_схемы>***
***## для установок, использующих BPM модуль с Camunda (точка в параметре префикса - ОБЯЗАТЕЛЬНА)***
***camunda.bpm.database.schema-name = <имя_схемы>***
***camunda.bpm.database.table-prefix = <имя_схемы>.***
***camunda.bpm.metrics.db-reporter-activate = false***
***camunda.bpm.metrics.enabled = true***

***### Разное***
***## отключение JMX***
***spring.jmx.enabled = false***

*### Параметры HikariCP для работы с СУБД*
*### https://www.codetd.com/en/article/15245350*
*### указанно для тестовых стендов, для продуктивных оценивается отдельно*
*# spring.datasource.hikari.connection-timeout=60000*
*# spring.datasource.hikari.idle-timeout=300000*
*spring.datasource.hikari.maximum-pool-size = 5*
*# spring.datasource.hikari.minimum-idle = 5*
*# Применять, если есть проблемы в работе с базой, однозначно не связанные с запросами (сложностью, оптимизацией)*
*spring.datasource.hikari.leak-detection-threshold = 2000*

*### Детализация информации об ошибках*
*## Для продуктовых площадок (при возникновении ошибки в сообщении пользователю будут отсутствовать детали, в том числе SQL-запрос)*
*ice.exception.detailed-info = false*
*## Для стендов разработки и тестирования (расширенная информация об ошибках в журнале работы)*
*ice.exception.detailed-info = true*

*### Уровни журналирования работы приложений (TRACE,DEBUG,INFO,WARN,ERROR,FATAL,ALL,OFF)*
*logging.level.root = ERROR*

*### Встроенные проверки сервисов (применять по необходимости)*
*management.health.jms.enabled = false*
*management.health.elasticsearch.enabled = false*
*management.health.ldap.enabled = false*
*management.health.cassandra.enabled = false*

*### Actuator*
*management.endpoint.health.show-details = always*
*## Публикация все методов*
*management.endpoints.web.exposure.include = **
*## ... только health*
*management.endpoints.web.exposure.include = health*
*## Запрет публикации методов*
*management.endpoints.web.exposure.exclude=actuator,documentation*

### 1.3.5. Настройка интерфейса управления

Для предоставления доступа к консоли управления приложениями "Manager App" необходимо описать доступ в файлах контекста обоих приложений <TMC>/webapps/manager/META-INF/context.xml таким образом:

*...*
*# "по умолчанию"*

```
   allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
# вариант "разрешить всем"
   allow="^.*$" />
# вариант "разрешить только из сетей БФТ" (самый правильный), исключая один
   allow="127\.0\.0\.\d+|10\.255\.253\.\d+|10\.255\.254\.\d+|10\.255\.255\.\d+|172\.\d+\.\d+\.\d
+"
   deny="172\.21\.3\.5" />
</Context>
```

Аналогичным образом настраивается доступ к "Host Manager" и другим (если они нужны). После изменения - перезапускаем Tomcat.

```
# для ENSI-8080
sudo systemctl restart tomcat-ENSI-8080.service
```

Если же требуется глобально ограничить доступ к **Tomcat** (сразу по всем приложениям), то в файле **<TMC>/conf/server.xml** добавим строки (в примере - ограничения для доступа только из внутренних сетей **БФТ**):

```
...
# в конце файла в блоке
   <Host name="localhost"  appBase="webapps"
#    добавим следующие
   <Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="127\.0\.0\.\d+|10\.255\.253\.\d+|10\.255\.254\.\d+|10\.255\.255\.\d+|172\.\d+\.\d+\
.\d+"
      deny="172\.21\.3\.5" />
</Context>
```

Открытие доступа к порту:

а) на межсетевом экране **firewalld**:

```
# открываем доступ к контейнерам и интерфейсу управления для указанной
сети
# добавляем требуемые порты 8080
sudo firewall-cmd --permanent --new-service=tomcat-http
sudo firewall-cmd --permanent --service=tomcat-http --add-port=8080/tcp
sudo firewall-cmd --permanent --add-service=tomcat-http
sudo firewall-cmd --reload
### Далее - примеры (!) для настройки более "тонкой" настройки доступа:
sudo firewall-cmd --permanent --add-rich-rule 'rule family="ipv4" source
address="192.168.1.0/24" service name="http" accept'
```

б) в **SELinux**:

Проверить, что **SELinux** активен:

***sudo sestatus***

Если в ответе на эту команду в строке **"SELinux status:"** будет **enabled** - проверяем целевые порты и добавляем по необходимости:

***# Посмотреть, открыты ли порты наших серверов Tomcat***

***sudo semanage port -l | grep -E "8080"***

***# если какого либо из требуемых портов нет - добавить в этот список (на примере 8080)***

***sudo semanage port -a -t http_port_t -p tcp 8080***

Так же проверяем, если ли блокировки со стороны **SELinux**:

***sudo cat /var/log/audit/audit.log | grep denied | grep catalina.sh***

Если в выводе будут строки, то создаём и применяем на основе их разрешающее правило:

***sudo cat /var/log/audit/audit.log | grep denied | grep catalina.sh | audit2allow -M my_tomcat***

***sudo semodule -i my_tomcat.pp***

При случае, когда утилита **semanage** не установлена на сервере - установим её (и повторим затем предыдущие команды):

***sudo yum install policycoreutils-python***

Для завершения разворачивания в конфигурационный файл **<TMC>/conf/catalina.properties** внести параметры разворачиваемого приложения (для каждого - свой список) и скопировать файл приложения (обычно **app.war**) в каталог **<TMC>/webapps** (при необходимости установив ему права и разрешения (660 и tomcat:tomcat)). Если Tomcat был запущен, он сам развернёт это приложение, если запущен не был - надо его запустить.

### 1.3.6. Установка шрифтов

Для поддержки правильного отображения документов добавляются шрифты:

***# пример для RedOS 7.3+***

***dnf install -y fontconfig***

***wget https://srv-nexus-3.bftcom.com/repository/devops/fonts/ms_tt_core.tar.gz***

***mkdir -p /usr/share/fonts***

***tar xvf ./fonts/ms_tt_core.tar.gz -C /usr/share/fonts***

***/usr/bin/fc-cache -f /usr/share/fonts/msttcore***

### 1.3.7. Установка JodConverter

Данный сервис запускается в виде Docker-контейнера, поэтому на сервер, где он будет запускаться, должны быть установлены docker и docker-compose (в форме отдельного приложения или в виде расширения docker-a):

***sudo dnf install docker-ce docker-compose***

Предварительно необходимо загрузить файл: jodconverter.save.gz.

Загруженный вместе с дистрибутивом «БФТ.ЕНСИ» файл докер-образа jodconverter.save.gz загружаем на сервере в локальное хранилище:

***zcat jodconverter.save.gz | docker load***

Проверяем, что всё нормально:

***docke images***

В списке образов в локальном хранилище будет образ:

***localhost:5000/jodconverter:v1***

Создаём каталог для размещения файлов приложения:

***mkdir –p /opt/_Docker/jodconverter***

Создаём файл **/opt/_Docker/jodconverter/docker-compose.yaml** :

sudo vi /opt/_Docker/jodconverter/docker-compose.yaml

и приводим его к виду:

```
---
version: '3.7'
services:
  jodconverter:
#---------
# в сети БФТ
    image: harbor.bftcom.com/base_int/common/jodconverter/rest:4.4.8.9
# на площадке заказчика
    image: localhost:5000/jodconverter/rest:4.4.8.9
#---------
    restart: always
    labels:
      - name_service=jodconverter
```

```
    networks:
      - jodconverter_net
    ports:
      - 8081:8080
    volumes:
      - ./application.properties:/etc/app/application.properties
networks:
  jodconverter_net:
    driver: bridge
```

Создаём файл /opt/_Docker/jodconverter/.env с содержимым:

***PORT_EXT=8881***

Создаём файл /opt/_Docker/jodconverter/application.properties с содержимым:

***spring.servlet.multipart.max-file-size: 50MB***

***spring.servlet.multipart.max-request-size: 50MB***

Запуск JodConverter:

***cd /opt/_Docker/jodconverter***

***docker-compose up -d***

### 1.3.8. Установка и настройка СЭП

Инструкция по установке и настройке СЭП представлена в Приложении «Установка и настройка СЭП» (Приложение 2).

### 1.3.9. Установка и настройка sDWL

#### 1.3.9.1. Создание папки для работы приложения

***mkdir -p /opt/ensi_sdwl***

#### 1.3.9.2. Создание каталога для журналов и технологического каталога

***mkdir /opt/ensi_sdwl/{config,logs,share,temp}***

#### 1.3.9.3. Создание пользователя

***useradd -r -s /usr/sbin/nologin bft-sdwl***

#### 1.3.9.4. Размещение файлов sDWL в корневом каталоге web-сервиса

Получить файлы корневого каталога SDWL от технической поддержки ООО.БФТ.
Поместить файл ***bftDsudDownloader-fat.jar*** в каталог ***/opt/ensi_sdwl.***
Поместить файлы ***application.yml, logback.xml*** в каталог ***/opt/ensi_sdwl/config.***

#### 1.3.9.5. Редактирование файла application.yml

Взять на редактирование файл /opt/ensi_sdwl/config/application.yml, сверить настройки:

```
ktor:
  development: true
  deployment:
    port: "8085"
    host: "0.0.0.0"
  application:
    modules:
      - com.bftcom.dsud.downloader.application.ServerKt.module

database:
  jdbcUrl: "jdbc:postgresql://10.26.124.222:5432/db_ensi25?currentSchema=sdwl"
  driverClassName: "org.postgresql.Driver"
  username: "db_user"
  password: "********"
  maximumPoolSize: 10

connections:
  - id: "cnfg-main"
    type: "cnfg"
```

```
  url:  "https://10.26.124.233:8443/app"
  user: uploader
  pass: *******
  apiVersion: ""
  tokenAuthEnable: true
  tokenLife: 3600000
  addressLogin: "/api/auth/login"
  addressInsert: "/apis/sDWL/egryul_id"
  sink:
   sendBatchSize: 1000
  httpClient:
   maxConnectionsCount: 100
   socketTimeout: 300

updateTaskStatusIntervalSeconds: 10
```

#### 1.3.9.6. Установка прав на заданную папку

```
chown -R bft-link:bft-sdwl /opt/ensi_sdwl
```

#### 1.3.9.7. Создание файла systemd-юнита для запуска в качестве сервиса

```
sudo vi /etc/systemd/system/ensi_sdwl.service
----------------------Начало скрипта ----------------------
# Systemd unit file for sDWL
[Unit]
Description="Service for ensi_sdwl"
After=syslog.target

[Service]
Type=simple

WorkingDirectory=/opt/ensi_sdwl
```

*User=bft-sdwl*

*Group=bft-sdwl*

*RestartSec=10*

*Restart=always*

*Environment='JAVA_HOME=/usr/lib/jvm/jdk-17.0.12'*

*Environment='CATALINA_OPTS=-Xmx6G -server -XX:+UseParallelGC'*

*ExecStart=/usr/bin/java *

*-Dfile.encoding=UTF-8 *

*-Djava.awt.headless=true *

*-Djava.io.tmpdir=/opt/ensi_sdwl/temp *

*-Djava.security.egd=file:/dev/./urandom -Dlogback.configurationFile=/opt/ensi_sdwl/config/logback.xml *

*-jar /opt/ensi_sdwl/bftDsudDownloader-fat.jar -config=/opt/ensi_sdwl/config/application.yml*

*ExecStop=/bin/kill -15 $MAINPID*

*[Install]*

*WantedBy=multi-user.target*

*----------Конец скрипта----------*

#### 1.3.9.8. Запуск сервиса

*systemctl daemon-reload*

*systemctl start ensi_sdwl.service*

### 1.3.10. Установка и настройка SPK25

#### 1.3.10.1. Создание папки для работы приложения

*mkdir -p /opt/_Tomcat/spk25*

#### 1.3.10.2. Установка Tomcat

Установить Tomcat в папку **/opt/_Tomcat/spk25** (см. пункт 1.3.5)

#### 1.3.10.3. Удаление содержимого папки /opt/_Tomcat/spk25/webapps

Удалить содержимое папки.

***rm -r /opt/_Tomcat/spk25/webapps/****

#### 1.3.10.4. Размещения файла сборки «app.war» в корневом каталоге web-сервиса

Получить файл сборки «app.war» от технической поддержки ООО.БФТ.

Поместить файл ***«app.war»*** в каталог ***/opt/_Tomcat/spk25/webapps***

#### 1.3.10.5. Редактирование файл catalina.properties

Взять на редактирование файл ***/opt/_Tomcat/spk25/conf/catalina.properties,*** сверить настройки:

***package.access=sun.,org.apache.catalina.,org.apache.coyote.,org.apache.jasper.,org.apache.tomcat.***

***package.definition=sun.,java.,org.apache.catalina.,org.apache.coyote.,***

***org.apache.jasper.,org.apache.naming.,org.apache.tomcat.***

***common.loader="${catalina.base}/lib","${catalina.base}/lib/*.jar","${catalina.home}/lib","${catalina.home}/lib/*.jar"***

***server.loader=***

***shared.loader=***

***tomcat.util.scan.StandardJarScanFilter.jarsToSkip=***

***annotations-api.jar,***

***ant-junit*.jar,***

***ant-launcher*.jar,***

***ant*.jar,***

***asm-*.jar,***\
***aspectj*.jar,***\
***bcel*.jar,***\
***biz.aQute.bnd*.jar,***\
***bootstrap.jar,***\
***catalina-ant.jar,***\
***catalina-ha.jar,***\
***catalina-ssi.jar,***\
***catalina-storeconfig.jar,***\
***catalina-tribes.jar,***\
***catalina.jar,***\
***cglib-*.jar,***\
***cobertura-*.jar,***\
***commons-beanutils*.jar,***\
***commons-codec*.jar,***\
***commons-collections*.jar,***\
***commons-compress*.jar,***\
***commons-daemon.jar,***\
***commons-dbcp*.jar,***\
***commons-digester*.jar,***\
***commons-fileupload*.jar,***\
***commons-httpclient*.jar,***\
***commons-io*.jar,***\
***commons-lang*.jar,***\
***commons-logging*.jar,***\
***commons-math*.jar,***\
***commons-pool*.jar,***\
***derby-*.jar,***\
***dom4j-*.jar,***\
***easymock-*.jar,***\
***ecj-*.jar,***\
***el-api.jar,***\
***geronimo-spec-jaxrpc*.jar,***\
***h2*.jar,***\
***ha-api-*.jar,***\

***hamcrest-*.jar,***\
***hibernate*.jar,***\
***httpclient*.jar,***\
***icu4j-*.jar,***\
***jasper-el.jar,***\
***jasper.jar,***\
***jaspic-api.jar,***\
***jaxb-*.jar,***\
***jaxen-*.jar,***\
***jaxws-rt-*.jar,***\
***jdom-*.jar,***\
***jetty-*.jar,***\
***jmx-tools.jar,***\
***jmx.jar,***\
***jsp-api.jar,***\
***jstl.jar,***\
***jta*.jar,***\
***junit-*.jar,***\
***junit.jar,***\
***log4j*.jar,***\
***mail*.jar,***\
***objenesis-*.jar,***\
***oraclepki.jar,***\
***org.hamcrest.core_*.jar,***\
***org.junit_*.jar,***\
***oro-*.jar,***\
***servlet-api-*.jar,***\
***servlet-api.jar,***\
***slf4j*.jar,***\
***taglibs-standard-spec-*.jar,***\
***tagsoup-*.jar,***\
***tomcat-api.jar,***\
***tomcat-coyote.jar,***\
***tomcat-dbcp.jar,***\
***tomcat-i18n-*.jar,***\

*tomcat-jdbc.jar,*

*tomcat-jni.jar,*

*tomcat-juli-adapters.jar,*

*tomcat-juli.jar,*

*tomcat-util-scan.jar,*

*tomcat-util.jar,*

*tomcat-websocket.jar,*

*tools.jar,*

*unboundid-ldapsdk-*.jar,*

*websocket-api.jar,*

*wsdl4j*.jar,*

*xercesImpl.jar,*

*xml-apis.jar,*

*xmlParserAPIs-*.jar,*

*xmlParserAPIs.jar,*

*xom-*.jar*

*tomcat.util.scan.StandardJarScanFilter.jarsToScan=*

*log4j-taglib*.jar,*

*log4j-web*.jar,*

*log4javascript*.jar,*

*slf4j-taglib*.jar*

*tomcat.util.buf.StringCache.byte.enabled=true*

*org.apache.el.GET_CLASSLOADER_USE_PRIVILEGED=false*

*ice.rls.enabled = false*

*spring.profiles.active=postgresql,combinator*

*ice.consistency.enabled=false*

*spring.datasource.url=jdbc:postgresql://10.26.124.222:5432/db_ensi25?currentSchema=sdwl*

*spring.datasource.username=db_user*

*spring.datasource.password=*******

#### 1.3.10.6. Установка прав на заданную папку

*chown -R tomcat:tomcat /opt/_Tomcat/spk25*

#### 1.3.10.7. Создание файла systemd-юнита для запуска в качестве сервиса

*sudo vi /etc/systemd/system/spk25.service*

*----------Начало скрипта ----------*

*# Systemd unit file for Tomcat – Application*

*[Unit]*

*Description=Apache Tomcat Web Application Container*

*After=syslog.target network.target*

*[Service]*

*Type=forking*

*### OpenJDK – JRE*

*Environment='JAVA_HOME=/usr/lib/jvm/jdk-17*

*Environment='CATALINA_PID=/opt/_Tomcat/spk25/temp/tomcat.pid'*

*Environment='CATALINA_HOME=/opt/_Tomcat/spk25'*

*Environment='CATALINA_BASE=/opt/_Tomcat/spk25'*

*Environment='CATALINA_OPTS=-Xmx4G -server -XX:+UseParallelGC'*

*Environment='JAVA_OPTS=--add-opens*
*java.xml/com.sun.org.apache.xerces.internal.util=ALL-UNNAMED *
*--add-opens java.desktop/sun.font=ALL-UNNAMED*

*WorkingDirectory=/opt/_Tomcat/spk25*

*ExecStart=/opt/_Tomcat/spk25/bin/startup.sh*

*ExecStop=/opt/_Tomcat/spk25/bin/shutdown.sh*

*User=tomcat*

*Group=tomcat*

*UMask=0007*

*RestartSec=10*

*Restart=always*

*[Install]*

*WantedBy=multi-user.target*

*----------Конец скрипта----------*

#### 1.3.10.8. Запуск сервиса

***systemctl daemon-reload***

***systemctl start spk25.service***

ПРИЛОЖЕНИЕ 2

УСТАНОВКА И НАСТРОЙКА СЭП

Установка и настройка СЭП состоит из следующих этапов:

- Установка приложения JAR (на примере RedOS 7.3);
- Настройка КриптоПро JCP;
- Настройка портов брандмауэра;
- Настройка логирования;
- Настройка СЭП;
- Настройка УЦ.

## 2.1. Установка приложения JAR (на примере RedOS 7.3)

Установка приложения JAR (на примере RedOS 7.3) состоит их следующих шагов:

а) Устанавливаем из репозитория ОС JDK:
***sudo dnf install java-17-openjdk***

б) Устанавливаем версию java-17 по умолчанию:
***sudo update-alternatives --config java***

в) Проверяем правильность установки:

***java -version***

г) В конфигурационном файле Java /usr/lib/jvm/jdk-17/conf/security/java.security (путь размещения может быть другим: не jre-17, например, а java-17) вносим правки, после строки security.provider.12=SunPKCS11 добавляем строки:

***security.provider.13=JCP***
***security.provider.14=JCSP***
***security.provider.15=Crypto***
***security.provider.16=RevCheck***

д) Создаём каталог приложения и системного пользователя:

***sudo mkdir -p /opt/eds***
***sudo groupadd eds_user***
***sudo useradd -M -d /opt/eds --system eds_user***

***sudo chmod -R 770 /opt/eds***

***sudo find /opt/eds -type d -exec chmod 755 {} \;***

***sudo find /opt/eds -type f -exec chmod 644 {} \;***

е) Копируем в каталог приложения само приложение, там же создаём каталог для логов, системный файлы:

***sudo cp eds.jar /opt/eds***

***sudo cd /opt/eds***

***sudo mkdir /opt/eds/logs***

***touch application.yaml***

***touch logback.xml***

***touch hikari.properties***

***sudo chown -R eds_user:eds_user /opt/eds***

ж) Создаём в каталоге приложения /opt/eds файлы конфигураций - application.yaml:

```
application:
  version: "@project.version@"

server:
  port: 8090
  contextPath: /eds
  use-forward-headers: true

# Base64 encoded hash значение строки name:password администратора
приложения
authentication:
  admin: C9PZ0eFB033XF121x4Cu+1lbMlpszJ1Q5JaV09cxjYA=

hikari:
  properties-path: /opt/eds/hikari.properties

spring:
  messages:
    basename: i18n/messages
  h2:
```

```
  console:
    enabled: true
 jackson:
  default-property-inclusion: non_empty
 http:
  multipart:
    location: /opt/eds
    max-file-size: -1
    max-request-size: -1
#     max-file-size: 500MB
#     max-request-size: 500MB

rest:
 timeoutMSec: 5000

crlDownload:
 # Обновлять ли crl при проверке сертификата на валидность, если crl не актуален на текущую дату. По-умолчанию - true
 crlUpdateByRequest: true
 # Загружать ли в БД crl при проверке сертификата на валидность, если crl отсутствует. По-умолчанию - true
 crlDownloadByRequest: true
 # Работает ли автоматическая загрузка CRL по scheduler-у.
 crlDownloadEnabled: true
 # Период времени (msec), через который стартует автоматическая загрузка CRL по scheduler-у.
 crlDownloadDelayMsec: 20000
 # Максимальное число CRL, к-е грузятся за один старт загрузки CRL по scheduler-у.
 # Если загрузки ожидает большее число CRL, то оставшиеся загрузятся при следующем старте через период crlDownloadDelayMsec.
 crlDownloadCount: 200
 # Максимальное количество попыток загрузки CRL по скедулеру
 attemptsMaxCount: 3
```

```
  # Таймаут на скачивание мсек
  timeoutMs: 5000
  # Количество потоков для скачивания CRL (должно быть меньше чем размер пула соединений БД)
  threadCount: 5
  defaultCrlTtlHours: 24
  safeExpiryTime: 60

settings:
  reloadDelayMsec: 15000

crlResetAttemptsDownload:
  # Работает ли автоматический сброс попыток скачивания CRL.
  crlResetAttemptsEnabled: true
  # Период времени (msec), через который стартует автоматический сброс попыток скачивания CRL.
  crlResetAttemptsDelayMsec: 200000
  # Максимальное число CRL для которых сбрасываем кол-во попыток скачивания.
  # Если сброса попыток ожидает большее число CRL, то оставшиеся сбросятся при следующем старте через период crlResetTryDelayMsec.
  crlResetAttemptsCount: 300

# Адрес TSA
# TSAAddress: http://testca2012.cryptopro.ru/tsp/tsp.srf
TSAAddress: https://www.cryptopro.ru/tsp/tsp.srf

## SpringBoot
management:
  endpoints:
    web:
      exposure:
        include: '*'
  endpoint:
```

```
    health:
      show-details: always
endpoints:
  logfile:
    external-file: '/opt/eds/logs/eds.startup.out'
  sensitive: false
# если надо подключить к Springboot Admin
spring.boot.admin.client:
  url: 'http://srv-ice-tmc-d1.bft.local:1111'
  instance:
    service-url: 'http://astra17se.bft.local:8090/eds01'
    name: 'SEP test (astra17se:8090)'
  username: '*******'
  password: '*******'
logging:
  file: '/opt/eds/logs/eds.startup.out'
  pattern:
    file: '%clr(%d{yyyy-MM-dd HH:mm:ss.SSS}){faint} %clr(%5p)
%clr(${PID}){magenta} %clr(---){faint} %clr([%15.15t]){faint} %clr(%-
40.40logger{39}){cyan} %clr(:){faint} %m%n%wEx'

cacerts:
  password: changeit

fakeCaRegisterAllowed: true

# Включение Сваггера. При работе в промышленном контуре рекомендуется
отключать.
eds:
  enable:
    swagger: true

# ключ для шифрования паролей к контейнерам (ниже - подробности)
encryption:
```

***type: KEY***

***key: AF63B4D843E6102E9EB3F08109C1636C***

Ключи конфигурационного файла для настройки типа шифрования:

| Ключ | Пример значения | Значение по умолчанию | Комментарий |
|---|---|---|---|
| encryption.type | AES | AES | Устанавливает тип шифрования.. |
| encryption.key | AF63B4D843E6102E 9EB3F08109C1636C | - | Параметр необходим только при установке типа шифрования - encryption.type: KEY. Может быть произвольной строкой символов, которая будет является ключом для дальнейшего шифрования паролей. |

**- hikari.properties** (создание роли и БД описано ниже):

***driverClassName=org.postgresql.Driver***

***jdbcUrl=jdbc:postgresql://192.168.1.7:5432/db_sep***

***username=u_sep***

***password=*********

***maximumPoolSize=15***

***leakDetectionThreshold=20000***

***connectionTimeout=1000***

**- logback.xml:**

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration scan="true">
   <property name="LOG_FILE_NAME" value="/opt/eds/logs/eds" />
   <appender
name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
      <file>${LOG_FILE_NAME}.log</file>
      <encoder>
```

```
        <pattern>%-120(%d %level [%thread] [%logger{20}] [%X{request_id}]
[%X{trace_id}]) - %msg%n</pattern>
      </encoder>
      <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
        <fileNamePattern>${LOG_FILE_NAME}.%d{yyyy-MM-
dd}.%i.log</fileNamePattern>
        <timeBasedFileNamingAndTriggeringPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">
          <maxFileSize>50MB</maxFileSize>
        </timeBasedFileNamingAndTriggeringPolicy>
        <maxHistory>14</maxHistory>
        <totalSizeCap>1GB</totalSizeCap>
      </rollingPolicy>
      <append>true</append>
    </appender>
    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
      <encoder>
        <pattern>%-120(%date{YYYY-MM-dd HH:mm:ss.SSS} %level [%thread]
[%logger{20}] [%X{request_id}] [%X{trace_id}]) - %msg%n</pattern>
      </encoder>
    </appender>
    <logger name="RestSniffer" level="DEBUG" />
    <logger name="java.util.prefs" level="DEBUG" />
    <logger name="com.bftcom.eds" level="DEBUG" />
    <root level="DEBUG">
      <appender-ref ref="STDOUT" />
      <appender-ref ref="FILE" />
    </root>
</configuration>
```

з) Создаём systemd-сервис **/etc/systemd/system/eds.service**:

```
[Unit]
Description=EDS
# Documentation=
Requires=network.target remote-fs.target
```

```
After=network.target remote-fs.target

[Service]
Environment='JAVA_HOME=/usr/lib/jvm/jre-17'
Environment='PID_FILE=/var/run/eds.pid'

Type=simple
User=eds_user
Group=eds_user
WorkingDirectory=/opt/eds
# ExecStart=/usr/lib/jvm/jre-17/bin/java -Xms2G -Xmx5G -Xmn1G -
XX:+UseParallelGC -Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=9010 -
Dcom.sun.management.jmxremote.rmi.port=9010 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -
Dspring.config.location=/opt/eds/application.yaml -
Dlogging.config=/opt/eds/logback.xml -jar /opt/eds/eds.jar
ExecStart=/usr/lib/jvm/jre-17/bin/java -Xms2G -Xmx5G -Xmn1G -XX:+UseParallelGC
-Dspring.config.location=/opt/eds/application.yaml -
Dlogging.config=/opt/eds/logback.xml -jar /opt/eds/eds.jar
ExecStop=/bin/kill -15 $MAINPID
Restart=on-abnormal
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=eds_service

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

Параметры памяти -Xms, -Xmx, -Xmn подбираются исходя из ресурсов сервера.

и) На целевом сервере СУБД PostgreSQL создаём роли и БД для СЭП:

***CREATE ROLE bfteds_user NOLOGIN ENCRYPTED***

***PASSWORD 'md567ba111a82b6a5fb71273ba1b14ddf71' NOSUPERUSER INHERIT***

***NOCREATEDB NOCREATEROLE NOREPLICATION;***

***CREATE USER u_sep WITH LOGIN NOSUPERUSER;***

***\password u_sep***

***CREATE DATABASE db_sep OWNER u_sep;***

***COMMENT ON DATABASE db_sep IS 'EDS';***

***GRANT bfteds_user TO u_sep;***

Примечание: Если для авторизации в СУБД PostgreSQL используется алгоритм scram-sha-256 (является алгоритмом «по умолчанию», начиная с версии 13), то для пользователя u_sep необходимо задать хэш пароля по алгоритму md5 (СЭП авторизуется только по md5):

***# проверяем используемый алгоритм (или смотрим в postgres.conf)***

***SHOW password_encryption;***

***# если в выводе будет 'scram-sha-256'***

***SET password_encryption='md5';***

***CREATE USER u_sep WITH LOGIN NOSUPERUSER;***

***\password u_sep***

***SET password_encryption='scram-sha-256';***

***# можно проверить***

***SELECT rolname, rolpassword FROM pg_authid WHERE rolname IN***

***('postgres','u_sep') \gx***

и в pg_hba.conf для этого прописываем (перед 0.0.0.0/0, если он есть):

***host db_sep u_sep 192.168.1.12/32 md5***

где 192.168.1.12 – адрес сервера, где развёрнут СЭП.

к) Запустить приложение, добавить в автозапуск:

***sudo systemctl start eds.service***

***sudo systemctl status eds.service***

***sudo systemctl enable eds.service***

Если будет использоваться список УЦ, прогружаем его с сайта МинЦифры (и затем для постоянного периодического обновления - добавляем в cron):

***curl -X POST "http://127.0.0.1:8090/eds/api/v1.0/authority?url=https%3A%2F%2Fe-***

***trust.gosuslugi.ru%2FCA%2FDownloadTSL%3F?schemaVersion%3D0" -H "accept:***

****/*"***

Для подключения к swagger-у СЭП требуется в браузере указать строку, составленную по шаблону:

***http://[имя_хоста]:8090/eds/swagger-ui.html***

При обновлении JRE требуется переустановить CryptoPro JCP и снова поправить вышеуказанные параметры в /usr/lib/jvm/jre-1.8.0/lib/security/java.security.

По умолчанию файл доверенных сертификатов /etc/pki/ca-trust/extracted/java/cacerts имеет установленные разрешения 444 ("только чтение" для всех) и root:root в качестве владельца. Поэтому функционал СЭП, позволяющий записывать в cacerts новые доверенные сертификаты работать не будет (возникает ошибка "Отказано в доступе"). Поэтому на данный файл надо скорректировать разрешения:

***sudo chown root:eds_user /etc/pki/ca-trust/extracted/java/cacerts***

***sudo chmod 464 /etc/pki/ca-trust/extracted/java/cacerts***

## 2.2. Настройка КриптоПро JCP

Примечание: Лицензия КриптоПро JCP в данной установке является демонстрационной, действует 3 месяца. По истечению этого срока, следует приобрести и установить официальную версию.

Команды для работы с лицензиями КриптоПро JCP:

***# требования к лицензии на данном сервере***

***/usr/lib/jvm/jre/bin/java ru.CryptoPro.JCP.tools.License -required***

***# проверка установленных***

***/usr/lib/jvm/jre/bin/java ru.CryptoPro.JCP.tools.License***

***# дата первой установки***

***/usr/lib/jvm/jre/bin/java ru.CryptoPro.JCP.tools.License -first***

***# проверка заданной лицензии***

***java -cp .:*: ru.CryptoPro.JCP.tools.License -serial "serial_number" -company "company_name"***

***java -cp .:*: ru.CryptoPro.JCP.tools.License -serial "serial_number" -combase "company_name_in_base64"***

***# проверка заданной лицензии и ее сохранение***

***java -cp .:*: ru.CryptoPro.JCP.tools.License -serial "serial_number" -company "company_name" -store***

***java -cp .:*: ru.CryptoPro.JCP.tools.License -serial "serial_number" -combase "company_name_in_base64" -store***

***# справка по командам***

***/usr/lib/jvm/jre/bin/java ru.CryptoPro.JCP.tools.License ?***

Количество ядер, на которое приобретена лицензия, проверяется при каждом запуске, поэтому в строку запуска приложения, использующего **JCP** необходимо добавить параметр **-XX:ActiveProcessorCount=1**, в качестве значения которого, необходимо указать лицензированное количество ядер (в данном примере - 1 ядро). Этот же параметр должен указываться при любых действиях, выполняемых с использованием JCP, иначе будет ошибка лицензии.

Для того, чтобы скрипт работал нужен дистрибутив **КриптоПро JCP** или **JCSP**, т.к. интегрированный в приложение СЭП недоступен для данного применения: он используется только в процессе работы самого приложения. Поэтому необходимо в каталог **/opt/jcp** распаковать **КриптоПро JCP** или **JCSP** (пример для установки СЭП в сети БФТ):

***mkdir -p /opt/jcp***

***wget https://srv-nexus-3.bftcom.com/repository/devops/cryptopro/jcsp/5.0/java-csp-5.0.45549-A-56fe5758.tar.gz***

***tar -zxvf java-csp-5.0.*.tar.gz --strip-components=1 -C /opt/jcp***

Для активации лицензии необходимо:

а) создать каталог active, выполнить команду:

***sudo mkdir -p /opt/eds/active***

б) открыть каталог /opt/eds/active, создать файл .env, выполнить команды:

***cd /opt/eds/active***

***touch .env***

в) открыть файл .env, вставить следующее содержимое:

***PROCESSOR_COUNT="1"***

***LICENSE_NUMBER="XXXXX-XXXXX-XXXXX-XXXXX-XXXXX"***

***COMPANY_NAME="<COMPANY_NAME>"***

***USER_NAME="<USER_NAME>"***

Примечание:

– PROCESSOR_COUNT – количество ядер с процессора, с приобретенной лицензией (из расчета 1 ядро=1 лицензия);

– LICENSE_NUMBER – номер лицензии;

– COMPANY_NAME – название юридического лица;

– USER_NAME – учетная запись, под которой будет запускаться приложение.

г) открыть каталог /opt/eds/active, создать файл jcp-license.service, выполнить команды:

***cd /opt/eds/active***

***touch jcp-license.service***

д) открыть файл jcp-license.service, вставить следующее содержимое:

***[Unit]***

***Description=JCP license set***

***[Service]***

***Type=oneshot***

***### Переменные***

***Environment='JAVA_HOME=/usr/lib/jvm/jre-17'***

***User=USER_NAME***

***# Распакованный дистрибутив JCP***

***WorkingDirectory=/opt/jcp***

***### Вывод информации об применённой/установленной лицензии***

***ExecStart=/usr/lib/jvm/jre-17/bin/java -***

***XX:ActiveProcessorCount=PROCESSOR_COUNT -cp .:*:***

***ru.CryptoPro.JCP.tools.License***

***### Проверка первой активации лицензии на этом сервере***

***#ExecStart=/usr/lib/jvm/jre-17/bin/java -***

***XX:ActiveProcessorCount=PROCESSOR_COUNT -cp .:*:***

***ru.CryptoPro.JCP.tools.License -first***

***### Проверка заданной лицензии***

***#ExecStart=/usr/lib/jvm/jre-17/bin/java -***

***XX:ActiveProcessorCount=PROCESSOR_COUNT -cp .:*:***

***ru.CryptoPro.JCP.tools.License -serial "LICENSE_NUMBER" -company***

***"COMPANY_NAME"***

***### Проверка заданной лицензии и ее сохранение***

***#ExecStart=/usr/lib/jvm/jre-17/bin/java -***

***XX:ActiveProcessorCount=PROCESSOR_COUNT -cp .:*:***

***ru.CryptoPro.JCP.tools.License -serial "LICENSE_NUMBER" -company***

***"COMPANY_NAME" -store***

***[Install]***

***WantedBy=default.target***

Примечание:

– Environment – путь каталогу Java17 (используемая Java);

– WorkingDirectory – каталог с распакованным дистрибутивом JCP (дистрибутив предоставляется Заказчиком);

– ExecStart – раскомментировать необходимый вариант ExecStart (см. комментарии в файле jcp-license.service).

е) открыть каталог /opt/eds/active, создать файл jcp-license.sh, выполнить команду:

***cd /opt/eds/active***

***touch jcp-license.sh***

ж) открыть файл jcp-license.sh, вставить следующее содержимое:

***#!/usr/bin/env bash***

***export $(grep -v '^#' .env | xargs)***

***cp ./jcp-license.service /etc/systemd/system/***

***sed -i 's/PROCESSOR_COUNT/'${PROCESSOR_COUNT}'/' /etc/systemd/system/jcp-license.service***

***sed -i 's/LICENSE_NUMBER/'${LICENSE_NUMBER}'/' /etc/systemd/system/jcp-license.service***

***sed -i 's/COMPANY_NAME/'${COMPANY_NAME}'/' /etc/systemd/system/jcp-license.service***

***sed -i 's/USER_NAME/'${USER_NAME}'/' /etc/systemd/system/jcp-license.service***

***systemctl daemon-reload***

***systemctl start jcp-license.service***

***journalctl -u jcp-license.service > ./jcp-license.log***

***#rm -f /etc/systemd/system/jcp-license.service***

***systemctl daemon-reload***

- выдать права на каталоги, выполнить команды:

***sudo chown -R eds_user:eds_user /opt/jcp***

***sudo chown -R eds_user:eds_user /opt/eds/active***

- запустить файл jcp-license.sh, выполнить команду:

***sh jcp-license.sh***

- открыть каталог /opt/eds/active, проверить лог в файле jcp-license.log.

## 2.3. 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 application.yaml в директории eds - в параметре port тэга server) принимать запросы извне.

***# добавляем требуемый порт 8091***

***firewall-cmd --permanent --service=http --add-port=8091/tcp***

***# если не нужен - удаляем неиспользуемый HTTP-порт, например, 80 (порт "по умолчанию")***

***firewall-cmd --permanent --service=http --remove-port=80/tcp***

***# добавляем конфигурацию сервиса http к разрешённым***

***firewall-cmd --permanent --add-service=http***

Для применения правил необходимо их перезагрузить:

***firewall-cmd --reload***

## 2.4. Настройка СЭП

Предварительно в СЭП необходимо загрузить актуальный справочник аккредитованных удостоверяющих центров.

Для этого используется метод /api/v1.0/authority. В параметре URL надо указать:

***https://e-trust.gosuslugi.ru/CA/DownloadTSL?schemaVersion=0***

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/authority?url=https%3A%2F%2Fe-trust.gosuslugi.ru%2FCA%2FDownloadTSL%3FschemaVersion%3D0" -H "accept: */*"***

Для подписания с авторизацией по сертификату (методы группы signature-generator-auth-by-cert-controller) необходимо зарегистрировать на стенде СЭП сертификат (открытый ключ) и его контейнер (закрытый ключ):

а) Добавить сертификат (метод /api/v1.0/addCert):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/addCert" -H "accept: */*" -H "Content-Type: multipart/form-data" -F "certificate=@AEKSA256.crt;type=application/x-x509-ca-cert"***

б) Регистрация контейнера/приватного ключа (предварительно контейнер должен быть размещён в хранилище JCP) (метод /api/v1.0/admin/registerKey):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/admin/registerKey?keyAlias=AEKSA256" -H "accept: */*" -H "Authorization: Basic YWRtaW46cGF2bGlu"***

в) Привязка сертификата/публичного ключа к контейнеру/приватному ключу (метод /api/v1.0/linkKeyToContainer):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/linkKeyToContainer?containerId=3&keyId=15079" -H "accept: */*"***

Если сертификат был выдан аккредитованным удостоверяющим центром, то его привязка к УЦ на уровне базы будет осуществлена автоматически.

Для самоподписанных сертификатов потребуется их привязка к специально заведённому в базе УЦ (в дальнейшем эта необходимость будет устранена).

## 2.5. Настройка УЦ

Для настройки УЦ надо выполнить шаги:

а) Заводим УЦ (метод /api/v1.0/addAuthority):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/addAuthority?name=CAforSelfSignedCerts&ogrn=112233" -H "accept: */*"***

б) Добавляем новое событие в историю аккредитации УЦ (делаем УЦ активным) (метод /api/v1.0/addEvent):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/addEvent?authId=3927&state=ACTIVE&validFromEpoch=1000000000000" -H "accept: */*"***

в) Привязываем УЦ к публичному ключу (метод /api/v1.0/linkToKey):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/linkToKey?authId=3927&keyId=15079" -H "accept: */*"***

Имеется возможность подписания по строке авторизации (методы группы signature-generator-controller). Для этого в СЭП необходимо создать организацию, привязав её к

контейнеру/приватному ключу, а затем создать пользователя, привязав его к организации. Порядок шагов:

г) Ввод подразделения/организации (метод /api/v1.0/admin/insertOwner):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/admin/insertOwner?keyAlias=AEKSA256&ownerName=Org1Name" -H "accept: */*" -H "Authorization: Basic YWRtaW46cGF2bGlu"***

Через приложение администрирования: http://srv-esb-stand2:8081/app/#/owner

д) Создание пользователя для подразделения/организации (метод api/v1.0/admin/insertUser?ownerName):

***curl -X POST "http://srv-esb-stand2:8090/eds01/api/v1.0/admin/insertUser?ownerName=eds1&userName=eds1-2&userPassword=eds1-2" -H "accept: */*" -H "Authorization: Basic YWRtaW46cGF2bGlu"***

В ответе будет строка для авторизации.

Для выполнения методов администрирования СЭП можно установить свою строку авторизации вместо строки по умолчанию "Basic YWRtaW46cGF2bGlu".

Например, логин будет «admin1», пароль – «pass1». Из логина и пароля требуется составить строку: «admin1:pass1».

Захэшировать строку методом /api/v1.0/admin/encodedhash - будет выдан результат Tb0hUDXCVlNJ/QLri1v1fr/tr3MSIHv//2qGMVLYHls=.

Полученный хэш требуется указать в файле свойств СЭП в параметре «authentication:admin»:

***authentication:***

***admin: Tb0hUDXCVlNJ/QLri1v1fr/tr3MSIHv//2qGMVLYHls=***

Чтобы выполнить метод, требующий авторизации администратора (например, /api/v1.0/admin/registerKey), в параметре Authorization нужно будет указать:

***Basic YWRtaW4xOnBhc3Mx***

Здесь YWRtaW4xOnBhc3Mx – закодированная в Base64 строка admin1:pass1

ПРИЛОЖЕНИЕ 3

ИМПОРТ КОНФИГУРАЦИИ ОБЪЕКТОВ И ЗАПИСЕЙ

1. Для первоначального заполнения необходимо получить у службы технической поддержки конфигурацию объектов и записей для импорта (см. Раздел 3.7.10.2 документа «Руководство оператора: Администратор системы, Аналитик метаданных»).

2. Удалить полнотекстовые индексы в таблицах:

2.1. Открыть раздел «Администрирование» → «База данных»:

– в области с отображением таблиц базы данных, в поисковой строке указать одно из значений fns, fsa, egrip, clean, нажать Enter;

– в перечне отображенных таблиц базы данных, выделить по очереди каждую, открыть вкладку «Индексы», удалить индексы, содержащие постфикс _textindex (Рисунок 6), нажать на кнопку ;

– повторить действия со всеми перечисленными таблицами.

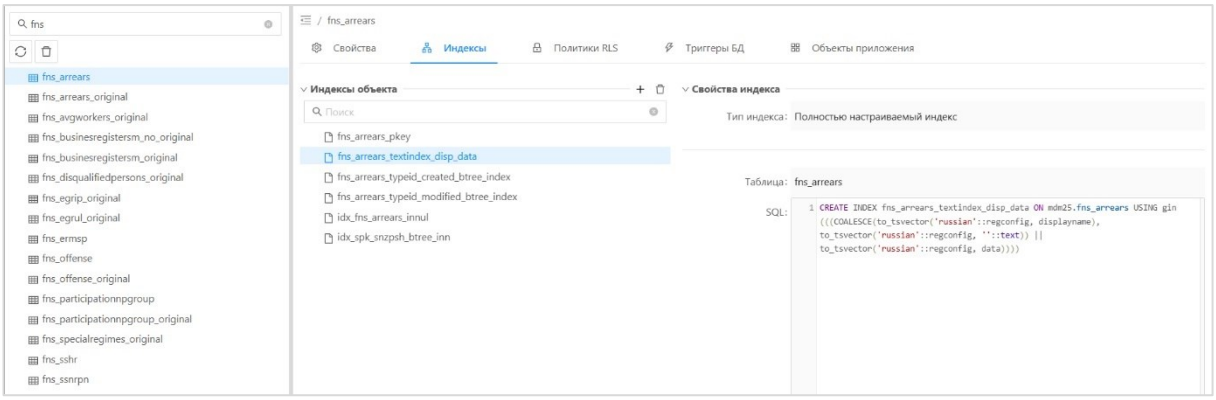


Рисунок 6 – Удаление индекса

3. Установить расширение, выполнить команду (подключится к базе данных):

***CREATE extension if not exists “uuid-ossp”;***

4. Создать объект базы данных последовательность, выполнить команду (подключится к базе данных):

```
do
$$
declare
   dicts text;
   seq text;
begin
```

```
    for dicts, seq in
        select descriptor -> 'dataModel' ->> 'tableName', descriptor -> 'dataModel' ->>
'tableName' || '_for_id_record_seq'
        from appobject a
        where moduleid = 'dwl_dict'
        and descriptor -> 'dataModel' ->> 'tableName' ilike '%origin%'
    loop
        execute format('CREATE SEQUENCE IF NOT EXISTS %I', seq);
        execute format('ALTER TABLE %s ALTER COLUMN id_record SET
DEFAULT nextval(%L)', dicts, seq);
        execute format('ALTER TABLE %s ALTER COLUMN id SET DEFAULT
public.uuid_generate_v4()', dicts);
    end loop;
end
$$;
```

5. Задать параметры в настройках загрузки:

5.1. Открыть раздел «Справочники» → «STAGE_0 (ТЕХНИЧЕСКИЕ СПРАВОЧНИКИ)» → «Контроль загрузки и обработки данных» → «Настройки загрузки sDWL», в правом верхнем углу нажать кнопку «Табличный вид».

5.2. Открыть настройку загрузки (Рисунок 7), одного из указанных справочников: «AVGworkers», «offense», «specialregimes», «arrears», «businesregisterSM», «rss», задать параметры:

- url – адрес сервиса sDWL;
- targetDbJdbcUrlString – адрес БД «БФТ.ЕНСИ»;
- targetDbdriverClassNameString – драйвер БД;
- targetDbusername – логин пользователя БД;
- targetDbpassword – пароль пользователя БД.

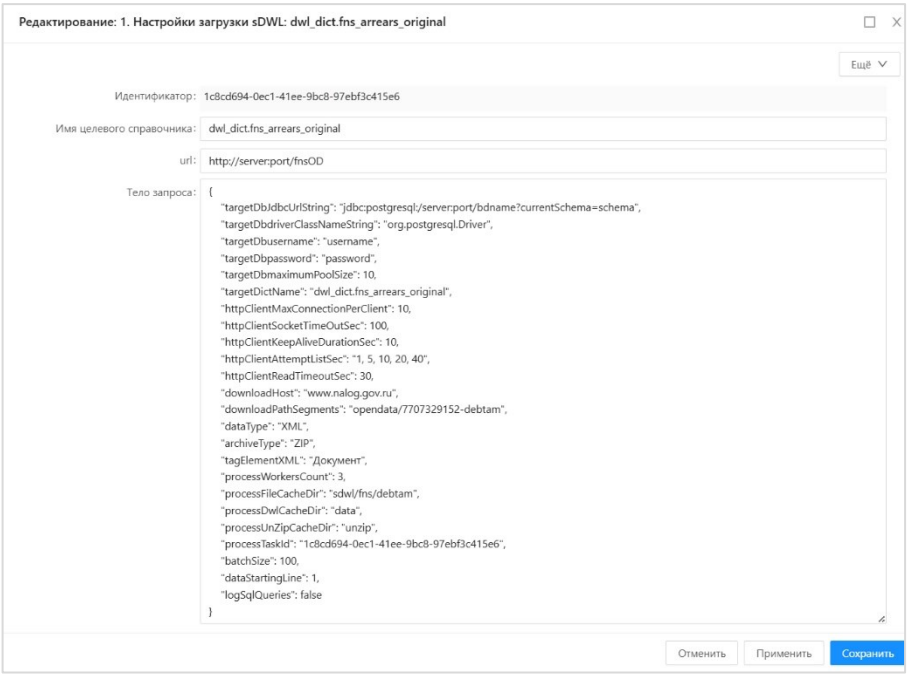


Рисунок 7 – Настройка загрузки

где:

– имя целевого справочника – имя справочника;

– url – адрес севиса SDWL;

– targetDbJdbcUrlString – адрес БД «БФТ.ЕНСИ»;

– targetDbdriverClassNameString – драйвер БД;

– targetDbusername – логин пользователя БД;

– targetDbpassword – пароль пользователя БД;

– targetDbmaximumPoolSize – размер пула соединений с БД;

– targetDictName – имя целевого справочника;

– httpClientMaxConnectionPerClient – максимальное количество соединений, открытых клиентом (в том числе в несколько потоков);

– httpClientSocketTimeOutSec – временной интервал ожидания ответа источника на запрос (в сек.);

– httpClientKeepAliveDurationSec – временной интервал, определяющий частоту направления sDWL пакетов подтверждения работоспособности к источнику (в сек.);

– httpClientAttemptListSec – список временных интервалов, определяющий количество повторных попыток направления запросов к источнику и таймаутов между такими попытками;

– httpClientReadTimeoutSec – временной интервал ожидания ответа источника на запрос (в сек.);

– downloadHost – хост источника данных;

– downloadPathSegments – сегменты URL для запроса данных в соответствии со структурой источника;

– datatype – тип данных в источнике данных;

– archiveType – тип архива в источнике данных;

– tagElementXML - параметр для раскладки xml;

– processWorkersCount – количество потоков для обработки параллельной загрузки страниц;

– processFileCacheDir – путь к каталогу для временных файлов загрузки;

– processDwlCacheDir – директория в processFileCacheDir для хранения исходных данных (архивов);

– processUnZipCacheDir – директория в processFileCacheDir для хранения разархивированных данных;

– processTaskId – UUID задачи для загрузки справочника;

– batchSize – определяет количество записей источника, которые sDWL размещает в БД в рамках одного sql-запроса;

– dataStartingLine – начальная строка данных;

– logSqlQueries – ведение журнала sql-запросов.

5.3. Открыть настройку загрузки, справочников «ЕГРЮЛ», «ЕГРИП» (Рисунок 8) и задать параметры:

– url – адрес сервиса sDWL;

– targetDbJdbcUrlString – адрес БД «БФТ.ЕНСИ»;

– targetDbdriverClassNameString – драйвер БД;

– targetDbusername – логин пользователя БД;

– targetDbpassword – пароль пользователя БД.

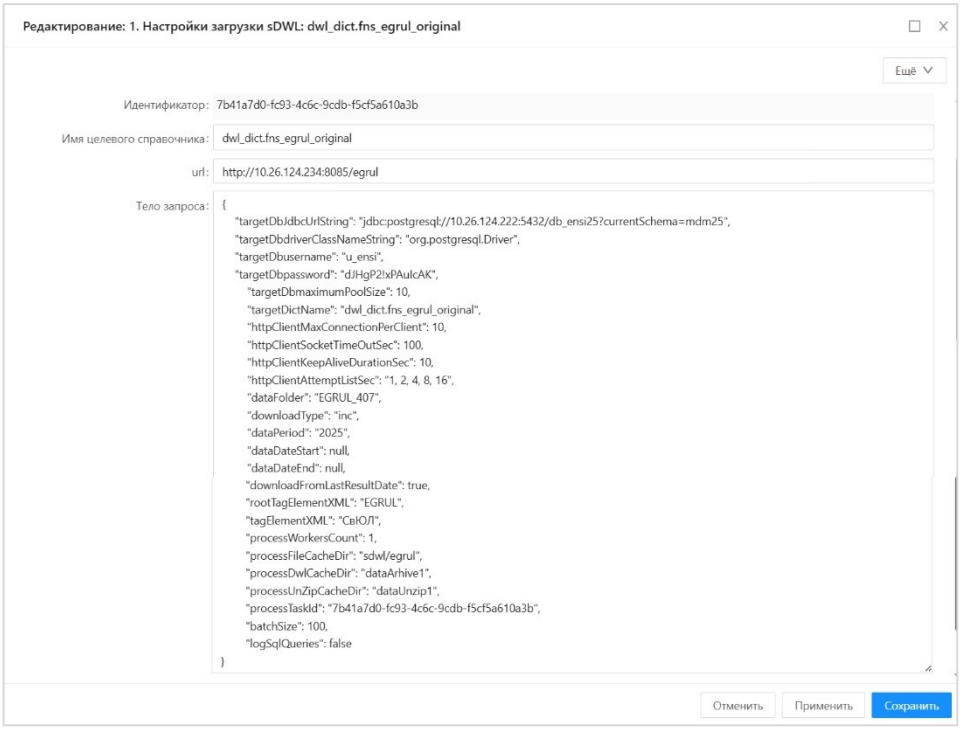


Рисунок 8 – Настройка загрузки справочника «ЕГРЮЛ»

где:

– имя целевого справочника – имя справочника;

– url – адрес сервиса sDWL;

– targetDbJdbcUrlString – адрес БД «БФТ.ЕНСИ»;

– targetDbdriverClassNameString – драйвер БД;

– targetDbusername – логин пользователя БД;

– targetDbpassword – пароль пользователя БД;

– targetDbmaximumPoolSize – размер пула соединений с БД;

– targetDictName – имя целевого справочника;

– httpClientMaxConnectionPerClient – максимальное кол-во соединений, открытых клиентом (в том числе в несколько потоков);

– httpClientSocketTimeOutSec – временной интервал ожидания ответа источника на запрос (в сек.);

– httpClientKeepAliveDurationSec – временной интервал, определяющий частоту направления sDWL пакетов подтверждения работоспособности к источнику (в сек.);

– httpClientAttemptListSec – список временных интервалов, определяющий количество повторных попыток направления запросов к источнику и таймаутов между такими попытками;

– httpClientReadTimeoutSec – временной интервал ожидания ответа источника на запрос (в сек.);

– dataFolder –наименование директории в источнике, из которой требуется забор данных;

– downloadType – тип загрузки данных (первичная или инкрементальная загрузка). Принимает одно из значений «full» или «inc»;

– dataPeriod - указывается год, за который необходимо получить с источника данные с типом «full»;

– dataDateStart – определяет начальную дату для отбора дельт на источнике, включая задаваемую дату;

– dataDateEnd – определяет конечную дату для отбора дельт на источнике, включая задаваемую дату;

– downloadFromLastResultDate – определяет порядок определения дат загрузки дельт с источника;

– tagElementXML – параметр, который нужен для раскладки xml;

– processWorkersCount – количество потоков для обработки параллельной загрузки страниц;

– processFileCacheDir – путь к каталогу для временных файлов загрузки;

– processDwlCacheDir – директория в processFileCacheDir для хранения исходных данных (архивов);

– processUnZipCacheDir – директория в processFileCacheDir для хранения разархивированных данных;

– processTaskId – UUID задачи для загрузки справочника;

– batchSize – определяет количество записей источника, которые sDWL размещает в БД в рамках одного sql-запроса;

– dataStartingLine – начальная строка данных;

– logSqlQueries – ведение журнала sql-запросов.

6. Провести первоначальную загрузку:

6.1. Для справочников «ЕГРЮЛ», «ЕГРИП»:

6.1.1. Загрузить полные данные за требуемый период (с начала года, по текущую дату):

– в параметре downloadType указать значение full;

– в параметре dataPeriod указать значение 2025.

6.1.2. Изменить настойки для загрузки дельты за год:

– в параметре downloadType указать значение inc;
– в параметре dataDateStart указать значение 01.01.2025;
– в параметре dataDateEnd указать значение «текущая дата минус 1 день»;
– в параметре downloadFromLastResultDate указать значение false.

6.1.3. Настроить ежедневные загрузки данных (не подлежат дальнейшей корректировке):

– в параметре downloadType указать значение inc;
– в параметре dataDateStart указать значение null;
– в параметре dataDateEnd указать значение null;
– в параметре downloadFromLastResultDate указать значение true.

Данная настройка будет осуществлять дозагрузку данных без ежедневной корректировки пользователем запроса. Инициировать загрузку, начиная со дня, следующего за днем загрузки инкремента.

6.2. Активировать автоматическую загрузку обновлений справочников «ЕГРЮЛ», «ЕГРИП»:

6.2.1. Открыть раздел «Настройки» → «Планировщик заданий».

6.2.2. Открыть задание «Загрузка. ФНС. ЕГРИП», передвинуть ползунок в статус «Активна», нажать кнопку «Сохранить» (Рисунок 9).

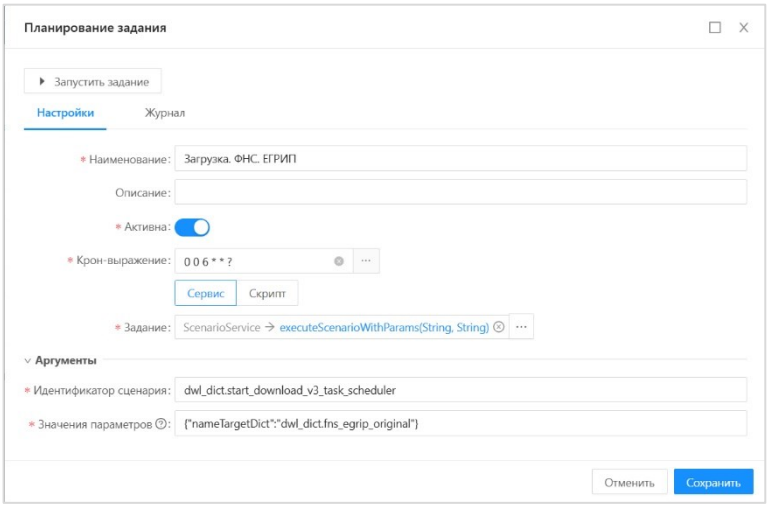


Рисунок 9 – Планировщик заданий

6.2.3. Открыть задание «Загрузка. ФНС. ЕГРЮЛ», передвинуть ползунок в статус «Активна», нажать кнопку «Сохранить».

6.3. Для остальных справочников сделать активными задания:

– «Загрузка. ФНС. ЕРМСП»;

– «Загрузка. ФНС. СНПМО»;

– «Загрузка. ФНС. ССНЗ»;

– «Загрузка. ФНС. ССНР»;

– «Загрузка. ФНС. ССЧР»;

– «Загрузка. ФСА. РСС».

7. Проверить статус загрузок:

– Открыть раздел «Справочники» → «STAGE_0 (ТЕХНИЧЕСКИЕ СПРАВОЧНИКИ)» → «Контроль загрузки и обработки данных» → «История запусков sDWL»;

– В правом верхнем углу нажать кнопку «Табличный вид»;

– Отключить фильтр «Незавершенные»;

– Отсортировать по колонке «Время начала»;

– Проверить загрузки, успешное прохождение стадий: инициализации, загрузки, обработки.

7.1. В случае появления ошибок, провести анализ логов сервиса «sDWL» и «БФТ.ЕНСИ».

ПРИЛОЖЕНИЕ 4
ОСТАНОВКА СИСТЕМЫ

## 4.1. Остановка Apache Tomcat

Остановка Apache Tomcat как зарегистрированного сервиса:
***systemctl stop tomcat-$NAME.service.***

## 4.2. Остановка СУБД PostgreSQL 15

Остановка приложения СУБД:
***systemctl stop postgresql-15***

## 4.3. Остановка приложения СЭП

Остановка приложения eds:
***systemctl stop eds.***

# Перечень терминов

| Термин | Определение термина |
|---|---|
| ActiveMQ | Свободный и открытый брокер сообщений, реализующий протокол JMS. |
| Apache Kafka | Распределенная платформа потоковой передачи данных с открытым исходным кодом. |
| Apache Tomkat | Свободный веб-сервер с открытым исходным кодом, разработанный Apache Software Foundation. |
| API | API (Application Programming Interface): Набор определений и протоколов, позволяющих программным компонентам взаимодействовать друг с другом. |
| CSV | CSV (Comma-Separated Values): Формат текстового файла, в котором данные разделены запятыми. |
| JSON | JSON (JavaScript Object Notation): Формат обмена данными, основанный на объектах JavaScript. |
| OpenJDK | Реализация Java Development Kit (JDK) с открытым исходным кодом, разработанная Oracle. |
| RabbitMQ | Это брокер сообщений с открытым исходным кодом, который реализует протокол Advanced Message Queuing Protocol (AMQP). Он позволяет приложениям передавать сообщения друг другу в асинхронном режиме, обеспечивая надежную и масштабируемую передачу данных. RabbitMQ часто используется для реализации архитектурных шаблонов, таких как очереди заданий, публикации-подписки и интеграции между системами. |
| REST | REST: Архитектурный стиль для проектирования веб-сервисов, который использует протокол HTTP. |
| sDWL | Веб-сервис загрузки данных с первоисточника |
| XML | XML (Extensible Markup Language): Язык разметки, используемый для определения данных в структурированном и иерархическом формате. |
| XSD | XSD (XML Schema Definition): Язык, который определяет структуру и элементы в XML-документах. |
| Брокер | Посредник, который маршрутизирует сообщения между производителями и потребителями. |
| Веб-сервер | Программное обеспечение, которое обрабатывает запросы от веб-браузеров и возвращает соответствующие веб-страницы. |
| Дифференциальный бэкап | Резервное копирование только тех частей данных, которые изменились с момента последнего полного бэкапа. |
| Лог | Файл, в который записываются события и сообщения, создаваемые системой или программным обеспечением. |
| Полный бэкап | Резервное копирование всех данных в системе. |
| Репозиторий | Централизованное хранилище версий кода и других артефактов. |

| Термин | Определение термина |
|---|---|
| Роль | Совокупность прав и возможностей пользователей. Для одного пользователя может быть определено несколько ролей. |
| Сервер баз данных | Программное обеспечение, которое управляет доступом к базе данных и обеспечивает ее работу. |
| Сервис | Набор операций, выполняемых для определенного программного компонента. |
| Сетевое хранилище | Система хранения данных, доступная через компьютерную сеть. |
| Технологический каталог | Инструмент для управления и организации разных программных компонентов. |
| Юнит | Небольшой независимый модуль программного обеспечения. |

# Перечень сокращений

| Сокращение | Расшифровка сокращения |
|---|---|
| ОС | Операционная система |
| ПО | Программное обеспечение |
| Программа, Система, «БФТ.ЕНСИ» | Единая система управления нормативно-справочной информацией «БФТ.ЕНСИ» версия 1.11 |
| СУБД | Система управления базами данных. |
| СЭП | Сервис электронной подписи |
| УЦ | Удостоверяющий центр |
| ЭЦП | Электронная цифровая подпись |

## Перечень рисунков

## Перечень таблиц