|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | | | |
|  |  |  | | | |
|  |  |  | | | |
|  |  |  | | | |
|  |  |  | | | |
| **Публичное акционерное общество «Интер РАО ЕЭС»** | | | | | |
| **АВТОМАТИЗАЦИЯ ПРОЦЕССОВ СИСТЕМЫ УПРАВЛЕНИЯ РИСКАМИ И ВНУТРЕННЕГО КОНТРОЛЯ И АНТИМОНОПОЛЬНОГО КОМПЛАЕНСА** | | | | | |
| **руководство системного администратора** | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | |  |  |
|  | | |  |  |
|  | | |  |  |
|  | | |  |  |
|  | | |  |  |
|  | | |  |  |
|  | | |  |  |
|  | | |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г. |
|  | | |  |  |
| г. Москва | | | | | |

предисловие

Настоящий документ является руководством системного администратора программы для ЭВМ «Система управления рисками и внутреннего контроля и антимонопольного комплаенса», состоящая из модуля «СУРиВК» и модуля «АМК (далее – Система).

Руководство администратора разработано в соответствии с рекомендациями ГОСТ 19.503-79.

СОДЕРЖАНИЕ

[Обозначения и сокращения 5](#_Toc123055504)

[1 Введение 7](#_Toc123055505)

[1.1 Область применения …………………………………………………………………………... 7](#_Toc123055506)

[1.2 Требования к квалификации администратора ………………………………………………. 7](#_Toc123055507)

[1.3 Перечень эксплуатационных документов …………………………………………………… 9](#_Toc123055508)

[2 Назначение и условия применения 10](#_Toc123055509)

[2.1 Обозначение и наименование………………………………………………………………….10](#_Toc123055510)

[2.2 Назначение системы …………………………………………………………………………...10](#_Toc123055511)

[2.3 Условия выполнения …………………………………………………………………………..10](#_Toc123055512)

[2.3.1 Минимальные требования к серверному оборудованию (высоконагруженный режим работы) 12](#_Toc123055513)

[2.3.2 Требования к оборудованию рабочих станций 12](#_Toc123055514)

[3 Подготовка к работе 14](#_Toc123055515)

[3.1 Установка ПО Triafly на ОС Astra Linux Common Edition………………………………….14](#_Toc123055516)

[3.1.1 Обозначение версии дистрибутива ИС 15](#_Toc123055517)

[3.1.2 Подготовка Системы 15](#_Toc123055518)

[3.1.3 Установка необходимых приложений и библиотек 15](#_Toc123055519)

[3.1.4 Установка кодировки 19](#_Toc123055520)

[3.1.5 Настройка Apache2 19](#_Toc123055521)

[3.1.6 Создание пользователя 20](#_Toc123055522)

[3.1.7 Установка проекта Triafly 20](#_Toc123055523)

[3.1.8 Запуск ПО Triafly 21](#_Toc123055524)

[3.1.9 Инициализация БД для Triafly 22](#_Toc123055525)

[3.1.10 Старт ПО Triafly 22](#_Toc123055526)

[3.1.11 Логгирование попыток входа в систему 22](#_Toc123055527)

[3.1.12 Описание остальных команд 22](#_Toc123055528)

[3.1.13 Изменение настроек приложения 24](#_Toc123055529)

[3.1.14 Запуск по https 27](#_Toc123055530)

[3.1.15 Дополнительные настройки 27](#_Toc123055531)

[3.1.16 Ротирование логов 30](#_Toc123055532)

[3.1.17 Быстрый старт/остановка системы 31](#_Toc123055533)

[4 Описание операций 33](#_Toc123055534)

[4.1 Описание операций настройки системы для решения прикладной задачи ……………….33](#_Toc123055535)

[4.1.1 Справочники и работа с ними 33](#_Toc123055536)

[4.1.2 Загрузка данных в систему 40](#_Toc123055537)

[4.1.3 Настройка процесса сбора данных 88](#_Toc123055538)

[4.2 Описание операций администрирования …………………………………………………..110](#_Toc123055539)

[4.2.1 Настройка главного меню 110](#_Toc123055540)

[4.2.2 Работа с метаданными, данными, базой данных 115](#_Toc123055541)

[4.2.3 Импорт и экспорт метаданных 120](#_Toc123055542)

[4.2.4 Настройка стилей интерфейса системы 121](#_Toc123055543)

[4.2.5 Точки восстановления 124](#_Toc123055544)

[5 Перенос настроек и тестовых данных 130](#_Toc123055545)

[5.1 Описание переноса настроек и тестовых данных …………………….……………………130](#_Toc123055546)

[6 Аварийные ситуации 132](#_Toc123055547)

[6.1 Действия в случае сбоев в работе программного обеспечения …………………..……….132](#_Toc123055548)

Обозначения и сокращения

Используемые в настоящем документе термины и основные понятия области автоматизированных систем определены в ГОСТ Р 59853–2021 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения». Также в тексте настоящего документа введены сокращения (Таблица 1).

Таблица – Перечень сокращений и обозначений

| Сокращение | Расшифровка |
| --- | --- |
| АРМ | Автоматизированное рабочее место |
| БД | База данных |
| Система | Автоматизированная информационная система |
| НСИ | Нормативно-справочная информация |
| ОС | Операционная система |
| СУБД | Система управления базами данных |
| ЭВМ | Электронная вычислительная машина |
| API | От англ. *Application Programming Interface* – программный интерфейс приложения, интерфейс прикладного программирования – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах |
| HTML | От англ. *HyperText Markup Language* – язык гипертекстовой разметки, который используется для создания документов в Интернет (веб-страниц) |
| HTTP | От англ. *HyperText Transfer Protocol –* протокол передачи гипертекста – протокол прикладного уровня передачи данных гипертекстовых документов |
| JSON | От англ. *JavaScript Object Notation* – текстовый формат обмена данными, основанный на JavaScript |
| ODBC | От англ. *Open Database Connectivity* – программный интерфейс доступа к базам данных |
| REST | От англ. *Representational State Transfer* – передача состояния представления – архитектурный стиль взаимодействия компонентов распределенного приложения в сети |
| SOAP | От англ. *Simple Object Access Protocol* – протокол обмена структурированными сообщениями в распределенной вычислительной среде |
| SQL | От англ. *Structured Query Language* – язык структурированных запросов – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных |
| URL | От англ. *Uniform Resource Locator* – единый указатель ресурса – единообразный определитель местонахождения ресурса |
| XML | От англ. *eXtensible Markup Language* – расширяемый язык разметки – язык, предназначенный для хранения и передачи данных |

# Введение

## Область применения

Система предназначена для автоматизации процессов обеспечения функционирования управления рисками и внутреннего контроля и антимонопольного комплаенса в Группе «Интер РАО», таких как составление отдельных слайдов бизнес-плана и отчета о его исполнении, управление типовыми рисками, планирование и контроль исполнения мероприятий по управлению рисками сбор и подготовка карт антимонопольных рисков, планов мероприятий, отчетов о функционировании антимонопольного комплаенса, ведение реестра общений в динамике (от поступления обращения и до вступления в силу завершающего процессуального документа).

## Требования к квалификации администратора

Основными квалификационными требованиями к обслуживающему персоналу Системы являются:

* Законченное высшее образование (техническое, со знанием экономики; либо экономическое, с техническим уклоном);
* Практические навыки работы или администрирования программной платформы «Триафлай»;
* Навыки работы с реляционными базами данных (PostgreSQL);
* Понимание структуры и этапов ИТ-проектов, умение выдерживать сроки;
* Знание основ языков программирования;
* Знание Linux;
* Знание Python – практические навыки программирования, знание технологий django, pandas, uwsgi, celery, daphne, rabbitmq, redis;
* Хорошее знание MS Office (в особенности MS Excel);
* Практические навыки работы или администрирования современных BI-систем
* Хорошие коммуникативные навыки, умение работать в команде.
* необходимый стаж самостоятельной работы;
* самостоятельная работа с современным серверным оборудованием, сетевым оборудованием, периферийным оборудованием, ленточными библиотеками, дисковыми массивами, сканерами, коммутационным оборудованием.

Уровень квалификации обслуживающего персонала должен соответствовать требованиям разработчиков (производителей) программного обеспечения и технических средств, используемых для обеспечения функционирования Системы.

К администраторам Системы предъявляются следующие требования:

* владение навыками администрирования локальной вычислительной сети, образующего ее активного сетевого оборудования, серверов под управлением используемой операционной системы, СУБД, то есть способность самостоятельно выполнять:
  + управление учетными записями пользователей операционной системы и СУБД;
  + установку и настройку параметров функционирования серверов и СУБД;
  + диагностические процедуры по определению целостности информации СУБД и информационных баз;
  + резервное копирование и восстановление данных;
* знание состава и назначения средств, входящих в Систему;
* знание способов работы со средствами мониторинга Системы, в том числе способность самостоятельно осуществлять:
  + управление учетными записями пользователей, группами пользователей;
  + управление правами доступа к данным Системы и функциональными полномочиями пользователей;
  + изменение настроек компонентов Системы;
  + установку программного обеспечения компонентов Системы.

## Перечень эксплуатационных документов

Перед началом работы администратору рекомендуется ознакомиться с настоящим документом.

# Назначение и условия применения

## Обозначение и наименование

Программа для ЭВМ «Система управления рисками и внутреннего контроля и антимонопольного комплаенса», состоящая из модуля «СУРиВК» и модуля «АМК.

Условное обозначение: Система.

## Назначение системы

Система предназначена для автоматизации процессов обеспечения функционирования управления рисками и внутреннего контроля и антимонопольного комплаенса в Группе «Интер РАО», таких как составление отдельных слайдов бизнес-плана и отчета о его исполнении, управление типовыми рисками, планирование и контроль исполнения мероприятий по управлению рисками сбор и подготовка карт антимонопольных рисков, планов мероприятий, отчетов о функционировании антимонопольного комплаенса, ведение реестра общений в динамике (от поступления обращения и до вступления в силу завершающего процессуального документа).

## Условия выполнения

* Система представляет собой веб-приложение, развернутое на сервере под управлением AstraLinux Orel 2.12.44, пакеты debian 9, использующее СУБД Postgres Pro Standart 13 и веб-сервер Apache.

Заказчик отвечает за своевременное обслуживание ИТ-инфраструктуры, включая Postgres Pro Standart 13 и Apache с устранением критических уязвимостей.

Для хранения данных Система использует реляционную СУБД. Хранение файлов осуществляется средствами файловой системы.

Система поддерживает работу СУБД Postgres Pro Standart 13.

Система обеспечивает стабильную работу на следующих платформах виртуализации:

* Mware vSphere (версия 6.5 и выше);
* KVM (версия 2.12 и выше).

Система поддерживает работу под управлением операционных систем:

* Red Hat Enterprise Linux версии 7.7 и выше;
* Ubuntu 20.04;
* AstraLinux 1.7;
* Altlinux 9;
* RedOS 7.3;

Система полностью web-ориентированная и поддерживает доступ ко всем рабочим интерфейсам пользователя через все наиболее распространенные Интернет-браузеры, включая:

* Edge последних версий;
* Google Chrome последних версий;
* Mozilla Firefox последних версий;
* Яндекс.Браузер последних версий.

Система имеет встроенные средства интеграции с LDAP-каталогами.

Система поддерживает следующие способы аутентификации пользователей:

* локальная аутентификация с хранением учетных данных во внутренней базе системы.
* аутентификация через Microsoft Active Directory с использованием персонального сертификата пользователя в формате X.509, Kerberos или NTLM не ниже версии 2.

Система обеспечивает стабильную работу на следующих пользовательских платформах:

* Microsoft Windows 10 и выше;
* Debian 10/11;
* Ubuntu 20,04/22,04
* Astra Linux CE 2.12
* macOS Monterey

Система корректно работает в ситуациях, когда несколько пользователей меняют одно и те же данные.

### Минимальные требования к серверному оборудованию (высоконагруженный режим работы)

Для работы в высоконагруженном режиме Система может быть развернута на трех серверах приложений со следующими характеристиками:

* процессор с тактовой частотой не менее 3.5GHz, 8 ядер;
* ОЗУ 64 Гбайт DDR4;
* не менее двух SSD NVMe дисков размером 500 Гбайт, объединенных в RAID1-массив;
* сервер СУБД – 1 шт. со следующими характеристиками:
* процессор с тактовой частотой не менее 3.5GHz, 8 ядер;
* ОЗУ 64 Гбайт DDR4;
* не менее двух SSD NVMe дисков размером 1 Тбайт, объединенных в RAID1-массив;
* балансировщик нагрузки – 1 шт. со следующими характеристиками:
* процессор с тактовой частотой не менее 3.5GHz, 2 ядра;
* ОЗУ 16 Гбайт;
* жёсткий диск 200 Гбайт.

### Требования к оборудованию рабочих станций

Система реализована как приложение с веб-интерфейсом. В качестве клиентов используются рабочие места пользователей. Работа с информацией осуществляется посредством программной оболочки с веб-интерфейсом с использованием веб-браузера. Технические средства (ТС), обеспечивающие функционирование АРМ Пользователя, должны иметь характеристики не ниже указанных:

* процессор 2 ядра;
* ОЗУ 2 Гбайт;
* жёсткий диск 10 Гбайт;
* сетевой порт (LAN): не менее 1 Мбит/сек.

# Подготовка к работе

## Установка ПО Triafly на ОС Astra Linux Common Edition

Установка системы осуществляется не от пользователя с правами администратора, а от отдельного пользователя (например netdbuser). Установка системы осуществляется от пользователя, не являющемся администратором/root.

Для работы c БД PostgreSQL необходимо добавить пользователя:

*# sudo -u postgres -i createuser -s -w имя\_пользователя*

Для запуска Triafly после установки пользователю будут необходимы права суперпользователя:

*# visudo*

Добавить строчку:

*имя\_пользователя ALL=(ALL) NOPASSWD: ALL*

Доступ к БД осуществляется от пользователя netdbuser с расширенными правами с возможностью создания баз данных.

Установка Triafly должна производиться от пользователя не-«root».

Попытки входа в систему логгируются в файл auth.log.

Исходный код проприетарных компонентов или стороннего свободно распространяемого программного обеспечения не содержит аутентификационной информации, а именно учетных записей, паролей, токенов аутентификации. Сторонние компоненты загружаются из репозиториев, где проходят проверку на отсутствие аутентификационной информации.

В Системе используются компоненты или подключаемые библиотеки стороннего свободно распространяемого программного обеспечения на базе открытого исходного кода (open-source). Их перечень и источники загрузки приведены в п. 3.1.3.

### Обозначение версии дистрибутива ИС

Обозначение версии дистрибутива имеет вид:

\* triafly\_{version}\_installer\_{platform}.sh

, где «version» - версия платформы Triafly;

«platform» - операционная система, под которую собран дистрибутив.

### Подготовка Системы

В данном разделе в качестве основного используемого дистрибутива операционной системы Linux рассматривается Astra Linux Common Edition.

В качестве пакетного менеджера используется утилиты \*apt-get\* и \*dpkg\*.

Примеры выполнения команд в инструкции представляют собой работу с командной строкой (используемой командной оболочкой является BASH).

Во всех примерах символы "#" и "$" в начале строки являются приглашением командной строки для пользователя "root" и для обычного пользователя соответственно.

Предполагается, что:

* установка платформы выполняется квалифицированным администратором ОС.
* операционная система установлена и функционирует;
* установка производится из предоставляемого дистрибутива ИС.

### Установка необходимых приложений и библиотек

При установке Системы требуется установка пакетов из стандартного перечня пакетов ОС:

* apache2;
* postgresql;
* unixodbc;
* odbc-postgresql;
* python3-virtualenv;
* postgresql-12-postgis-3;
* rabbitmq-server;
* redis;
* libreoffice;
* supervisor;
* unzip;
* gcc;
* libssl-dev;
* zlib1g-dev;
* libapache2-mod-auth-kerb.

Источник загрузки для этих пакетов настраивается в ОС силами системных администраторов заказчика.

Также в дистрибутиве поставляются сторонние библиотеки Python:

* amqp;
* backports.functools-lru-cache;
* base32-crockford;
* billiard;
* bottleneck;
* cairocffi;
* cairosvg;
* celery;
* certifi;
* cffi;
* chardet;
* configparser;
* contextlib2;
* croniter;
* cssselect;
* cycler;
* cython;
* dbf;
* decorator;
* django;
* django-celery-beat;
* django-honeypot;
* django-timezone-field;
* ecdsa;
* et-xmlfile;
* fabric;
* fbprophet;
* filelock;
* honcho;
* html5lib;
* idna;
* importlib-metadata;
* iso8601;
* jdcal;
* jsonpath-rw;
* kiwisolver;
* kombu;
* lockfile;
* lxml;
* m3-cdecimal;
* matplotlib;
* meta;
* mock;
* more-itertools;
* numexpr;
* numpy;
* openpyxl;
* pandas;
* pathlib2;
* patsy;
* pkg-resources;
* ply;
* profilehooks;
* psycopg2;
* pyasn1;
* pybars3;
* pycparser;
* pycrypto;
* pymeta3;
* pyparsing;
* pyphen;
* pystan;
* python-crontab;
* python-dateutil;
* python-docx;
* python-jose;
* python-keycloak;
* pytz;
* requests;
* requests-unixsocket;
* rsa;
* scandir;
* scipy;
* sentry-sdk;
* six;
* ssh;
* statprof;
* statsmodels;
* subprocess32;
* suds-jurko;
* tinycss;
* urllib3;
* uwsgi;
* vine;
* weasyprint;
* webcolors;
* webencodings;
* xlrd;
* xlutils;
* xlwt;
* zipp.

Данные библиотеки загружаются с pypi.org при сборке дистрибутива.

### Установка кодировки

Добавить кодировку \*en\_US\*:

# localedef en\_US.UTF-8 -i en\_US -fUTF-8

note:: Команда проверки существования кодировки::

# locale -a

### Настройка Apache2

Подключить модули Apache2:

# a2enmod rewrite

# a2enmod proxy

# a2enmod proxy\_http

# a2enmod proxy\_http2

# a2enmod proxy\_wstunnel

# a2enmod ssl

Произвести рестарт сервиса Apache2:

# systemctl restart apache2

### Создание пользователя

Установка ПО Triafly должна производиться от пользователя не-"root".

Создание пользователя:

# adduser имя\_пользователя

# passwd имя\_пользователя

Для работы c БД PostgreSQL необходимо добавить пользователя:

# sudo -u postgres -i createuser -s -w имя\_пользователя

Для запуска Triafly после установки пользователю будут необходимы права суперпользователя:

# visudo

Добавить строчку:

имя\_пользователя ALL=(ALL) NOPASSWD: ALL

После создания пользователя для установки Triafly необходимо переключиться на этого пользователя и перейти в его домашнюю папку:

# su имя\_пользователя

$ cd

### Установка проекта Triafly

Все команды выполняются без sudo (тем командам, которым нужны права суперпользователя, использование sudo прописано внутри самих команд).

Запустить инсталлятор:

$ bash triafly\_{version}\_installer\_{platform}.sh

Дополнительными аргументами командной строки инсталлятору можно передать дополнительные параметры в виде параметр1=значение1,параметр2=значение2 (набор "параметр=значение", разделенные запятыми). Например, если

$ bash triafly\_{version}\_installer\_{platform}.sh dbhost=IP\_АДРЕС\_БД,dbport=ПОРТ

Доступные параметры:

\*dbname\* - имя БД приложения, по умолчанию \*nedbapp\_db\*.

\*dbuser\* - имя пользователя для доступа к БД, по умолчанию - \*текущий пользователь\*, от которого устанавливается приложение.

\*dbpassword\* - пароль для доступа к БД, по умолчанию - \*пустая строка\*.

\*dbhost\* - имя или адрес хоста БД, по умолчанию - \*пустая строка\*.

\*dbport\* - порт сервера БД, по умолчанию - \*пустая строка\*, и в этом случае работа с БД производится через локальный сокет.

\*server\_name\* - имя виртуального хоста проекта для http-сервера Apache2 , по умолчанию \*netdb.local\*.

\*guniport\* - номер порта для работы приложения, по умолчанию - \*10003\*.

\*admin\_email\* - почтовый адрес администратора, по умолчанию - \*admin@lists.chtd.ru\*.

\*processes\* - количества рабочих процессов приложения, по умолчанию - \*2\*.

\*mapdbname\* - имя БД с картографической информацией, по умолчанию - \*map\_db\*.

\*timeout\* - таймаут работы приложения, по умолчанию - \*180 секунд\*.

### Запуск ПО Triafly

После установки приложения пользователю, от которого производилась установка, доступны команды:

\* netdb\_init\_db

\* netdb\_init

\* netdb\_restart

\* netdb\_update\_config

\* netdb\_command

\* netdb\_load\_dump

Если сразу после установки при попытке выполнения команды возникает ошибка "команда не найдена", то необходимо "обновить" параметры текущей сессии пользователя командой:

$ source ~/.bashrc

или начать новую сессию:

$ bash

### Инициализация БД для Triafly

Выполнить команду:

$ netdb\_init\_db

### Старт ПО Triafly

Выполнить команду:

$ netdb\_start

После этого приложение будет доступно по указанному при установке server\_name или по ip-адресу сервера, если server\_name не был указан.

Приложение готово к работе.

### Логгирование попыток входа в систему

Попытки входа в систему логгируются в файл auth.log

### Описание остальных команд

Ниже приведено описание остальных команд.

$ netdb\_update\_config

- :ref:`изменение настроек <netdb\_update\_config>` приложения. Настройки задаются аргументом команды в виде параметр1=значение1,параметр2=значение2 (набор параметр=значение, разделенные запятыми). Допустимые параметры:

$ netdb\_restart

- рестарт приложения. Необходимое действие после изменения настроек командой netdb\_update\_config.

$ netdb\_command

- вызов служебных :ref:`команд <commands>` netdb.

$ netdb\_load\_dump

- загрузка данных в БД из дампа.

Настройка почты для рассылки уведомлений

Своевременная отсылка почтовых уведомлений пользователям представляет собой одну из важных составляющих правильной работы системы.

По умолчанию отправка почтовых уведомлений в системе настроена через почтовый сервер, установленный на одном стенде с работающим приложением:

\* Имя хоста почтового сервера - localhost.

\* Номер порта для отправки - 25 (порт по умолчанию для работы по протоколу SMTP).

В случае использования почтового сервера, настройки которого отличаются от настроек по умолчанию, необходимо через команду:

\*netdb\_update\_config\*

задать свои настройки для имени или IP-адреса почтового сервера, номера порта (для протокола SMTP - 25), имени и логина пользователя, если на почтовом указываемом сервере необходима авторизация для отправки почты, и в зависимости от того, включена ли TLS/SSL.

Пример использования команды для почтового сервера, расположенного на другом хосте (в качестве примера - example.com), с аутентификацией и шифрованием:

#

netdb\_update\_config email\_host=example.com,email\_port=25,email\_host\_user=user@example.com,email\_host\_password=user\_password,email\_use\_tls=yes

# netdb\_restart

only:: chtd\_dev

CHTD\_DEV

Для того, чтобы все почтовые сообщения для реальных адресов направлялись на один адрес (например, для отладки), следует в файле \_project/netdb\_demo/netdb\_demo/settings/email.py определить значение для параметр:

FAKE\_EMAIL = 'tester@example.com'

CHTD\_DEV\_END

\_netdb\_update\_config:

### Изменение настроек приложения

Для изменения параметров конфигурации приложения используется команда:

\*netdb\_update\_config\*

Список основных параметров конфигурации:

\* 'user' - имя пользователя, от которого работает приложение;

\* 'db' - имя БД приложения;

\* 'dbuser' - имя пользователя для доступа к БД, по умолчанию значение параметра 'user';

\* 'dbpassword' - пароль для доступа к БД, по умолчанию - пустая строка;

\* 'dbhost' - имя или адрес хоста БД, по умолчанию - пустая строка;

\* 'dbport' - порт сервера БД, по умолчанию - пустая строка, и в этом случае работа с БД производится через локальный сокет;

\* 'server\_name' - имя виртуального хоста проекта для http-сервера (Apache2 или Nginx),

по умолчанию 'netdb.local';

\* 'guniport' - номер порта для работы приложения;

\* 'admin\_email' - почтовый адрес администратора, по умолчанию - 'admin@lists.chtd.ru';

\* 'processes' - количество рабочих процессов приложения, по умолчанию - 2;

\* 'remote\_memdb\_processes' - количество ворекров memdb, по умолчанию - 1;

\* 'celery\_processes' - количество воркеров celery, по умолчанию - 2;

\* 'mapdbname' - имя БД с картографической информацией, по умолчанию - 'map\_db';

\* 'timeout' - таймаут работы приложения, по умолчанию - 180 секунд;

\* 'autostart' - признак того, что приложение запускается при старте supervisor,

по умолчанию true;

\* 'email\_host' - хост, где расположен почтовый сервер;

\* 'email\_port' - используемый почтовым сервером порт для SMTP (по умолчанию 25);

\* 'email\_host\_user' - имя пользователя, для аутентификации при отправке по SMTP;

\* 'email\_host\_password' - пароль пользователя;

\* 'default\_from\_email' - значение поля from для исходящих писем;

\* 'email\_use\_tls' - признак того, что используется TLS/SSl для шифрования;

\* 'modem\_url' - шабон url для отправки смс-сообщений. В {phonenumber} и {message} будут подставлены номер пользователя и сообщение, соответственно.

Пример: "http://sendsms.com?username\=sms&password\=sms&phonenumber\={phonenumber}&message\={message}"

\* 'certificate' - файл сертификата SSL;

\* 'certificate\_key' - файл ключа SSL;

\* 'login\_max\_attempts' - максимальное число попыток неуспешного входа, после которых пользователь блокируется, по умолчанию - 0 (отключено);

\* 'login\_lock\_time' - время блокировки, в секундах, по умолчанию - 0 (отключено);

\* 'user\_max\_inactive\_period' - время бездействия в секундах, после которого при попытке логина пользователь будет сделан "неактивным", т.е. не сможет пользоваться системой (чтобы активировать - необходимо поставить галочку "активен" в реестре пользователей), по умолчанию - 0 (отключено);

\* 'session\_cookie\_age' - настройка SESSION\_COOKIE\_AGE из Django, время сессии пользователя в секундах. По истечении этого времени залогиненного пользователя выкинет на страницу логина при выполнении какого-либо действия. По умолчанию - 2 недели;

\* 'user\_max\_sessions' - максимальное количество одновременных сессий одного пользователя, по умолчанию - 0 (отключено);

\* 'strict\_password\_checking' - настройка, включающая проверку сложности пароля, по умолчанию - False

\* 'admin\_password\_min\_len' - минимальная длина пароля для администратора, по умолчанию - 16 (только в случае 'strict\_password\_checking'=True);

\* 'user\_password\_min\_len' - минимальная длина пароля для обычного пользователя, по умолчанию - 8 (только в случае 'strict\_password\_checking'=True);

\* 'password\_expiration' - время, через которое система потребует сменить пароль в днях, по умолчанию - 90 (только в случае 'strict\_password\_checking'=True); Чтобы изменить значения параметров, необходимо выполнить команду со значениями параметров, например:

$ netdb\_update\_config timeout=10800,processes=4

и перезагрузить приложение, выполнив команду:

$ netdb\_restart

В случае изменения server\_name, ssl, celery\_processes , необходимо выполнить:

$ netdb\_start.

### Запуск по https

Необходимо задать параметры SSL:

$ netdb\_update\_config certificate= /path/to/certificate.crt,certificate\_key=/path/to/key.key

и запустить командой:

$ netdb\_start ssl.

### Дополнительные настройки

#### Настроить белый список Mime-types загружаемых файлов

Необходимо проверить настройки параметров в .fabfile:

check\_incoming\_files = yes

allowed\_incoming\_files\_types = xlsx:docx:xls:doc:ods:odt:pdf:png:svg

Система определяет тип загружаемого файла по содержимому, а не по расширению файла.

#### Максимальный размер загружаемого файла

Максимальный размер загружаемого файла настраивается в common.py:

DATA\_UPLOAD\_MAX\_MEMORY\_SIZE = 104857600 # 20 MB (default is 2.5 MB).

#### Настроить максимальное количество попыток входа в Систему

Выставить максимальное количество попыток входа в Систему до блокировки можно командой (см 3.1.13):

login\_max\_attempts = 5

#### Настроить использование заголовков HTTPOnly, Content-Security-Policy и X-XSS-Protection

По умолчанию в файле common.py стоит настройка:

SESSION\_COOKIE\_SECURE = SERVER\_PROTOCOL == 'https'

При необходимости следует добавить параметр в файл:

.fabfile:

server\_protocol = https

В файле common.py уже есть определение настроек:

SECURE\_CONTENT\_TYPE\_NOSNIFF = True

SECURE\_BROWSER\_XSS\_FILTER = True

#### Максимальное время неактивности пользователя

Максимальное время неактивности пользователя определяется параметром (см 3.1.13):

user\_max\_inactive\_period, которому необходимо присвоить значение 1800 (30 минут).

#### Проверка прав на запись у пользователя netdbuser за пределами /home/netdbuser.

Пользователь netdbuser создается на этапе установки системы стандартными средствами ОС, соответственно, не имеет прав доступа на запись за пределами домашней директории.

#### Настройка использования заголовков

Для того, чтобы настроить в апаче использование заголовков HTTPOnly, Content-Security-Policy и X-XSS-Protection необходимо проконтролировать установку следующих параметров:

В файле common.py должна быть настройка:

SESSION\_COOKIE\_SECURE = SERVER\_PROTOCOL == 'https'

В файле уже должна быть настройка:

fabfile:

server\_protocol = https

В файле common.py должно быть определение настроек:

SECURE\_CONTENT\_TYPE\_NOSNIFF = True

SECURE\_BROWSER\_XSS\_FILTER = True

#### Настройка TLS 1.2 или 1.3 с использованием EDH

Для настройки TLS 1.2 или 1.3 с использованием EDH с ключом 4096 бит необходимо проверить наличие в конфигурационном файл Апача `\_conf/httpd.gunicorn.ssl.conf` следующих директив:

SSLProtocol -all +TLSv1.3 +TLSv1.2

SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH

#### Настройка HSTS

Настройка HSTS выполняется следующим способом:

В конфигурационном файле Апача - \_conf/httpd.gunicorn.ssl.con прописано:

Header always set Strict-Transport-Security "max-age= 86400; includeSubdomains; preload"

#### Сохранение бэкапов

Бэкапы сохраняются с использованием ПО CRONTAB в папке:

/home/netdbuser/backup/dumps/

от пользователя netdbuser.

#### Создать пользователя "администратор ИБ", который может смотреть /home/netdbuser, править ничего (в т.ч. логи) не может

Создан пользователь `reader` с правами на чтение конфигурационных файлов и логов приложения, но без прав на их изменение

#### Ограничение на длительность пользовательской сессии

Ограничение на длительность пользовательской сессии задается настройкой SESSION\_COOKIE\_AGE из Django, которая определяет время сессии пользователя в секундах. По истечении этого времени пользователя, вошедшего под учетной записью, при выполнении какого-либо действия, перенаправит на страницу ввода логина и пароля. По умолчанию время пользовательской сессии задано 2 недели.

### Ротирование логов

Ротирование логов приложения можно настроить с помощью утилиты:

logrotate.

В основном конфигурационном файле /etc/logrotate.conf находится глобальная конфигурация, то есть директивы, которые находятся в этом конфигурационном файле, распространяться на все логи, их можно переопределить в файлах конфигураций сервисов, которые должны хранится в каталоге /etc/logrotate.d.

По умолчанию логи приложений netdb\_demo и netdb\_proj не ротируются.

Для приложений, развернутых из дистрибутива, необходимо в каталоге /etc/logrotate.d создать файл netdbapp (пример конфигурации для приложения):

/home/netdbuser/apps/netdbapp/\_project/netdb\_demo/\_log/\*.log

/home/netdbuser/apps/netdbapp\_memdb/\_project/netdb\_proj/\_log/\*.log

{

copytruncate

daily

rotate 5

compress

delaycompress

missingok

notifempty

}

где:

\* copytruncate - обрезать файл журнала вместо перемещения старого файла журнала и создания нового;

\* daily - ежедневное ротация файлов журнала (доступные опции - daily, weekly, monthly, можно по размеру файла);

\* rotate 5 - файлы журнала ротируются заданное количество раз (в примере 5 раз), предыдущие удаляются;

\* compress - сжимать старые файлы журналов с помощью gzip;

\* delaycompress - отложить сжатие предыдущего файла журнала до следующего цикла ротации;

\* missingok - если файл журнала отсутствует, перейти к следующему без создания сообщения об ошибке;

\* notifempty - не ротировать журнал если он пуст.

Полный список опций можно посмотреть - <https://linux.die.net/man/8/logrotate>.

### Быстрый старт/остановка системы

Все действия выполняются под учетной записью netdbuser  
Основной рабочий каталог /home/netdbuser/apps/netdbapp/\_project/netdb\_demo

Заходим любым доступным SSH клиентом на сервер

ssh netdbuser@%hostname% #где hostname - где имя сервера, на котором установлен программный комплекс

cd /home/netdbuser/apps/netdbapp/\_project/netdb\_demo #для перехода в рабочую папку, для дальнейшей работы с программным комплексом

source ../../bin/activate #данным действием мы активируем окружения и дальнейшие действия будут происходить для программный комплекса

<https://docs.triafly.ru/commands.html#commands> - описаны все доступные команды

$ netdb\_start # запуск системы  
$ netdb\_stop # остановка  
$ netdb\_restart # перезапуск

deactivate # выполнять после всех действий над системой, чтобы не оставались открытые соединения

# Описание операций

## Описание операций настройки системы для решения прикладной задачи

### Справочники и работа с ними

В Системе существует два вида справочников – простые и иерархические. Иерархический справочник – это справочник с иерархической структурой. Для каждого элемента такого справочника (кроме корневых) должен быть задан родительский элемент, при этом элементы не должны циклически ссылаться друг на друга. Для того чтобы справочник стал иерархическим, следует добавить в него столбец, элементы которого будут ссылаться на этот же справочник (на поле «Название»). Например, в справочнике административной структуры для района родительским элементом может быть область, а для области – округ (рисунок 1).

Создание и наполнение справочников может выполняться вручную (см. данный раздел) и автоматически при загрузке данных через коннекторы (см. п. 4.1.2.2.4)

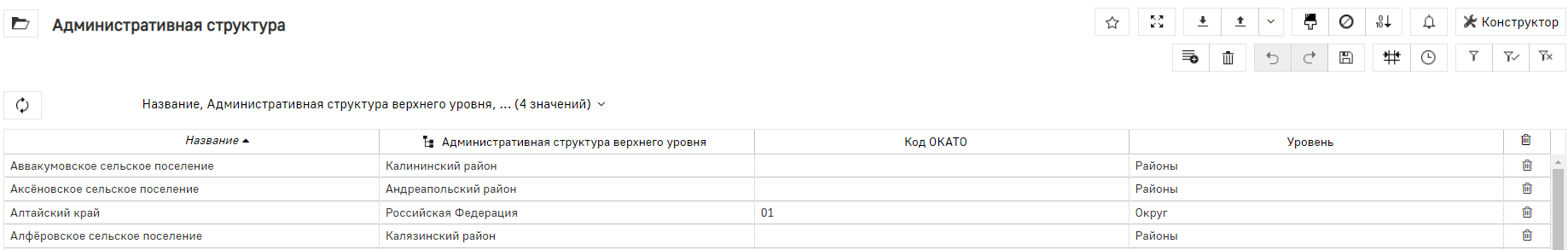


Рисунок – Фрагмент справочника «Административная структура»

#### Просмотр списка справочников

Для просмотра списка справочника, существующих в Системе, следует в главном меню выбрать пункт «Справочники». Откроется страница со списком справочников (рисунок 2).

В правой части страницы расположена кнопка «Показать/Скрыть фильтры». По нажатию кнопки становится доступна возможность фильтрации по полю «Иерархический». Для просмотра только простых справочников следует выбрать значение «Нет», для просмотра иерархических справочников – значение «Да». Фильтры будут применены автоматически.

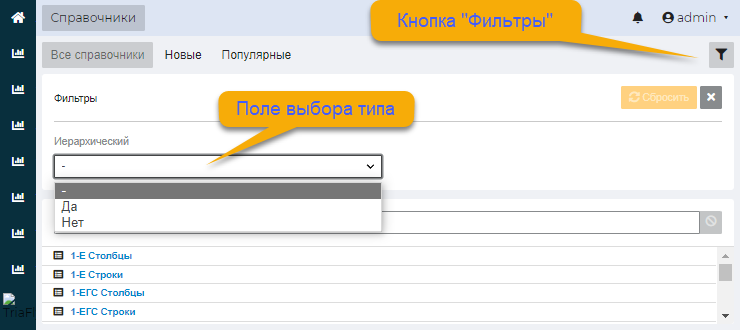


Рисунок – Список справочников, кнопка и панель «Фильтры»

#### Просмотр справочника

Для просмотра справочника следует в главном меню выбрать пункт «Справочники». Откроется страница со списком справочников (рисунок 2). Затем необходимо найти нужный справочник, при необходимости используя фильтр и щелкнуть левой кнопкой мыши по ссылке-названию справочника. Откроется страница просмотра выбранного справочника.

#### Системные справочники

В Системе существуют предварительно настроенные системные справочники, содержащие информацию о настройках Системы. Как правило, на системные справочники ссылаются соответствующие системные показатели. Системные справочники доступны для редактирования администраторам.

Примечание – Часть системных справочников в интерфейсе Системы не отображаются, так как они содержат информацию, необходимую разработчикам.

#### Создание справочника

Первыми в Системе создаются показатели с ограниченным (как правило, уже известным) множеством значений. Значения для таких показателей задаются через справочники, поэтому предварительно необходимо создать эти справочники.

##### Создание плоского справочника

Для того чтобы создать справочник, следует создать или открыть на редактирование паспорт показателя типа «Справочник». На странице редактирования паспорта показателя, в разделе «Из справочника», создать справочник:

* нажать пиктограмму 183;
* в появившемся поле ввести название нового справочника;
* нажать пиктограмму ;
* нажать пиктограмму 183 еще раз.

Справочник будет создан, даже если показатель не сохранен. Но использовать справочник, если нет соответствующего показателя, невозможно, поэтому рекомендуется сохранить соответствующий показатель.

Порядок редактирования и наполнения справочника описан в п. 4.1.1.5.

##### Создание иерархического справочника

Для того чтобы создать иерархический справочник, следует сначала создать плоский справочник, а затем отредактировать его: добавить в него столбец, элементы которого будут ссылаться на этот же справочник. Для этого необходимо открыть на редактирование показатель типа «Справочник», который необходимо сделать иерархическим. Затем следует установить флаг «Иерархический справочник, группировать по текущему показателю» (рисунок 3).



Рисунок – Флаг «Иерархический справочник, группировать по текущему показателю»

После сохранения показателя в справочник автоматически добавится поле с пиктограммой – это значит, что справочник сгруппирован по этому показателю. Затем следует открыть справочник и заполнить новое поле (показатель, по которым группируется справочник – выбрать родительские элементы) для всех элементов, кроме корневых.

#### Наполнение и редактирование справочника

Для того чтобы добавить новый элемент, нажать кнопку «Добавить строку» () или использовать сочетание клавиш «Ctrl+Enter», когда активна последняя строка справочника (либо выделена любая ячейка в последней строке) и затем ввести значение в поле «Название» (должно быть уникально).

Чтобы отредактировать значение, следует нажать на нужную ячейку таблицы и изменить значение элементы справочника, необходимо:

Если для элементов справочника должны быть определены дополнительные показатели (помимо стандартного показателя «Название»), нажать на кнопку со списком показателей над справочником и добавить показатель, отметив его флагом в списке (рисунок 4).

Если нужного показателя пока не существует в Системе, его необходимо создать, затем обновить страницу справочника по нажатию кнопки «Обновить» слева вверху и добавить в структуру справочника.

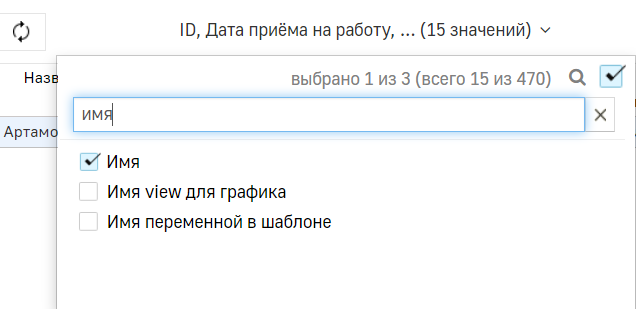


Рисунок – Дополнительный показатель в справочнике

Просмотр и редактирование данных справочника также возможно через соответствующий реестр. Это необходимо в тех случаях, когда требуется использовать в работе (просматривать/редактировать) справочник, но не все его поля, а только некоторые.

#### Настройка порядка отображения элементов в справочнике

Если необходимо явно задать порядок отображения элементов справочника в Системе, следует добавить в него показатель «Порядковый номер» (рисунок 5).

Порядковый номер задается вещественным числом, и это позволяет добавлять новые записи в упорядоченный справочник, вставляя их между двумя пронумерованными элементами справочника, при этом не приходится изменять присвоенные ранее порядковые номера. По умолчанию значения в справочнике не сортируются по порядковым номерам.

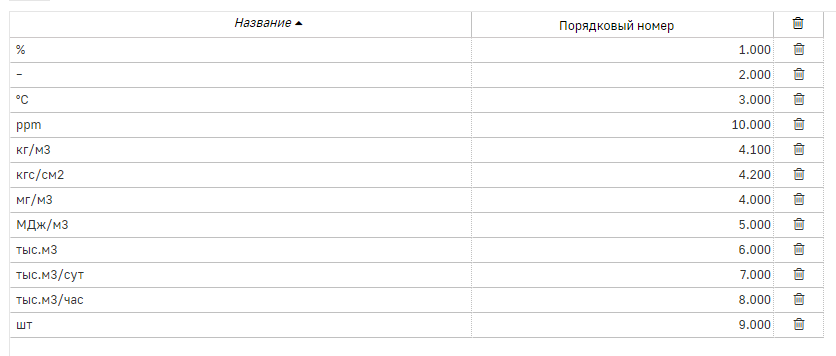


Рисунок – Настройка порядка отображения элементов в справочнике

#### Скрытие устаревших элементов справочника

Если какие-то элементы справочника устарели, их можно скрыть, добавив в справочник системный показатель «Элемент скрыт?» и установив для него значение «Да» для устаревших элементов справочника.

Скрытые элементы автоматически исключаются из списка элементов справочника, отображаемого в заголовках отчетов (в фильтрах при задании заголовков формулой, при автоматическом добавлении поддерева к элементу иерархического справочника), а также при выборе параметров отчета. Явно заданные заголовки (фильтр вида «Показатель = данный элемент») с такими элементами автоматически не скрываются.

Скрытые элементы можно будет просмотреть, если в настройках отчета включить настройку «Разрешить просмотр скрытых значений».

#### Переименование справочника

Для того чтобы переименовать справочник, необходимо найти и открыть показатель, который ссылается на нужный справочник (рисунок 6).

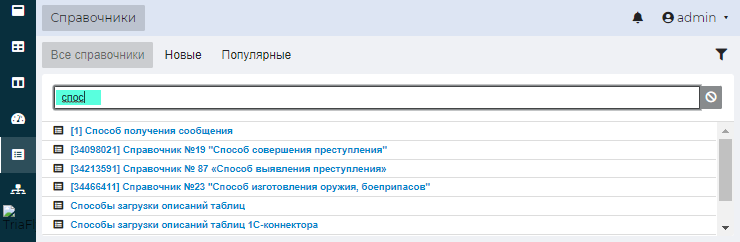


Рисунок – Поиск показателя, который ссылается на справочник

В открывшемся справа паспорте выбранного показателя следует нажать кнопку «Редактировать» (рисунок 7).

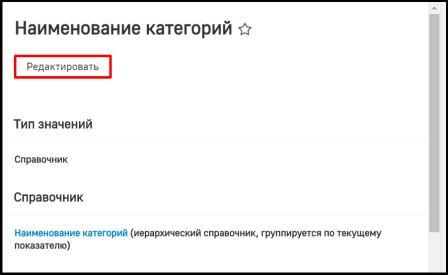


Рисунок – Переход к редактированию показателя

В разделе «Из справочника» надо нажать на кнопку редактирования  (рисунок 8).

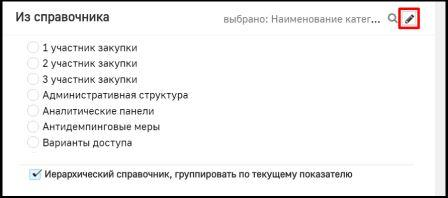


Рисунок – Переход к редактированию справочника

Затем следует найти справочник, название которого требуется изменить (рисунок 9).

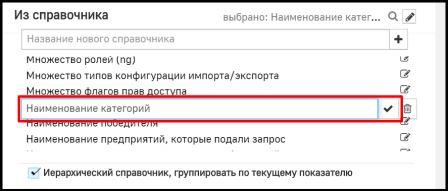


Рисунок – Поиск справочника, название которого требуется изменить

После этого нажать на название справочника и ввести новое название. Для завершения переименования нажать на пиктограмму завершения редактирования () рядом с измененным названием и затем снова на кнопку редактирования () для выхода из режима редактирования названия. В завершение необходимо сохранить показатель.

### Загрузка данных в систему

#### Создание коннекторов для загрузки данных

Компонент «Коннекторы» обеспечивает загрузку данных в справочники и реестры Системы из внешних информационных систем и внешних файлов различных форматов. Порядок загрузки данных приведен на рисунке 10.



Рисунок – Порядок загрузки данных

Система предоставляет базовый набор адаптеров для следующих типов источников данных:

* CSV – адаптер использует Python-библиотеку csv. Коннектор обрабатывает не только CSV-файлы (Comma-separated values – значения, разделенные запятыми), но и файлы с другими разделителями (DSV delimiter-separated values – значения, разделённые разделителем), например SCSV (Semi-colon-separated values – значения, разделенные точкой с запятой), TSV (Tab-separated values – значения, разделенные символом табуляции). В качестве разделителя можно также использовать двоеточие, пробел, вертикальную черту;
* OData.v3 – поддерживается протокол OData версии 3.0 (Open Data Protocol v 3.0). Протокол реализован частично, только для реализации возможности чтения данных с внешнего сервиса. Используется для подключения к данным 1С версий более 8.2;
* SOAP – обеспечивает загрузку данных по протоколу SOAP. При создании коннектора надо указать адрес сервера SOAP в формате http://login:password@address. После того, как коннектор будет создан, надо добавить JSON-запрос, определяющий параметры method, args и xpath;
* XLS – адаптер использует Python-библиотеку xlrd, загружает файлы формата xls, xlsx;
* XML – коннектор для загрузки данных из XML-файла. После того как коннектор будет создан, надо добавить произвольный запрос в формате XPath, определяющий путь к загружаемой таблице;
* БД по ODBC – адаптер к базам данных PostgreSQL, MySQL, Firebird, MSSQL, ORACLE работает через ODBC-драйвер;
* Текст – в отличие от формата CSV адаптер обрабатывает неструктурированный текст, без заданной структуры столбцов, не пытается разделить на столбцы «по умолчанию». При настройке коннектора нужно самостоятельно создавать вычисляемые поля на основе разбора строк данных;
* Битрикс24 – адаптер работает через Rest API https://dev.1c-bitrix.ru/rest\_help/;
* RetailCRM – адаптер работает через RestAPI v5 https://www.retailcrm.ru/docs/Developers/ApiVersion5;
* Google Adwords – адаптер использует API отчетов https://developers.google.com/adwords/api/docs/appendix/reports/all-reports;
* Google Spreadsheets – адаптер использует API к таблицам Google https://developers.google.com/sheets/api/.

Коннекторы к источнику данных настраиваются средствами пользовательского интерфейса. Коннекторов в Системе может быть создано неограниченное количество. Сохраненные коннекторы можно менять, перенастраивать и удалять. Каждый из коннекторов имеет пользовательское название, которое задается при его создании.

##### Страница работы с коннектором

Перейти к работе с коннекторами можно, выбрав в главном меню Системы пункт «Загрузка данных». Откроется страница, содержащая две вкладки:

* «Загрузка данных» – список коннекторов, существующих в Системе. Вкладка, в свою очередь, содержит две вкладки:
* «Все коннекторы» – список всех коннекторов Системы;
* «Новые» – последние созданные коннекторы;
* «Журнал загрузок» – список записей о загрузках данных в Систему.

На вкладке «Загрузка данных» по нажатию на название коннектора откроется страница работы с ним.

Страница работы с коннектором (рисунок 11)содержит:

* верхнее меню с кнопками, позволяющими:
* загрузить данные;
* создать новый коннектор;
* открыть коннектор;
* сохранить коннектор;
* сохранить коннектор как новый;
* открыть настройки коннектора;
* открыть настройки периодической загрузки данных;
* перейти к просмотру загруженных таблиц;
* удалить коннектор (доступна только при работе с уже сохраненными в Системе коннекторами);
* две области:
* левая – для работы с загружаемыми таблицами;
* правая, занимающая основную часть экрана, – для работы с загружаемыми данными.

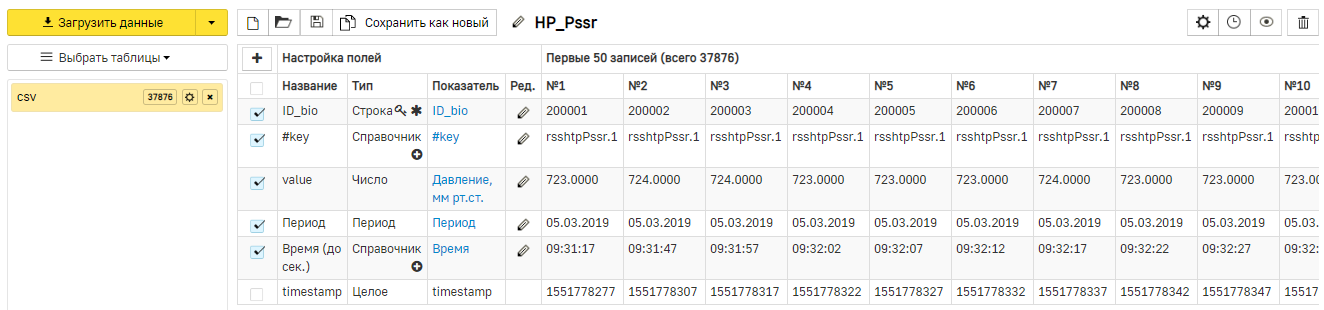


Рисунок – Страница работы с коннектором

##### Создание нового коннектора

Для создания нового коннектора необходимо:

1. в главном меню нажать на кнопку добавления () около пункта «Загрузка данных» или нажать кнопку создания нового коннектора () на странице работы с коннектором;
2. в открывшемся окне создания нового коннектора (рисунок 12) ввести название нового коннектора и выбрать его тип (MS SQL Server, MySQL, OData.v3, Oracle, PostgreSQL, XLS, коннектор к БД по ODBC и др.);
3. выполнить настройку коннектора в зависимости от его типа (см. пп. 4.1.2.1.2.1 – 4.1.2.1.2.5);
4. нажать кнопку «ОК».

После окончания обработки данных из источника окно создания коннектора закроется и станет активной кнопка «Выбрать таблицы» в поле работы с таблицами (подробнее см. п. 4.1.2.1.3).

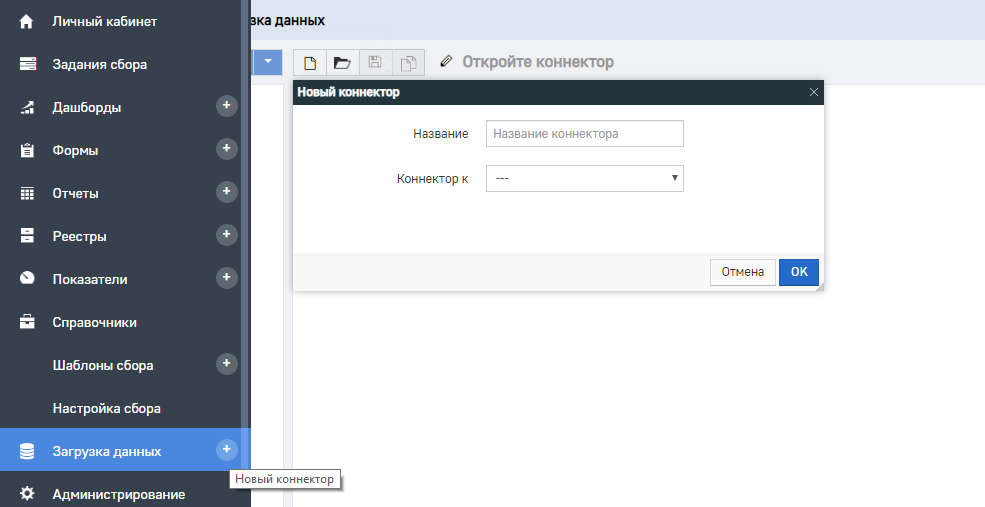


Рисунок 12 – Создание нового коннектора

###### XLS-коннектор

Если выбран XLS-коннектор, в форме создания коннектора необходимо нажать на кнопку «Выберите файл» и выбрать нужный файл в формате xls или xlsx, который будет закреплен за коннектором.

Чтобы файл правильно распознался в коннекторе, необходимо соблюдение следующий условий:

* в первой строке должны находиться названия столбцов;
* начиная со второй строки должны находиться данные;
* в файле не должно быть объединенных ячеек или формул;
* во избежание ошибок в файле должен использоваться тип данных «текст»;
* каждая строка в файле породит отдельный объект в Системе, поэтому в строке должна быть представлена единица информации (примеры неправильного и правильного форматов файлов приведены на рисунках 13 и 14).

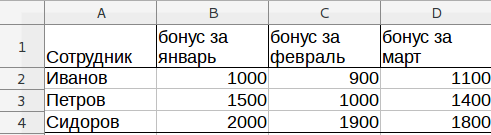


Рисунок 13 – Пример неправильного файла формата Microsoft Excel

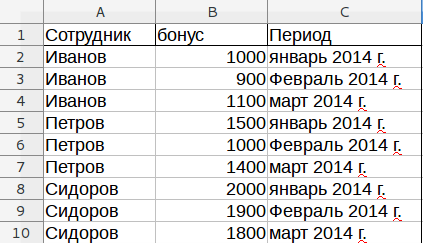


Рисунок 14 – Пример правильного файла формата Microsoft Excel

Когда появится надпись «Файл загружен» под кнопкой выбора файла, следует нажать кнопку «ОК» (рисунок 15). Начнется первичная обработка данных Системой.

Разные листы одного xls-файла будут рассмотрены Системой как разные таблицы с данными. При этом названия таблиц будут совпадать с названиями листов в файле.

Следует обратить внимание, что при загрузке данных первая строка таблицы рассматривается как заголовки столбцов (названия будущих показателей в Системе). Поэтому столбцы, в которых первая строка пуста, не будут загружены.

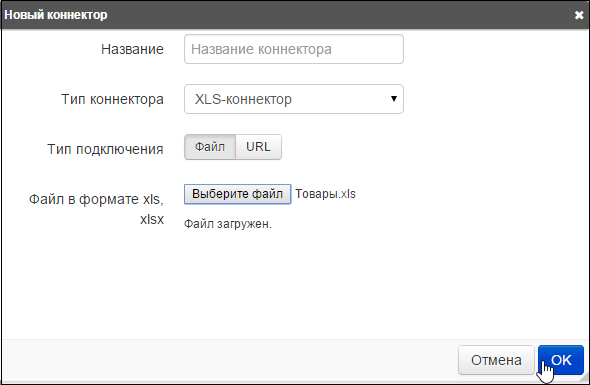


Рисунок – Настройка XLS-коннектора для файла Microsoft Excel

В случае если файл Microsoft Excel размещен в сети Интернет, можно указать его адрес. Для этого следует изменить тип подключения с «Файл» на «URL» и ввести адрес файла в формате xls в сети Интернет (рисунок 16).

При установленном флаге «Дополнительные заголовки» появляется возможность задать дополнительные заголовки HTTP-запроса к источнику данных.

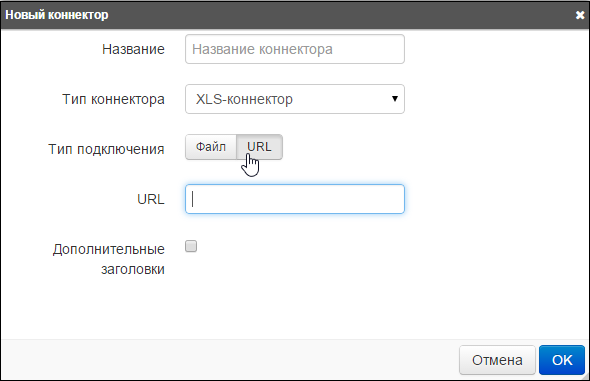


Рисунок – Настройка XLS-коннектора для файла, размещенного в сети Интернет

###### XML-коннектор

Коннектор позволяет загружать данные из XML-файла. Кроме того, XML-коннектор позволяет использовать XML API Системы.

Перед началом работы необходимо загрузить XML-файл или выбрать тип подключения URL и ввести адрес файла в сети Интернет (аналогично настройкам для XLS-коннектора, см. п. 4.1.2.1.2.1).

При установленном флаге «Дополнительные заголовки» появляется возможность задать дополнительные заголовки HTTP-запроса к источнику данных.

После того, как источник открыт, необходимо нажать на пиктограмму «плюс» возле кнопки «Выбрать таблицу», далее указать наименование таблицы, а в поле «Запрос к источнику данных» надо добавить запрос в формате XPath. Этот запрос определяет путь к таблице. Спецификация языка XML Path (XPath) версии 1.0 описана по ссылке http://citforum.ru/internet/xpath/index.shtml.

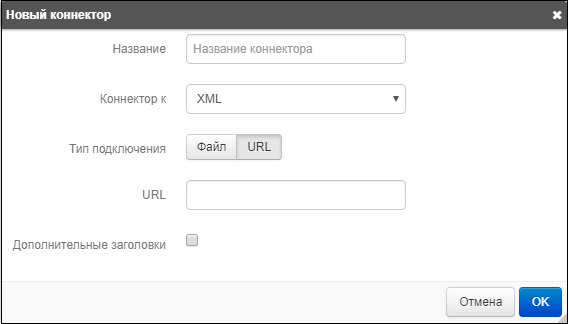


Рисунок – Настройка XML-коннектора

###### ODBC-коннектор

Для того чтобы при помощи коннектора был возможен доступ к БД, необходимо убедиться в том, что база данных доступна с сервера, где размещена Система. В БД должна быть создана учетная запись пользователя, который имеет права на чтение необходимых таблиц. При настройке подключения потребуется ввести логин и пароль пользователя для доступа к БД. Можно использовать те же параметры доступа, что уже настроены в БД, но рекомендуется добавить отдельного пользователя для доступа из Системы и назначить ему минимально необходимые права.

Для настройки подключения к базе данных по ODBC в форме создания коннектора дополнительно необходимо заполнить поля:

* «Подключение» – способ подключения к БД;
* «Способ предпросмотра» – в поле выбирается способ предпросмотра таблиц в БД (рисунок 18). Выбранное значение определяет объем предварительно загружаемых данных и, как следствие, время открытия коннектора и загрузки данных. Возможны следующие значения:
* «Все таблицы и данные» – загружается список всех таблиц в БД, после этого асинхронно подгружается содержимое каждой таблицы (позволяет видеть количество элементов в таблицах);
* «Все таблицы и данные (доступные пользователю)» – загружается список всех доступных пользователю таблиц БД (проверка на доступность может занимать длительное время), после этого асинхронно подгружается содержимое каждой таблицы (позволяет видеть количество элементов в таблицах);
* «Только названия таблиц» – загружаются названия всех таблиц БД, содержимое загружается только для выбранных таблиц;
* «Только названия таблиц (доступные пользователю)» – загружаются названия всех доступных пользователю таблиц БД (проверка на доступность может занимать длительное время), содержимое загружается только для выбранных таблиц;
* «Только настроенные таблицы» – при создании нового коннектора таблиц не будет, для загрузки необходимо писать SQL-запрос. Если коннектор уже настроен (например, при загрузке модуля), то будут загружены уже настроенные таблицы и ничего кроме них. Самый быстрый вариант загрузки.

Для БД с большим числом записей рекомендуется выбирать способ «Только названия таблиц»;

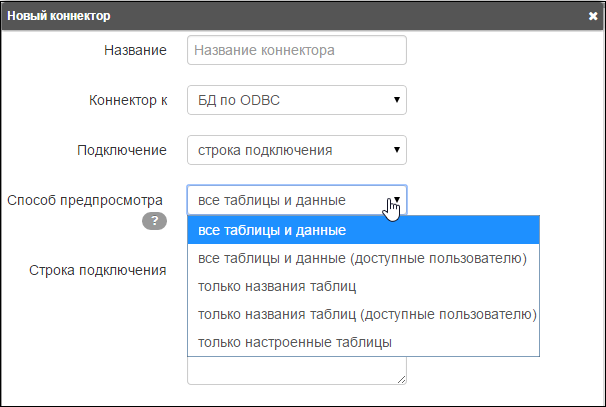


Рисунок – Выбор способа предпросмотра таблиц в БД

* если в поле «Подключение» выбрано значение «строка подключения», то в поле «Строка подключения» необходимо ввести строку подключения. Строка подключения имеет вид:

DRIVER=PostgreSQL Unicode;SERVER=servername;DATABASE=dbname;UID=username;PWD=password;SCHEMA=

Параметры:

* Servername – имя сервера;
* Dbname – имя БД;
* Username – пользователь БД;
* Password – пароль к БД;
* Schema – схема БД, по умолчанию – public.
* если в поле «Подключение» выбрано значение «указанные параметры», то потребуется выбрать подходящий драйвер в поле «Драйвер», а также ввести параметры:
* Имя БД;
* Сервер БД;
* Пользователь БД;
* Пароль к БД;
* Схема БД (по умолчанию – public);

Другие способы подключения (dmarts, ODBC Data Sources и др.) зависят от настроек в файле odbc.ini для пакета unixodbc.

###### Oracle, MS SQL Server, Firebird, PostgreSQL

Коннекторы к СУБД Oracle, MS SQL Server, Firebird или PostgreSQL работают через ODBC-драйвер. Для создания коннектора в поле «Коннектор к» необходимо указать соответствующее название СУБД. Остальные настройки выполняются аналогично описанным в п. 4.1.2.1.2.3.

###### OData-коннектор

Коннектор предназначен для работы с сервисами по протоколу OData версии 3.0.

Для нового коннектора типа «OData.v3» в поле URL необходимо указать адрес сервиса для подключения (рисунок 19).

Проверить подключение можно с помощью тестового адреса http://services.odata.org/V3/OData/OData.svc

Для доступа к сервису, использующему Basic Auth для аутентификации, имя пользователя (username) и пароль (userpass) необходимо включить в строку адреса в формате http://username:userpass@odata.service/odata.svc.

В коннекторе доступны вычисления на языке Python, но не все конструкции языка поддерживаются.

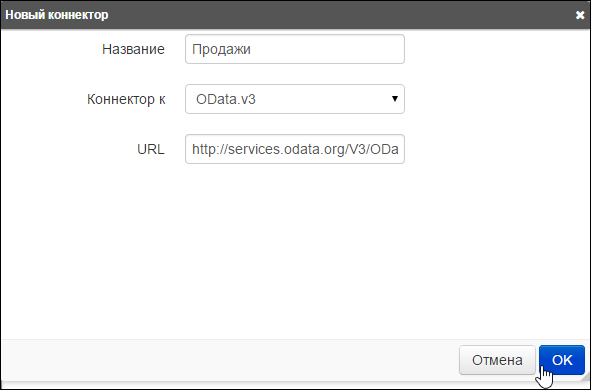


Рисунок – Настройка OData-коннектора

##### Настройка таблиц для загрузки данных

После окончания обработки данных из источника окно создания коннектора закроется и станет активной кнопка «Выбрать таблицы» в поле работы с таблицами (рисунок 20). По нажатию на нее откроется список, в котором можно выбрать, какие таблицы из источника данных необходимо загрузить в Систему. Их необходимо отметить флагами. Справа от названия каждой из таблиц указано, сколько строк записей она содержит.

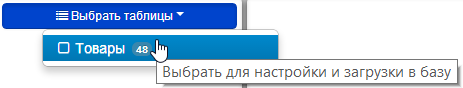


Рисунок – Выбор данных для загрузки

Все выбранные для загрузки таблицы отобразятся в списке под этой кнопкой. Чтобы удалить какую-либо таблицу из загрузки, необходимо нажать на пиктограмму «крестик» рядом с названием этой таблицы в списке под кнопкой либо снять флаг в списке с таблицами коннектора (рисунок 21).

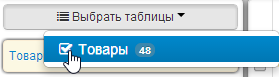


Рисунок – Исключение таблицы из списка загрузки

В ODBC-коннекторе помимо существующих таблиц можно создать новую с помощью произвольного SQL-запроса. Для этого нужно:

1. нажать пиктограмму «плюс» около кнопки «Выбрать таблицы» (рисунок 22);



Рисунок – Вызов функции создания новой таблицы в ODBC-коннекторе

1. в появившемся окне в поле «Название» ввести название таблицы, а в поле «Запрос к источнику данных» написать SQL-запрос к БД (рисунок 23);

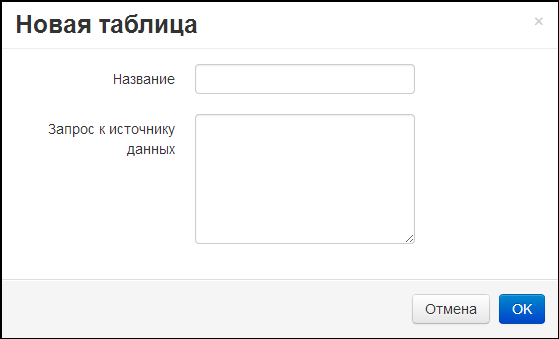


Рисунок – Форма запроса к источнику данных

1. нажать на кнопку «ОК». Новая таблица появится в перечне выбранных, и далее с ней можно работать так же, как и с другими таблицами.

По нажатию на любую из таблиц появится информация о выбранной таблице (рисунок 24), и далее можно переходить к процедуре настройки показателей для загрузки данных (см. п. 4.1.2.2).

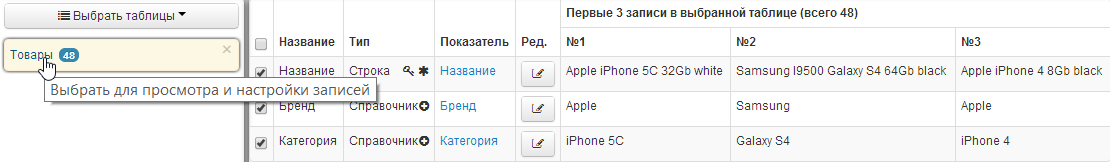


Рисунок – Форма просмотра данных таблицы

Для изменения параметров таблицы следует:

1. нажать на пиктограмму шестеренки (), расположенную справа от названия таблицы;

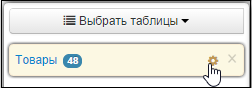


Рисунок – Выбор/отмена выбора всех показателей для загрузки

1. в открывшемся окне редактирования таблицы при необходимости изменить (рисунок 26):

* название таблицы;
* флаг «Загрузка в справочник». По умолчанию флаг «Загрузка в справочник» не используется, и данные таблицы загружаются не в справочник, а в обычный реестр. Этот флаг может использоваться для того, чтобы в дальнейшем было проще отделить данные, загруженные через определенный коннектор. Следует учитывать, что при загрузке в справочник большого объема данных (1000 записей и более), эта операция занимает много системных ресурсов.

Если установить данный флаг, то таблица будет загружена в справочник. Каждая строка таблицы будет являться элементом справочника.

Если при этом выбрать в поле «Справочник» вариант «Новый», то для загружаемой таблицы будет создан новый справочник. Название этого справочника будет таким же, как название таблицы; его можно изменить в поле «Название таблицы».

Для того чтобы загружать таблицу в существующий справочник, следует в списке «Справочник» выбрать соответствующее значение;

* флаг «Синхронизировать удаление» – если установить данный флаг, то при удалении записи в источнике данных эта запись будет удалена из Системы. Если на запись уже есть ссылки и удаление провести невозможно, то это приведет к ошибке загрузки данных;
* флаг «Пропускать дубликаты» – если установить данный флаг, то дубликаты по ключам таблицы не приведут к ошибке загрузки, а будут пропущены. Пропуск дубликатов производится после обработки вычисляемых полей. Из нескольких значений берется первое, кроме полей не ссылок с отметкой «Обработка синонимов», где объединяются значения через перевод строки;

1. для окончания настройки таблицы необходимо нажать «ОК».

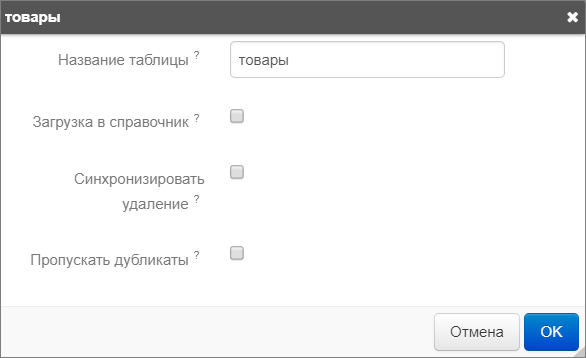


Рисунок – Настройка параметров загрузки

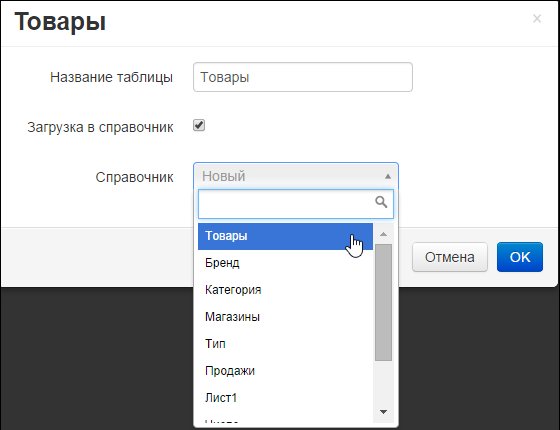


Рисунок – Параметры загрузки данных таблицы в справочник

#### Настройка показателей для загрузки данных

##### Виды загрузки данных в показатели системы

В Системе существуют следующие виды загрузки данных в показатели:

* значения показателя загружаются непосредственно из ячеек загружаемой таблицы в новый показатель;
* значения показателя загружаются непосредственно из ячеек загружаемой таблицы в показатель, существующий в Системе;
* значения показателя выбираются по ссылке из справочника, который существует в Системе;
* значения показателя выбираются по ссылке из другой таблицы того же источника данных;
* значения показателя вычисляются путем преобразования данных, полученных из того же источника данных;

В Системе реализована возможность как записывать данные в уже существующие показатели, так и создавать новые.

Кроме того, существует разделение по типам данных показателей. При обработке данных из источника Система автоматически определяет тип данных в каждом из столбцов.

Если в Системе уже существует показатель с таким же названием, какое указано в первой строке рассматриваемого столбца загружаемой таблицы, то Система автоматически («по умолчанию») подставляет его как показатель, соответствующий данному столбцу таблицы.

##### Настройка показателей

Выбрать таблицу для загрузки данных можно по нажатию на нее левой кнопкой мыши. После выбора таблицы для загрузки открывается область настройки показателей в правой части страницы, в том числе (рисунок 28):

* в левой части – состав полей (заголовков столбцов) таблицы:
* поле с флагами для выбора загружаемых показателей;
* «Название» – название показателей из загружаемой таблицы (берется из первой строки таблицы) ;
* «Тип» – указан текущий тип данных показателя;
* «Показатель» – указано название показателя в Системе. В этот показатель будут загружаться данные;
* «Ред.» – кнопки редактирования настроек показателей.
* в правой части – первые три строки с данными из таблицы, с которой ведется работа на настоящий момент (в транспонированном виде).



Рисунок – Форма просмотра деталей таблицы

Аналогично возможности загружать не все таблицы из файла (или источника данных), можно загрузить не все показатели, используемые в таблице. Для того чтобы выбрать показатели для загрузки, следует установить флаги в левом столбце в строке с названием показателя. Выбрать/снять выбор для всех показателей в таблице можно с помощью флага в заголовке таблицы (рисунок 29).

Чтобы впоследствии проще ориентироваться в загруженных данных, рекомендуется начинать загрузку с минимального числа показателей.



Рисунок – Выбор/отмена выбора всех показателей для загрузки

Для настройки показателя для загрузки данных и его типа необходимо:

1. в таблице, выбранной для загрузки, в строке интересующего показателя нажать на кнопку «Ред.» (рисунок 30);

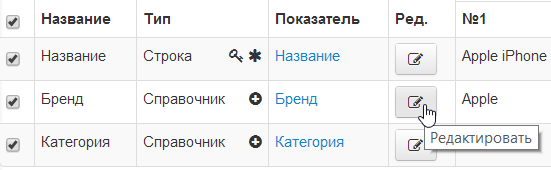


Рисунок – Вызов функции редактирования показателя

1. в открывшемся окне показателя для загрузки данных (рисунок 31) выбрать вид показателя:
2. «Значение» – значения этого показателя будут браться непосредственно из ячейки;
3. «Ссылка на таблицу» – значение в ячейке загружаемой таблицы будет ссылкой на другую таблицу этого же источника данных (сделать ссылку на таблицу из другого источника невозможно);
4. задать прочие настройки в зависимости от того, выполняется загрузка в существующий показатель (см. п. 4.1.2.2.2.1) или новый (см. п. 4.1.2.2.2.2);
5. нажать кнопку «ОК».

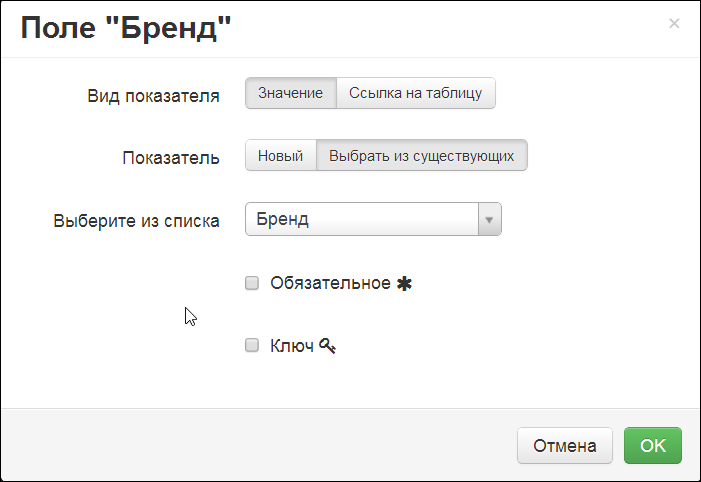


Рисунок – Настройка показателя для загрузки данных

###### Запись данных в существующие показатели

Для загрузки данных в существующий показатель необходимо в окне настройки показателя (рисунок 31):

1. выбрать значение «Выбрать из существующих» в поле «Показатель» (рисунок 32);

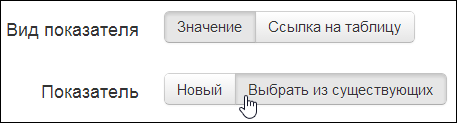


Рисунок – Выбор действия над показателями

1. выбрать показатель, существующий в Системе (рисунок 33). Показатели сгруппированы по типам данных, а также по ним работает поиск. Вводить полное название или его часть можно в специальном поле, которое находится в том же раскрывающемся списке;

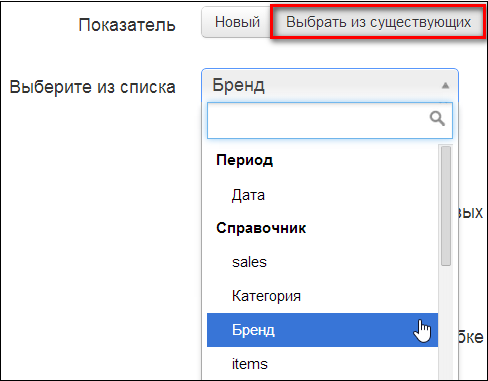


Рисунок – Выбор показателя из списка

###### Создание нового показателя

Для создания нового показателя необходимо в окне настройки показателя (рисунок 31):

1. в поле «Показатель» выбрать опцию «Новый»;
2. выбрать тип данных в раскрывающемся списке «Тип показателя» (рисунок 34);

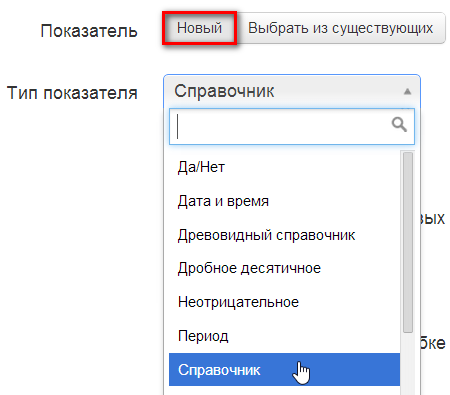


Рисунок – Форма создания нового показателя

1. в поле «Название показателя» при необходимости ввести название нового показателя. По умолчанию название совпадает с названием поля таблицы, указанным в первой строке загружаемой таблицы;
2. если в качестве типа данных показателя выбран «Справочник», следует указать справочник, из которого он будет брать свои значения (рисунок 35). Для удобства выбора при большом количестве справочников в верхней строке выбора есть поле поиска по названию.

Если нужного справочника в Системе нет, то в окне выбора «Справочник» нужно выбрать значение «Новый» (рисунок 36). Название вновь созданного справочника будет совпадать с введенным названием показателя.

Для сброса выбранного справочника нужно нажать на пиктограмму «крестик» в выбранном справочнике (рисунок 37).

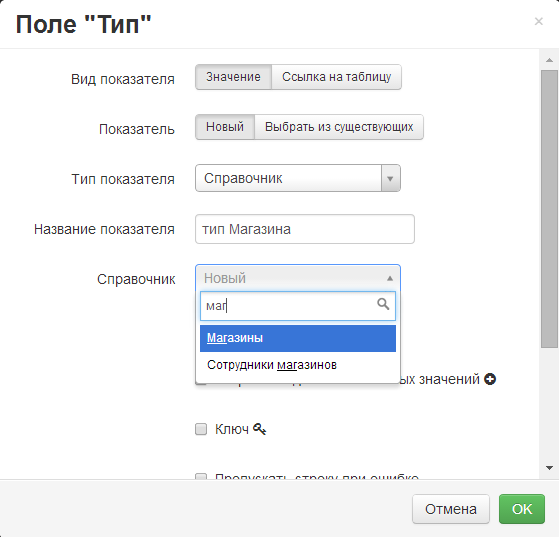


Рисунок – Выбор справочника в форме создания нового показателя с типом «Справочник»

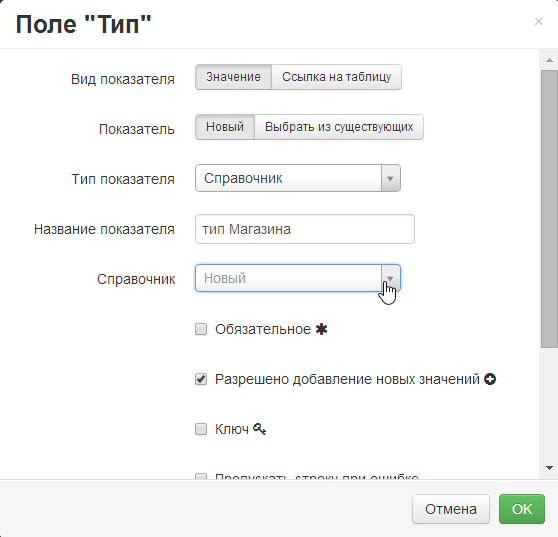


Рисунок – Создание справочника в форме создания нового показателя с типом «Справочник»

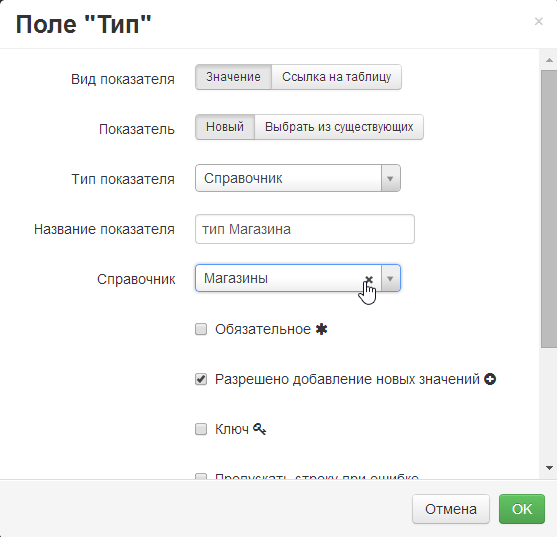


Рисунок – Отмена выбора справочника в форме создания нового показателя

При загрузке данные проходят форматно-логический контроль. Если данные не соответствуют указанному типу (в числовой показатель записывается текст, в показателе с типом «Целое» есть числа с дробной частью и т. п.), то таблица не будет загружена в Систему.

###### Ссылка на другую таблицу

Если в одной из таблиц определенного источника данных в каком-либо столбце содержатся значения, которые выбираются из другой таблицы того же источника, то для соответствующего этому столбцу показателя следует:

1. выбрать вид показателя не «Значение», а «Ссылка на таблицу» (рисунок 38);

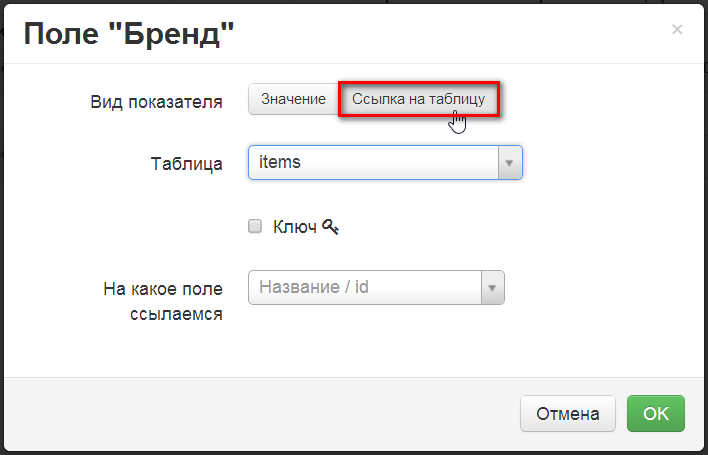


Рисунок – Ссылка на таблицу в форме создания нового показателя

1. указать таблицу источника данных, из которой выбираются значения;
2. выбрать поле таблицы, на которое должен ссылаться показатель (т. е. то поле таблицы, из которого выбираются значения).

Поле таблицы, на которое ссылается показатель, должно быть уникальным (т. е. каждое значение этого показателя единственно в таблице источника, из которой выбираются значения; например, уникальным рекомендуется сделать id).

Таблица, на которую ссылается какой-либо показатель из текущей таблицы, должна быть включена в загрузку в данном коннекторе. Необходимо задать в ней настройки для всех загружаемых показателей.

###### Настройка параметров загрузки

В окне настройки показателя можно указать дополнительные параметры загрузки (рисунки 31, 39):

* «Обязательное» – при загрузке будет проверяться наличие данных в каждой ячейке этого столбца. Если в каких-то ячейках данных нет, будет выведено сообщение об ошибке. В этом случае таблица не будет загружена;
* «Ключ» – при загрузке будет проверяться уникальность (единственность) значения этого показателя (либо сочетания двух и более ключевых показателей). Это позволяет избежать дублирования данных в Системе. При каждой новой загрузке данных из источника будет происходить обновление существующих данных в соответствии с заданным ключом.

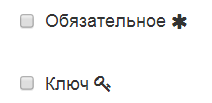


Рисунок – Настройка параметров загрузки

Если в качестве типа данных показателя указан справочник, появится возможность указать следующие настройки (рисунок 40):

* «Разрешено добавление новых значений» – все встреченные новые значения в столбце будут добавлены в справочник. Важно не забывать про эту настройку при первой загрузке показателя-справочника.
* «Пропускать строку при ошибке» – при ошибке (значение в ячейке этого показателя не встречается в справочнике в Системе) будет пропущена строка, содержащая эту ячейку, целиком. Все остальные строки таблицы будут загружены.
* «Пропускать поле при ошибке» – при ошибке (значение в ячейке этого показателя не встречается в справочнике в Системе) будет пропущена ячейка с этим показателем (для загружаемого в этой строке объекта данный показатель будет не определен). Данные для остальных показателей из этой строки и остальные строки таблицы будут загружены.

Справа от названий флагов указаны пиктограммы, которые будут отображаться в столбце «Тип» в строке показателя в левой таблице основной рабочей области окна для визуализации настроек коннектора.

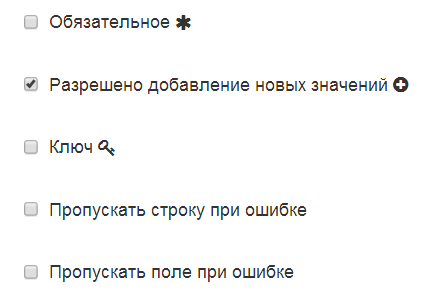


Рисунок – Настройка параметров загрузки при создании показателя с типом «Справочник»

##### Вычисляемые поля в коннекторах

В Системе существует возможность задания вычислений данных прямо в коннекторе для их дальнейшей загрузки.

Это может быть необходимо, если в данных содержатся ошибки, например, в названиях элементов будущего справочника содержатся лишние пробелы.

Также в вычисляемом поле можно настроить автозамену любых символов, если известно, что варианты наименования одного и того же элемента из разных файлов или баз данных различаются между собой, а данные о них следует привязывать к одному названию в Системе.

Кроме того, может возникнуть необходимость объединить несколько полей в одно (например, фамилию, имя и отчество в одно поле «ФИО») или наоборот, разделить одно поле из базы данных на два и более показателей в Системе.

Можно задавать вычисления как для уже существующих полей в коннекторе, так и создавать новые, полностью вычисленные поля.

###### Задание вычисления у существующего поля

Если необходимо задать вычисление уже существующего в коннекторе поля, следует:

1. в окне редактирования настроек загружаемого показателя установить флаг «Задать преобразование» (рисунок 41);

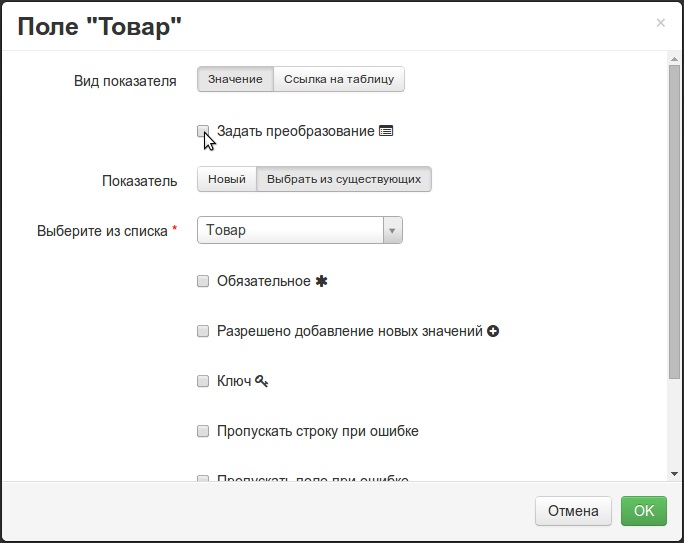


Рисунок – Добавление вычисляемых полей

1. в появившемся текстовое поле «Программа для вычисления» задать необходимые преобразования на языке Python (https://docs.python.org/2/, см. п. 4.1.2.2.3.3). Автоматически добавляется преобразование, обрезающие лишние пробелы (рисунок 42). Текущее (редактируемое) поле доступно через переменную field, все остальные поля доступны в переменной fields и f, необходимо указание их названия.

Примечание – Для получения подсказки следует нажать на пиктограмму «вопросительный знак» рядом с названием этого поля.

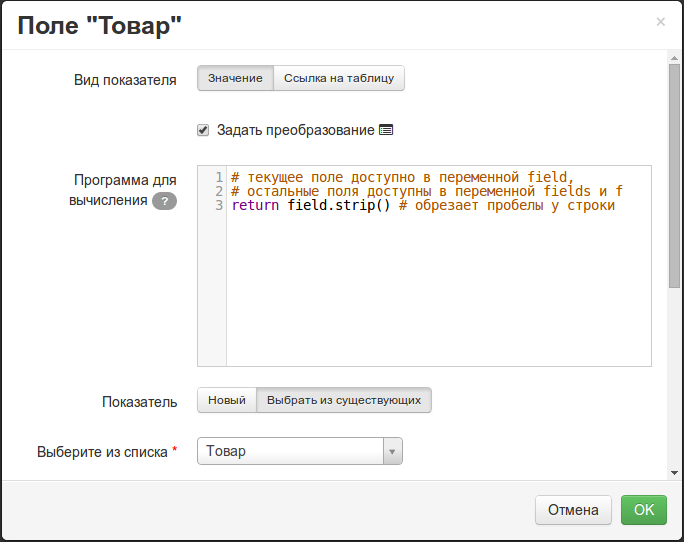


Рисунок – Форма настройки вычисляемых полей

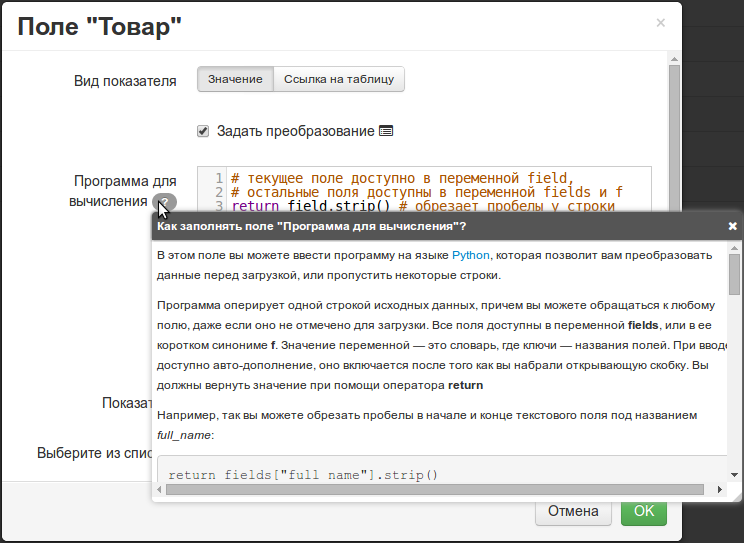


Рисунок – Отображение подсказки по процедуре создания вычисляемых полей

###### Создание нового вычисляемого поля

Также Система позволяет создать новое вычисляемое поле для загрузки данных. Для этого необходимо:

1. нажать на кнопку с пиктограммой «плюс» в левом верхнем углу области работы с загружаемыми данными;
2. в открывшемся окне (рисунок 44) заполнить поле «Программа для вычисления» (см. п. 4.1.2.2.3.3) и обязательное для заполнения поле «Название поля», куда нужно ввести название этого поля;
3. выбрать, в какой показатель будут загружаться вычисленные данные (в новый или в уже существующий);
4. если необходимо создать новый показатель, ввести его название и выбрать тип данных. Название показателя и название поля полностью независимы друг от друга;
5. задать все доступные данному типу данных настройки аналогично другим полям коннектора («обязательный», «ключ» и т. д.);

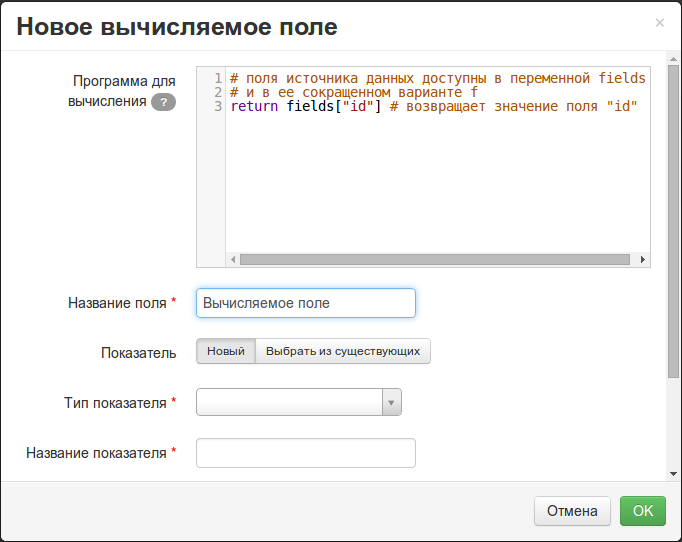


Рисунок – Форма создания нового вычисляемого поля

1. нажать кнопку «ОК» для сохранения внесенных данных (рисунок 45). Поле будет добавлено в таблицу с показателями для загрузки данных (рисунок 46).

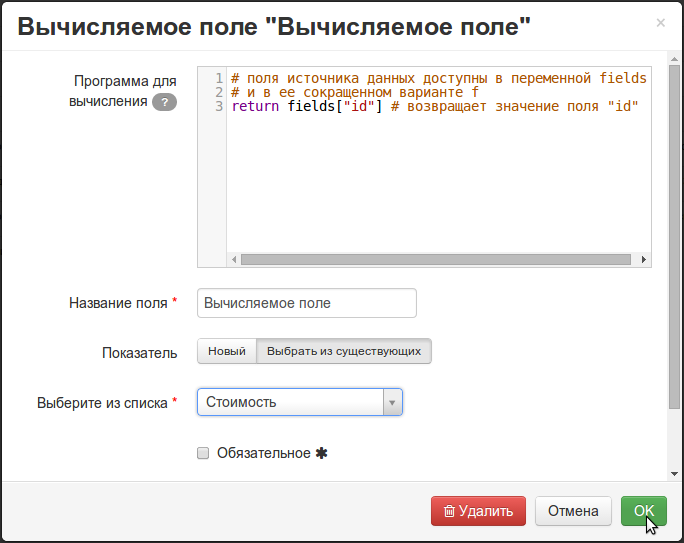


Рисунок – Сохранение созданного вычисляемого поля

Важно, что вычисляемое поле всегда обязательно для загрузки в коннекторе. Если необходимость загружать вычисляемое поле исчезла, нужно снова открыть окно его редактирования и удалить, нажав на соответствующую кнопку в окне.

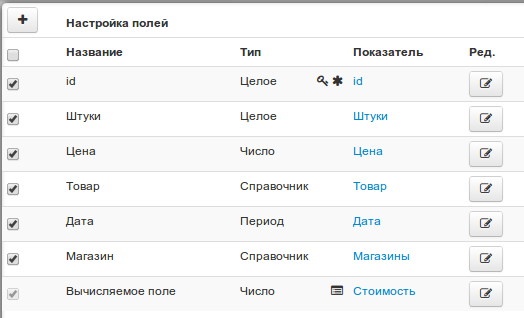


Рисунок – Отображение созданного вычисляемого поля в списке полей

###### Заполнение поля для вычисления

В окне редактирования настроек загружаемого показателя при заполнении поля «Программа для вычисления» можно ввести программу на языке Python, которая позволит преобразовать данные перед загрузкой или пропустить некоторые строки.

Программа оперирует одной строкой исходных данных, причем можно обращаться к любому полю, даже если оно не отмечено для загрузки. Все поля доступны в переменной fields или в ее коротком синониме f. Значение переменной – это словарь, где ключи – названия полей. При вводе доступно автодополнение, оно включается после того как, будет набрана открывающая скобка. В программе нужно вернуть значение при помощи оператора return.

Пример демонстрирует, как можно обрезать пробелы в начале и конце текстового поля под названием full\_name:

return fields["full\_name"].strip()

Пример демонстрирует, как можно разбить поле по дефису и вернуть первую часть:

return f["full\_name"].split('-')[0]

Пример демонстрирует, как можно обрезать последние два символа строки:

return f["some\_name"][:-2] # отрезать первые два символа: [2:]

Следует иметь в виду, что тип данных значения, который поступает на обработку, определяется типом в источнике данных и не зависит от типа показателя. Значения типа datetime.date приводятся к типу datetime.datetime.

Результат преобразования приводится к типу показателя того поля, в котором заданы вычисления.

Если задается программа преобразования существующего поля, то можно обращаться к нему, не указывая название, просто через переменную field. Например, так можно привести переменную к правильной единице измерения:

if field.endswith("m"):

return int(field.rstrip("m ")) \* 1024

else:

return int(field)

Доступен модуль регулярных выражений re – с его помощью можно извлекать фрагменты текста или производить замены. В этом примере извлекается число из поля index, которое возможно содержит и другую информацию:

m = re.search("\d+", f["index"])

if m:

return m.groups()[0]

Также можно использовать объекты из модуля datetime: timedelta для расчета разницы между моментами времени, и datetime для представления даты и времени. Например, используя встроенную переменную now, можно посчитать, сколько дней прошло с момента создания объекта:

return (now – f["created"]).days

Также есть переменная date, которая содержит текущую дату.

С помощью функции datetime.strptime можно распознать дату в любом формате, например:

return datetime.strptime(fields["date"], "%Y %m %d")

Можно пропустить некоторые записи, добавив исключение Skip. Например, так можно пропустить все записи, обновленные более двух часов назад:

if (now – f["updated"]).seconds > 2 \* 3600:

raise Skip

else: # возвращаем без изменений

return f["updated"]

Skip – глобальный объект во время загрузки. Для сложных вычислений можно сохранять в его атрибутах глобальные переменные. Пример нумерации всех строк загружаемых данных:

Skip.n = getattr(Skip, 'n', 0) + 1

return Skip.n

##### Настройка загрузки данных в справочник

Помимо показателей, в которые настраивается загрузка данных (см. п. 4.1.2.2), для каждого листа файла Microsoft Excel, а также для каждой таблицы из базы данных создается справочник и показатель, берущий значение из этого справочника. Названия этого справочника и показателя берутся из названия листа или таблицы.

При загрузке одностраничного файла Оборудование.xls будет создан справочник «Оборудование», элементами которого будут строки из этого файла, а также показатель «Оборудование» типа «Справочник», который ссылается на справочник «Оборудование».

Когда создается любой справочник, одно из полей рассматривается как название всей строки. Пусть, например, таблица с товарами имеет такую структуру:

| Оборудование | Производитель | Категория |
| --- | --- | --- |
| Оборудование Тип 01 | Производитель 01 | Категория 01 |
| Оборудование Тип 02 | Производитель 02 | Категория 02 |

Элементом справочника является одна строка, и Системе нужно указать, какое из полей использовать в качестве ее названия. Для этого нужно настроить загрузку названия в показатель «Название». В данном примере поле «Оборудование» должно загружаться в «Название».

Это необходимо для того, чтобы, когда потребуется вывести информацию про конкретный элемент справочника (товар), можно было обратиться к элементу справочника по названию, например «Оборудование Тип 01».

Если ни одно поле не загружается в показатель «Название», то Система будет выводить элементы справочника по id. Другими словами, в отчете вместо «Оборудование Тип 01» будет выводиться идентификатор, сформированный Системой.

Выше говорилось, что по умолчанию каждая таблица загружается в справочник. Название справочника берется из названия таблицы, а название элементов справочника из того поля, которое загружается в показатель «Название». Эту возможность можно использовать для быстрого создания и обновления справочников, в том числе иерархических.

Название справочника можно изменить в настройках таблицы (рисунок 47), по умолчанию оно берется из названия исходной таблицы.

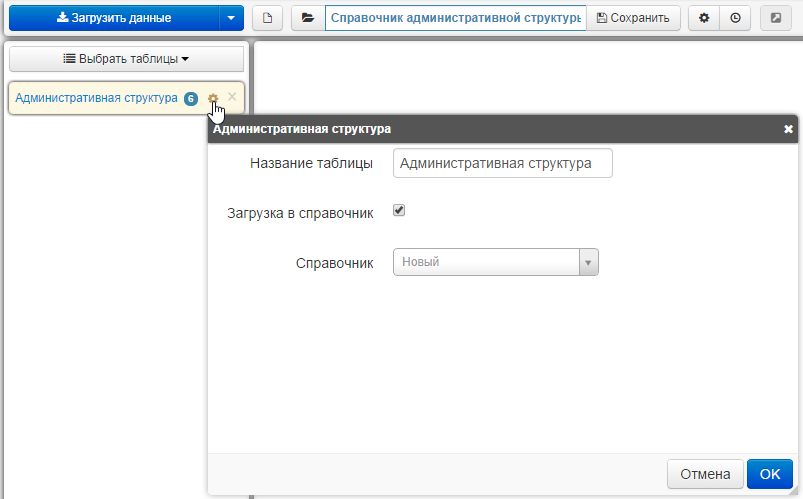


Рисунок – Загрузка справочника

Для того чтобы сделать справочник иерархическим, для всех его элементов (кроме корневых) должны быть указаны родительские элементы (рисунок 48).

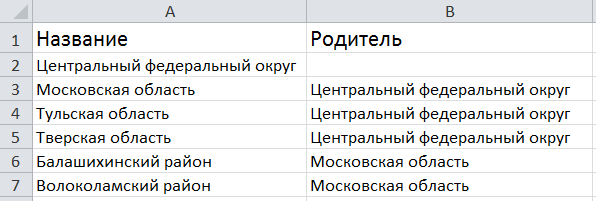


Рисунок – Пример структуры данных иерархического справочника для загрузки

В настройках коннектора по такой таблице следует указать, что показатель «Родитель» является ссылкой на эту же таблицу, на показатель «Название» (рисунок 49).

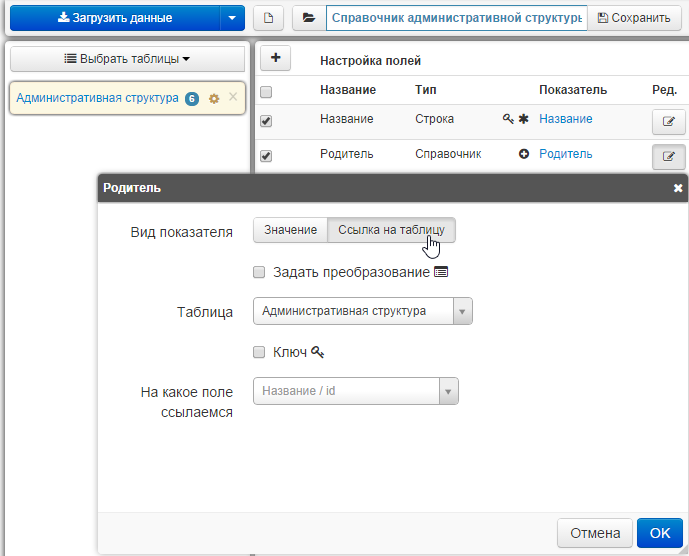


Рисунок – Настройка параметров загрузки справочника

Важно, чтобы все значения «родителя» имелись в показателе «Название», поскольку в «Названии» должны быть все элементы справочника, в том числе родительские.

Пока коннектор не сохранен, таблица тоже не сохранена, поэтому в ссылке «родителя» выводится «Целое» (рисунок 50).

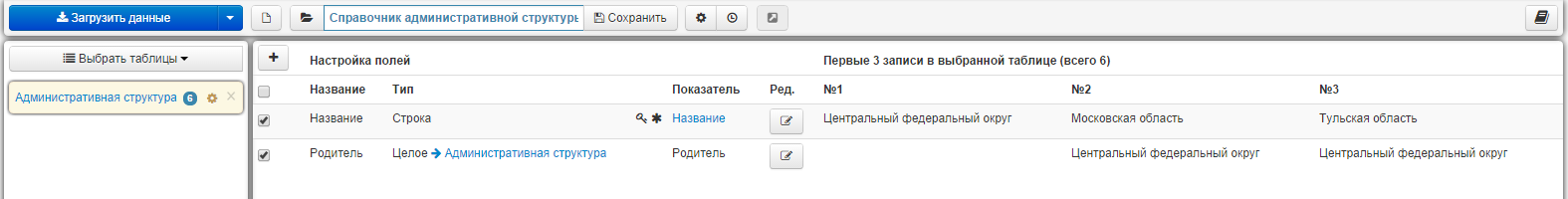


Рисунок – Форма настройки полей справочника

После сохранения коннектора или загрузки данных ссылка с «родителя» на текущую таблицу отображается в явном виде (рисунок 51).

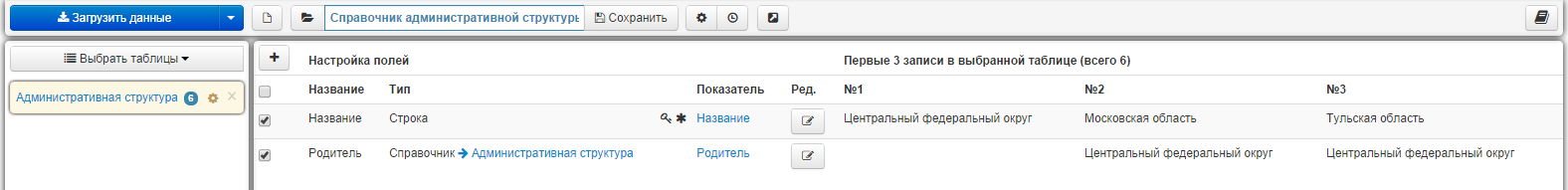


Рисунок – Форма настройки полей справочника после сохранения коннектора или загрузки данных

Для того чтобы убедиться, что справочник действительно получился иерархическим, можно открыть этот справочник и увидеть иерархическую структуру значений показателя «Родитель» (рисунок 52). Возможно, для этого придется обновить страницу (Сtrl + f5).

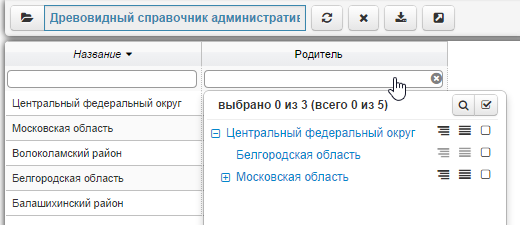


Рисунок – Просмотр структуры загруженного справочника

#### Сохранение коннектора

После того как заданы настройки для загрузки всех нужных показателей во всех интересующих таблицах, можно их сохранить. Для этого следует нажать кнопку «Сохранить» справа от поля с названием коннектора в верхнем меню страницы (рисунок 53).

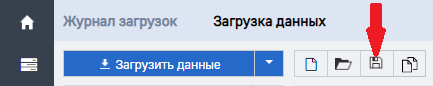


Рисунок – Сохранение коннектора

Также можно сразу перейти к загрузке данных в Систему по нажатию кнопки «Загрузить данные» (рисунок 53). Настройки коннектора при этом сохранятся автоматически.

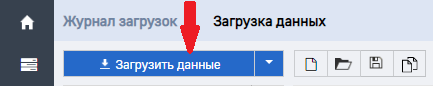


Рисунок – Вызов функции загрузки данных

При сохранении коннектора (при загрузке данных или по кнопке «Сохранить») в Системе создаются все новые показатели.

#### Загрузка данных

Прежде чем приступать к загрузке данных, необходимо настроить коннектор и показатели для загрузки (см. пп. 4.1.2.1, 4.1.2.2. 4.1.2.2.4).

##### Загрузка данных из всех таблиц коннектора

По нажатию на кнопку «Загрузить данные» на странице работы с коннектором (рисунок 11) происходит загрузка всех выбранных данных в Систему. Загрузка таблиц с большим числом строк может занять значительное время.

При загрузке данных происходит форматно-логический контроль загружаемых данных. Если какие-либо записи из одной из загружаемых таблиц его не прошли (несоответствие типов данных, отсутствие нужных значений в справочниках), не будет загружена ни одна из таблиц. В нижней части страницы появится сообщение об ошибках.

Новые показатели в Системе будут созданы, даже если никакие данные не загрузятся.

Если файл (или база данных) содержит данные, которые уже загружались в Систему, и ни один показатель при настройке коннектора не был отмечен как уникальный, такие данные будут восприняты Системой как самостоятельные и будут загружены повторно.

После успешной загрузки данных выводится сообщение с информацией о загрузке: сколько записей создано, обновлено и удалено (при повторных загрузках и с флагом «Синхронизировать удаление») по каждой из таблиц (рисунок 55).

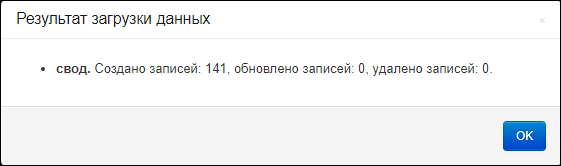


Рисунок – Форма отображения результата загрузки данных

##### Загрузка данных из одной таблицы

В Системе также существует возможность загрузить только отображаемую на экране таблицу, даже если для загрузки через коннекторы было выбрано больше одной таблицы.

Для этого необходимо на странице работы с коннектором (рисунок 11) нажать пиктограмму «стрелка» на кнопке «Загрузить данные» и в списке выбрать пункт «Только выбранную таблицу» (рисунок 56).

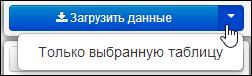


Рисунок – Загрузка выбранной таблицы

При этом будет загружена таблица, отображенная в данный момент на странице, а также все связанные с ней таблицы (например, если в таблице есть ссылка на соседнюю таблицу, то соседняя таблица тоже будет загружена).

Все настройки коннектора для других таблиц сохранятся в Системе.

Это функция удобна, например, если одно из полей таблицы должно ссылаться на другую таблицу.

Поле «Магазин» в таблице «Продажи» ссылается на таблицу «Магазины», находящуюся в том же файле. Можно либо сделать два отдельных коннектора для каждой из таблиц, либо сначала загрузить отдельно таблицу с магазинами, а затем – таблицу «Продажи» со ссылкой на «Магазины».

##### Настройка периодической загрузки данных

В Системе существует возможность настройки периодической загрузки данных. Для этого необходимо:

1. на странице работы с коннектором (рисунок 11) нажать пиктограмму «Периодическая загрузка», расположенную в верхнем меню страницы (рисунок 57);

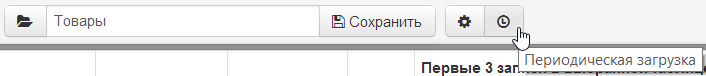


Рисунок – Вызов функции настройки периодической загрузки

1. в открывшемся окне настройки периодической загрузки данных в левом столбце «Расписание» указано время и дни, когда должна производиться загрузка, в правом столбце «Таблицы» указано, какие таблицы необходимо загружать в указанное время. Чтобы включить или выключить созданный шаблон периодической загрузки, следует установить или снять флаг рядом с настройкой времени (рисунок 58);

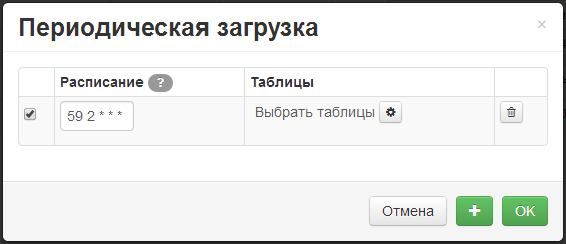


Рисунок – Форма настройки параметров периодической загрузки

1. чтобы указать время и дни, когда необходимо производить загрузку, следует ввести их в поле «Расписание». Подсказка о том, как это правильно делать, появляется при нажатии на вопросительный знак рядом с заголовком этого столбца (рисунок 59);



Рисунок – Подсказка по управлению загрузкой данных по расписанию

1. чтобы настроить, какие таблицы необходимо загружать в это время, следует нажать на пиктограмму «шестеренка» в столбце «Таблицы» и отметить нужные таблиц флагами. В список попадут только те таблицы из источника данных, которые будут отмечены как загружаемые (рисунок 60).

В правом верхнем углу окна находятся:

* кнопка с пиктограммой «лупа» позволяет открыть строку поиска и выполнить поиск при большом количестве загружаемых таблиц в источнике данных;
* флаг, который позволяет выделить/снять выделение всех таблиц;

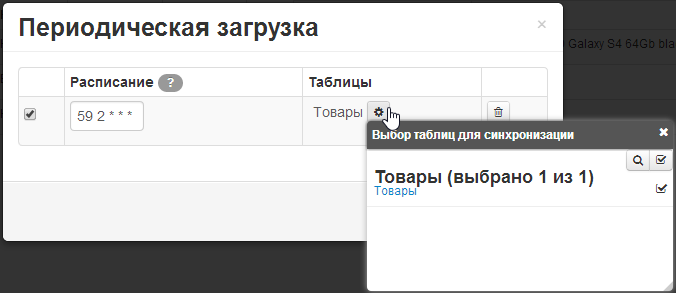


Рисунок – Выбор таблиц для периодической загрузки

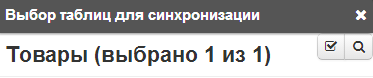


Рисунок – Форма выбора таблиц для синхронизации

Для того чтобы удалить настройку периодичности, необходимо нажать на пиктограмму «корзина» в правом столбце настройки (рисунок 62).



Рисунок – Удаление настройки периодической загрузки

Чтобы добавить новую строку с настройкой, следует нажать на зеленую кнопку «+» в нижней части окна и нажать на кнопку «ОК» в нижней части окна, чтобы вернуться к работе с коннекторами после завершения необходимых настроек (рисунок 63) (либо на кнопку «Отмена», если сохранять измененные настройки не нужно).

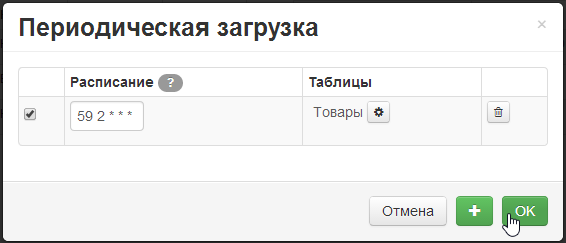


Рисунок – Сохранение настроек периодической загрузки

##### Просмотр загруженных данных

Для того чтобы просмотреть загруженные данные, следует нажать на кнопку «Перейти к просмотру загруженных таблиц» (рисунок 64). Откроется реестр с загруженными данными в режиме просмотра.

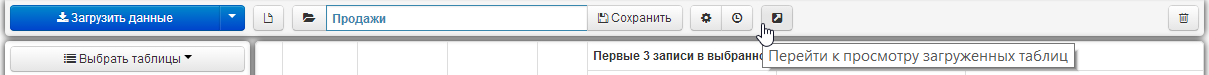


Рисунок – Вызов функции просмотра загруженных данных

Для возвращения к работе с коннектором нужно нажать на эту же кнопку (рисунок 65).

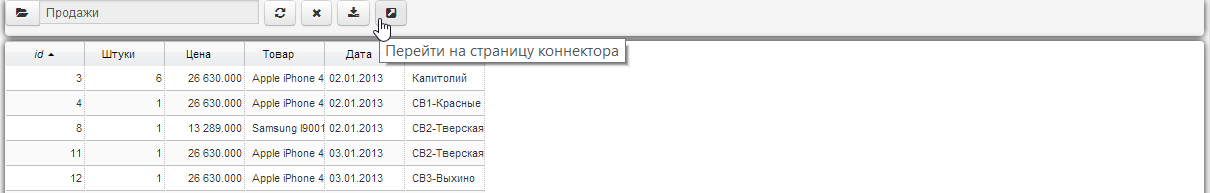


Рисунок – Возврат к форме работы с коннектором

#### Открытие и работа с сохраненным коннектором

Для работы с существующим коннектором следует на странице работы с коннектором (рисунок 11) нажать на пиктограмму «Открыть», расположенную слева от поля, где выводится название текущего коннектора, из списка выбрать нужный коннектор и нажать на его название (рисунок 66). Через некоторое время загрузится выбранный коннектор со всеми настройками. Дальнейшая работа аналогична работе с новым коннектором.



Рисунок – Переход к форме работы с коннектором

Если необходимо обновить настройки коннектора в Системе (обновить файл с данными, изменить название коннектора), необходимо:

1. нажать на кнопку «Настройки» в верхнем меню окна (рисунок 67);



Рисунок – Вызов функции управления настройками коннектора

1. после этого откроется всплывающее окно, аналогичное тому, что появлялось при создании нового коннектора. В нем можно заново приложить файл, из которого будут загружаться данные, ввести новое название для коннектора, скачать текущий файл с данными.

В одном коннекторе необходимо использовать файл с одним и тем же названием и с одной и той же структурой данных. Если необходимо загрузить файл с отличающейся структурой данных, следует создать новый коннектор;

1. для возврата к работе с коннектором нажать кнопку «ОК», если нужно сохранить изменения (или «Отмена», если сохранять внесенные изменения не нужно).

Если был обновлен файл с данными, нужно сначала нажать кнопку «Сохранить» в верхнем меню страницы и только после этого можно приступать к загрузке данных (кнопка «Загрузить данные» в верхнем меню страницы).

Также изменить название коннектора можно в поле в верхнем меню на странице работы с коннектором. После изменения названия следует нажать кнопку «Сохранить» справа от этого поля (рисунок 68).



Рисунок – Изменение названия коннектора

#### Удаление коннектора

Для удаления сохраненного в Системе коннектора необходимо:

1. нажать на кнопку «Удалить коннектор» в правом верхнем углу страницы (рисунок 69);



Рисунок – Удаление коннектора

1. в диалоговом окне указать, необходимо удалить только коннектор или коннектор со всеми справочниками, показателями, данными, которые используются в Системе только в этом коннекторе (в этом случае следует установить флаг «Удалить вместе с данными и показателями»), после чего нажать на кнопку «Удалить». Для отмены действия следует нажать кнопку «Отмена».

Примечание – При удалении коннектора и данных из Системы данные в источнике (файле, базе данных), из которого они загружены, не затрагиваются.

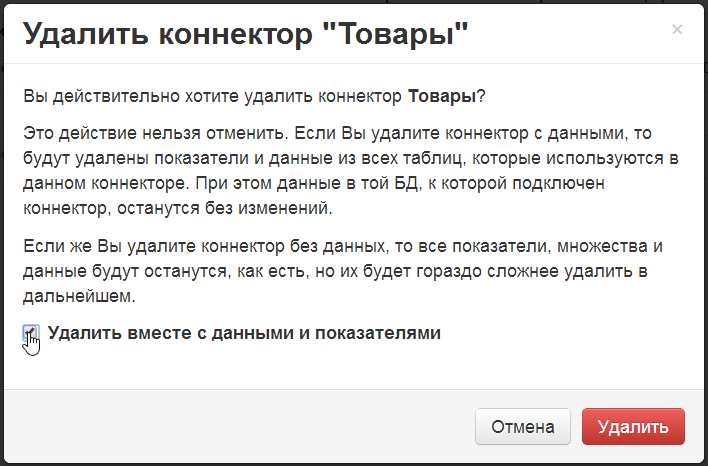


Рисунок – Запрос подтверждения удаления коннектора

#### Журнал загрузок

Журнал загрузок содержит информацию о данных, загружаемых через те или иные коннекторы, дате и времени загрузки, объеме загруженных данных, ошибках загрузки, а также предоставляет возможность просмотра загруженных таблиц.

Для того чтобы просмотреть историю загрузок данных, следует выбрать в главном меню пункт «Журнал загрузок». Откроется страница с журналом загрузок из всех коннекторов (рисунок 71).

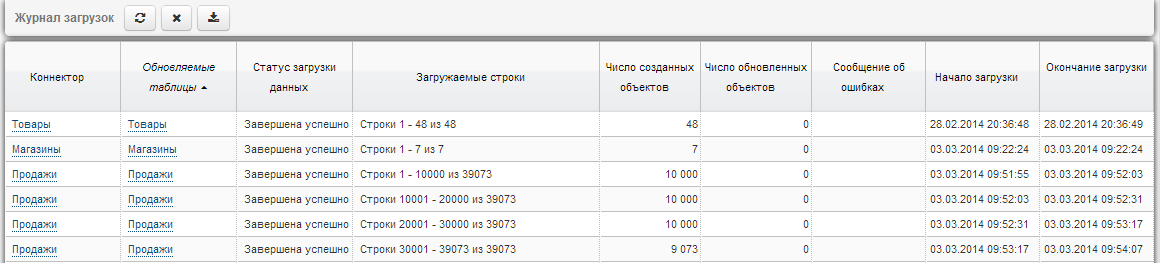


Рисунок – Журнал загрузки коннекторов

В верхней части страницы расположены кнопки (рисунок 72):

* «Обновить журнал» – предназначена для обновления информации на странице;
* «Сброс настроек размеров заголовков» – сбрасывает размеры заголовков до настроек по умолчанию;
* «Экспортировать в xls» – позволяет экспортировать журнал загрузок в файл Microsoft Excel.



Рисунок – Кнопки управления данными журнала загрузок

Работа с таблицей осуществляется так же, как с другими таблицами (отчетами и реестрами): можно изменять ширину и высоту заголовков столбцов, применять сортировку по нажатию на заголовок и т. п.

По нажатию на ссылку в столбце «Коннектор» откроется страница этого коннектора (рисунок 73).

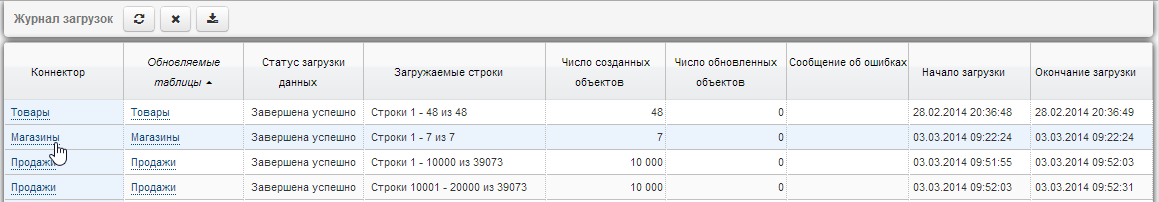


Рисунок – Выбор коннектора для просмотра деталей загрузки

По нажатию на ссылку в столбце «Обновляемые таблицы» откроется реестр с загруженными из этой таблицы данными (рисунок 74).

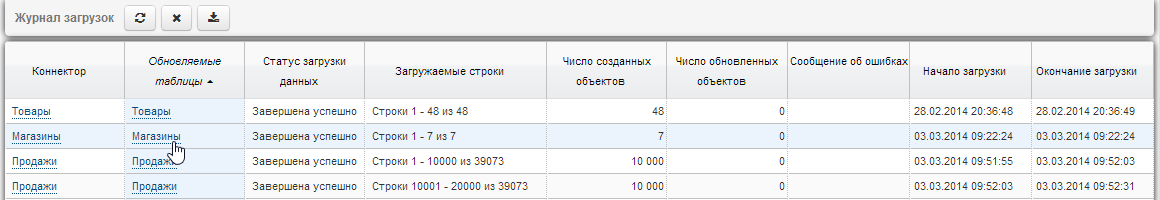


Рисунок – Выбор таблицы для просмотра загруженных данных

Если в столбце «Статус загрузки данных» указано, что есть ошибки, то эти ошибки выводятся в столбце «Сообщение об ошибках» (рисунок 75).

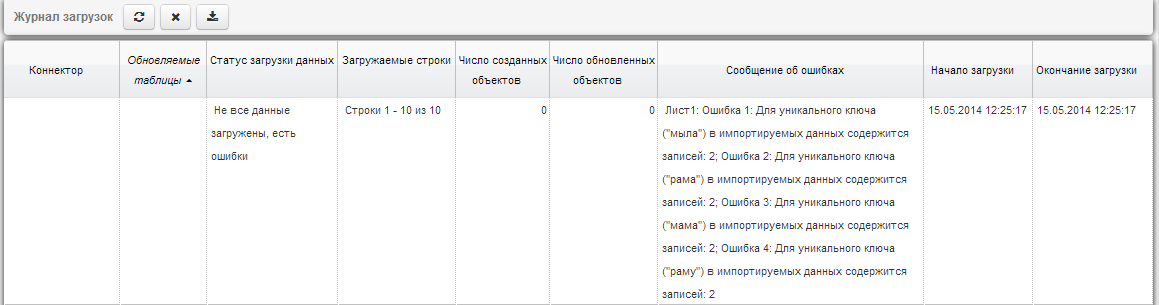


Рисунок – Вывод сообщений об ошибках загрузки данных

Большие файлы загружаются в несколько заходов, по 10 000 строк (рисунок 76).

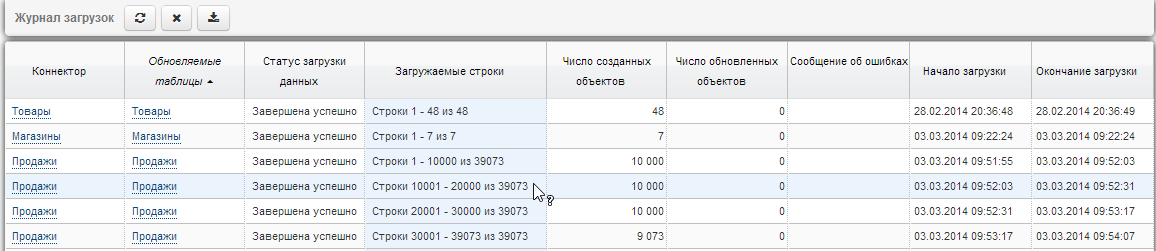


Рисунок – Отображение статуса загрузки больших (более 10 тысяч строк) файлов

Если после загрузки данных коннектор был удален, то запись о загрузке останется, но в столбцах «Коннектор» и «Обновляемые таблицы» будет пусто (рисунок 77).

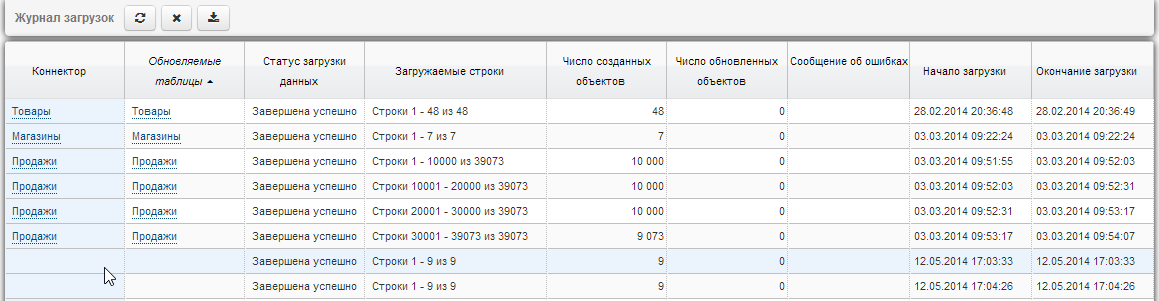


Рисунок – Отображение информации о загрузке при удалении коннектора

### Настройка процесса сбора данных

#### Общие сведения о процессе настройки сбора данных

Функциональность компонента «Конструктор сбора данных» обеспечивает настройку процессов сбора данных в соответствии с установленными регламентами сбора отчетности. В Системе предусмотрена возможность без программирования задавать:

* количество этапов сбора (шагов согласования);
* порядок этапов;
* регламентированный срок на выполнение каждого этапа;
* список ролей, которые на каждом шаге могут вносить изменения в собираемые данные;
* список действий, доступный каждой роли на текущем статусе.

Выделены следующие основные категории ролей:

* пользователи-администраторы – данной категории пользователей предоставляется возможность через интерфейс Системы создавать новые маршруты сбора данных, в том числе добавлять новые статусы и роли;
* пользователи-координаторы (редакторы шаблона) – данной категории пользователей предоставляется возможность создавать, редактировать и запускать шаблоны сборов, но только тем организациям и по тем тематикам, на которые пользователь-администратор выдал права;
* пользователи-исполнители – выполняют задания сбора (вводят данные), могут просматривать свои выполненные задания и архив заданий (закрытые задания). При получении задания сбора пользователь-исполнитель должен заполнить формы, которые получил в задании и затем поставить отметку о выполнении (нажать кнопку);
* пользователи-контролеры – это пользователи, ответственные за сбор, и другие согласующие роли (количество согласующих ролей определяется маршрутом сбора данных). Пользователи-контролеры проверяют задания сбора, выполненные исполнителем, просматривают ранее проверенные задания, архив заданий (закрытые задания) и отчеты по исполнительской дисциплине.

Используются следующие основные термины:

* рабочий процесс сбора – правила, определяющие количество участников (ролей сбора) и маршрут сбора данных (сколько всего статусов, правила перевода задания из статуса в статус);
* тип сбора – название для конкретного рабочего процесса сбора;
* тематика – направленность, зона ответственности сбора (например, «Бухгалтерия», «Безопасность»);
* шаблон сбора – правила, определяющие сроки сбора и список форм для заполнения. После запуска шаблона пользователям автоматически формируются задания и запускаются по маршруту, заданному в рабочем процессе сбора;
* структура сбора – справочник (плоский или иерархический), по которому осуществляется сбор (например, справочник организаций, справочник «Административная структура»);
* узел сбора – одно из значений справочника «Структура сбора» (например, ООО «Организация»);
* периодичность сбора – тип периода, которым параметризованы формы сбора (например, если формы ежемесячные, то периодичность сбора – месяц);
* период сбора – конкретный период, указанный в задании, за который заполняются данные (например, если периодичность сбора год, то период сбора  – 2018 год);
* доступы в сборе – правила, определяющие связку: «Пользователь – Тематика – Тип сбора – Организация (узел сбора) – Роль»;
* рекомендованные формы/отчеты/реестры – это сущности Системы, которые участники сбора должны заполнить/проверить/посмотреть. Особенность рекомендованных форм/отчетов/реестров состоит в том, что они должны быть параметризованы как минимум двумя показателями: «Структурой сбора» и «Периодичностью сбора». В каждом задании сбора они будут подставляться автоматически, и у пользователя не будет права изменить их (исправить прошлые данные);
* показатель для утверждения.

#### Общая процедура настройки сбора данных

После создания и настройки всех необходимых форм, реестров и отчетов можно переходить к созданию и настройке сбора данных. Для этого необходимо:

1. создать показатель с типом «Справочник» и справочник (см. п. 4.1.1.4), по которому будет осуществляться сбор данных;
2. создать показатель с типом «Множество значений из справочника», по которому будет выполняться сбор данных.
3. наполнить справочник, по которому будет осуществляться сбор данных (см. п. 4.1.1.5);
4. создать «Тематику сбора» в справочнике «Тематики» (см. п. 4.1.3.3.6);
5. назначить права пользователю на создание шаблонов сбора (в том числе пользователю admin) в справочнике «Доступы (см. п. 4.1.3.3.5). Для типов сбора по умолчанию – это роль «Редактор шаблона»;
6. создать шаблон сбора данных (см. п. 4.1.3.4.2);
7. включить шаблон сбора данных.

Если в сборе данных используется справочник «Административная структура», шаги 1 и 2 могут быть пропущены (данные показатели создаются автоматически).

#### Общие настройки процесса сбора данных в системе

##### Типы сбора данных, используемые в системе. Справочник «Типы сбора»

Сведения об основных типах сбора данных, используемых в Системе, перечислены в справочнике «Типы сбора» (рисунок 78). Для перехода к справочнику следует в главном меню Системы выбрать пункт «Настройки сбора», далее вкладка «Список типов сбора».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5. Существующие по умолчанию типы сбора изменить нельзя, создать новый – можно.

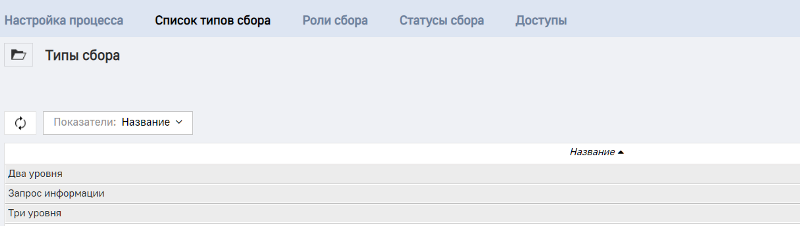


Рисунок – Справочник «Типы сбора»

По умолчанию в Системе предусмотрены следующие типы сбора:

* «Два уровня»:
  1. сбор с тремя основными статусами:
* «Новое – задание требует выполнения исполнителем (роль «Исполнитель»), у исполнителя одна кнопка «Выполнить».
* «Выполнено» – задание требует проверки ответственным за сбор (роль «Ответственный за сбор»): может быть переведено контролером в статус «Новое» (кнопка «На доработку») или «Закрыто» (кнопка «Закрыть»).
* «Закрыто» – задание в архиве;
  1. дополнительно введено три статуса:
* «Новое просрочено» – Система автоматически переводит задание в данный статус из статуса «Новое» в последний день периода сбора;
* «Сбор закрыт: новое» – Система автоматически переводит задание в данный статус из статусов «Новое» и «Новое просрочено» через 5 дней после окончания периода сбора;
* «Сбор закрыт: выполнено» – Система автоматически переводит задание в данный статус из статуса «Выполнено» через 5 дней после окончания периода сбора;
* показатель для утверждения не присваивается (можно заполнить на каждом стенде, но нельзя изменить после однократного заполнения – только создать новый тип сбора);
* редактировать шаблоны сбора с данным типом сбора могут пользователи, имеющие роль сбора «Редактор шаблона»;
* задание становится доступным для выполнения (видимым в «Текущих заданиях» у исполнителя) сразу после запуска шаблона сбора, если не указано иное в «Шаблоне сбора» («Дней от начала периода сбора» – «0», но может быть отредактировано в шаблоне);
* **«**Три уровня» (дополнительно к сбору «Два уровня» добавлен статус «На проверке»):
* «Новое» – задание требует выполнения исполнителем (роль «Исполнитель»), у исполнителя одна кнопка «На проверку»;
* «На проверке» – задание требует проверки Контролером (роль «Контролер»), может быть переведено в статус «Новое» (кнопка «На доработку) или «Выполнено» (кнопка – «Выполнить»);
* «Выполнено» – задание требует проверки Ответственным за сбор (роль «Ответственный за сбор»), может быть переведено контролером в статус «Новое» (кнопка «На доработку») или «Закрыто» (кнопка «Закрыть»);
* «Закрыто» – задание в архиве;
* «Запрос информации»:
* сбор с двумя статусами:
* «Новое» – задание требует выполнения исполнителем (роль «Исполнитель»), у исполнителя одна кнопка «Закрыть».
* «Закрыто» – задание в архиве;
* показатель для утверждения не присваивается (можно заполнить на каждом стенде, но нельзя изменить после однократного заполнения – только создать новый тип сбора);
* редактировать шаблоны сбора с данным типом сбора могут пользователи, имеющие роль сбора «Редактор шаблона»;
* Задание становится доступным для выполнения (видимым в «Текущих заданиях» у исполнителя) сразу после запуска шаблона сбора («Дней от начала периода сбора» – «0», но может быть отредактировано в шаблоне).

##### Настройка рабочего процесса сбора. Справочник «Рабочий процесс сбора»

Маршруты сборов описаны в справочнике «Рабочий процесс сбора» (рисунок 79). Данные справочника используются при создании шаблонов сбора данных (см. п. 4.1.3.4.2).

Для перехода к справочнику следует в главном меню Системы выбрать пункт «Настройки сбора», далее вкладка «Настройки процесса».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5. Каждая строка справочника соответствует одному переходу задания из статуса в статус.

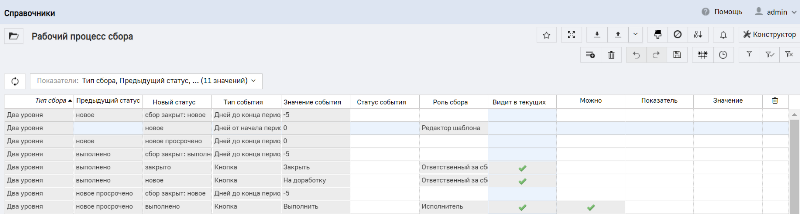


Рисунок – Справочника «Рабочий процесс сбора»

Описание полей справочника:

* «Тип сбора» – название типа сбора. Редактирование названия, добавление новых типов сбора осуществляется в справочнике «Типы сбора» (см. п. 4.1.3.3);
* «Предыдущий статус» – название статуса, для которого описывается правило: из какого статуса задание перейдет в следующий статус. Редактирование списка статусов осуществляется в справочнике «Статусы» (см. п. 4.1.3.3.4);
* «Новый статус» – название статуса, для которого описывается правило: в какой статус задание перейдет из предыдущего статуса;
* «Тип события» – это событие, по которому задание, сформированное по данному шаблону, поменяет статус с поля «Предыдущий статус» на «Новый статус». Существуют следующие типы событий (дополнительно см. п. 4.1.3.3.2.1):
* «Дней до конца периода сбора» – переход осуществляется автоматически, когда пройдет столько дней от конца периода сбора, сколько указано в поле «Значение события». Поле «Роль сбора» при этом можно оставить пустым или заполнить – переход все равно будет осуществляться автоматически. Конец периода сбора – это последний день периода сбора, указанного в шаблоне сбора (не в справочнике «Рабочий процесс сбора»). Например, если период сбора в шаблоне сбора указан месяц, то можно задать условие: задание становится просроченным за три дня до конца месяца/через три дня после окончания.
* «Дней от начала периода сбора» – переход осуществляется автоматически. Описание см. для поля «Дней до конца периода сбора»;
* «Дней от статуса» – переход осуществляется автоматически, после окончания дней, указанных в поле «Значение события», от момента, когда задание перешло в статус, указанный в поле «Статус события»;
* «Кнопка» – переход осуществляется, когда пользователь, имеющий роль, указанную в поле «Роль сбора», нажмет в паспорте задания кнопку, название которой указано в поле «Значение события»;
* «Значение события» – вторая часть поля «Тип события». Данное поле может быть изменено в шаблоне сбора, в «Типе сбора» задается значение «по умолчанию»:
* для типов событий «Дней до конца периода сбора» и «Дней от начала периода сбора» – положительное/отрицательное/нулевое значение;
* для типа события «Дней от статуса» – положительное значение (>0);
* для типа события «Кнопка» – «текстовое поле» (то есть название кнопки, которое будет в паспорте);
* «Статус события» – заполняется только для типа события «Дней от статуса»;
* «Роль сбора» – заполняется только для типа события «Кнопка». В ином случае игнорируется Системой. Исключения:
* исключение 1 – роль, отмеченная для перехода из «пустого» статуса (значение поля «Предыдущий статус» – пусто) в «Новое» (значение поля «Новый статус» – «Новое») – это роль, позволяющая создавать и редактировать шаблон (можно использовать системную роль «Редактор шаблона» или иную, если этого требует постановка задачи);
* исключение 2 – роль заполняется не для определения видимости кнопки, а для определения поля «Видит в текущих» (см. ниже пример 2);
* «Видит в текущих» – если указано «Да», то пользователь с ролью «Роль сбора» увидит данное задание в своем списке «Текущие задания», когда оно находится в статусе «Предыдущий статус». Как правило, заполняется только для типа события «Кнопка», но необязательно. Примеры:
* пример 1: пользователь с ролью «Исполнитель» должен видеть задание в статусе «Новое» и «Новое просрочено» с кнопкой «Выполнить». Тогда в строках, у которых поле «Предыдущий статус» заполнено как «Новое» или «Новое просрочено», «Тип события» – «Кнопка», «Значение события» – «Выполнить», «Роль сбора» – «Исполнитель», должно быть указано в поле «Видит в текущих» значение «Да»;
* пример 2: пользователь с ролью «Контролер» должен видеть задание в статусе «Проверено» 3 дня, но не должен иметь возможность сделать что-то с ним (например, задание используется как напоминание). Тогда в строке, у которой в поле «Предыдущий статус» – «Проверено», «Тип события» – «От статуса», «Значение события» – «3», «Статус» – «Проверено», «Роль сбора» – «Контролер», должно быть указано в поле «Видит в текущих» значение «Да»;
* «Можно редактировать» – если указано значение «Да», пользователь с ролью «Роль сбора» сможет редактировать формы и реестры, если задание находится в статусе «Предыдущий статус». Как правило, заполняется только для типа события «Кнопка», но необязательно;
* «Показатель утверждения» – название показателя, которому будет присвоено определенное значение (указанное в поле «Значение утверждения») при изменении статуса. Данное поле может быть изменено в «Шаблоне сбора», в поле «Тип сбора» задается значение «по умолчанию». Данный показатель необязательно должен быть в формах/реестрах для заполнения;
* «Значение утверждения» – вторая часть поля «Показатель утверждения»: значение показателя (например, «Да»). Например, можно при изменении статуса из «Новое» в «Выполнено» показателю «Утверждено пользователем» присвоить значение «Да»; при изменении статуса из «Выполнено» в «Закрыто» показателю «Утверждено контролером» присвоить «Да». Следует обратить внимание, что требуется заполнить все строки, когда требуется присвоить показатель: при переходе из статуса «Новое просрочено» в статус «Выполнено» значение показателя утверждения тоже должно присваиваться, строку надо заполнить повторно. Данное поле может быть изменено в «Шаблоне сбора», в поле «Тип сбора» задается значение «по умолчанию».

Редактирование справочника «Рабочий процесс сбора» разрешено, даже если для данного типа сбора уже созданы шаблоны сбора. Однако в этом случае все ранее созданные шаблоны сбора изменены не будут: необходимо отключить их и создать новые.

###### Автоматические переводы статусов

Если в справочнике «Рабочий процесс сбора» не указана роль, а в поле «Тип события» указана не кнопка, то Система изменяет статус задания автоматически, обрабатывая типы события в следующем порядке: «Дней до конца периода сбора», «Дней от статуса», «Дней от начала периода сбора».

Если задание должно перейти в статус «Сбор закрыт: новое» за 3 дня до конца периода сбора, а в статус «Новое просрочено» через 2 дня после статуса «Новое», то в случае конфликта по времени задание перейдет в статус «Сбор закрыт: новое», а не «Новое просрочено».

Проверка и автоматический перевод статусов происходит ночью.

Новое задание, сформированное днем и созданное за прошлый период (например, с использованием опции «Количество предыдущих периодов для генерации заданий» в шаблоне сбора), пройдет проверку на статус ближайшей ночью и будет переведено в статус «Закрыто».

Дни считаются включительно, т. е. если в рабочем процессе для перехода из статуса «Новое» в статус «Новое просрочено» указано, что:

* перевод через 3 дня от статуса «Новое», то 3 полных дня задание будет в статусе «Новое»: если первого числа днем запустили шаблон, то задание будет в статусе «Новое» 1-го (неполный день), 2-го, 3-го, 4-го числа. В ночь с 4 на 5 задание перейдет в статус «Новое просроченное»;
* перевод через 3 дня от начала периода (месяц): задание будет в статусе «Новое» 1-го, 2-го, 3-го числа. В ночь с 3 на 4 задание перейдет в статус «Новое просроченно»;
* перевод за 3 дня до конца периода (месяц январь): задание перейдет в статус «Новое просрочено» в ночь с 28.01 на 29.01;
* перевод до конца периода сбора «-3» (месяц январь): задание перейдет в статус «Новое просрочено» в ночь с 03.02 на 04.02.

Если администратор системы изменит статус задания сбора вручную, а по срокам «Рабочего процесс сбора» оно должно быть в статусе «Закрыто» (или ином), ночью оно будет переведено в соответствующий статус.

##### Настройка ролевой модели сбора. Справочник «Роли сбора»

Перечень ролей пользователей, которые задействованы в процессе сбора данных, содержится в справочнике «Роли сбора». В нем содержатся системные роли из справочника «Роли» (см. п. **Ошибка! Источник ссылки не найден.**), которые относятся к сбору данных. Для перехода к справочнику следует в главном меню Системы выбрать пункт «Настройки сбора», далее вкладка «Роли сбора».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5.

##### Настройка статусов сбора данных. Справочник «Статусы сбора»

Перечень возможных статусов заданий сбора данных содержится в справочнике «Статусы сбора». Для перехода к справочнику следует в главном меню Системы выбрать пункт «Настройки сбора», далее вкладка «Статусы сбора».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5.

##### Настройка доступа к конкретным заданиям. Справочник «Доступы»

Справочник «Доступы» содержит правила, определяющие связку: «Пользователь – Тематика – Тип сбора – Узел сбора – Роль».

Для перехода к справочнику следует в главном меню Системы выбрать пункт «Настройки сбора», далее вкладка «Статусы сбора».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5.

Правила, содержащиеся в справочнике, определяют привязки пользователей к заданиям сбора, т. е. описывают доступ к конкретным заданиям разных узлов сбора в зависимости от статуса заданий и тематики задания. В качестве узлов сбора должен использоваться показатель множественного выбора, который ссылается на справочник, определяющий структуру сбора (см. п. 4.1.3.2). Например, таким показателем может быть «Административная структура (множ. выбор)».

Для всех узлов сбора необходимо создавать отдельные записи в справочнике «Доступы» (например, если пользователь с ролью «Контролер» получает задания по всем узлам сбора, то в справочнике «Доступы» необходимо создать запись не только к организации (узлу сбора) верхнего уровня, но и для всех организаций (улов сбора)).

Для доступа к паспорту шаблона сбора (по ссылке) в справочнике должна быть создана связка «пользователь – тематика (относящаяся к шаблону) – любая роль – тип сбора – узел сбора, совпадающий хотя бы с одним узлом структуры сбора, указанным в шаблоне сбора в поле «Узлы» (см. п. 4.1.3.4.2).

Для редактирования, включения и удаления шаблона в справочнике должна быть создана связка: «пользователь – тематика (относящаяся к шаблону) – «нужная роль» (т. е. та роль, которая в типе сбора переводит из пустого в статус «Новое») – тип сбора – узлы сбора, совпадающие с узлами структуры сбора, указанными в шаблоне сбора в поле «Узлы» (см. п. 4.1.3.4.2) (в справочнике «Доступы» может быть больше узлов, чем в поле «Узлы» в шаблоне, но не меньше).

Примечание – Пользователю с ролью «Редактор шаблона» недоступны по умолчанию (если не настроены в справочнике «Рабочий процесс сбора») иные действия с заданиями.

##### Управление тематиками сбора данных. Справочник «Тематики»

Тематика сбора данных – это направленность сбора данных (например, «Бухгалтерия», «Безопасность»), сущность, через которую задается связь между пользователем, шаблоном сбора и узлами из дерева сбора. В Системе может быть несколько тематик сбора данных.

Для перехода к справочнику следует в главном меню Системы выбрать пункт «Шаблоны сбора», далее вкладка «Тематики».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5.

##### Настройка уведомлений. Справочник «Уведомления»

В справочнике «Уведомления» настраиваются шаблоны уведомлений (название шаблона, тема, текст), которые могут быть указаны в шаблоне сбора, в таблице «Рабочий процесс», для любого перехода из статуса в статус (см. п. 4.1.3.4.2).

Для перехода к справочнику следует в главном меню Системы выбрать пункт «Шаблоны сбора», далее вкладка «Уведомления».

Справочник доступен на редактирование пользователям-администраторам. Работа в справочнике осуществляется, как описано в п. 4.1.1.5.

#### Управление шаблонами сбора данных

Шаблон сбора обеспечивает формирование заданий сбора по выбранным узлам дерева сбора по одной тематике. В нем же задается то, какие формы, реестры и отчеты будет видеть пользователь в своем задании.

##### Просмотр списка шаблонов

Для перехода к списку шаблонов сбора следует выбрать в главном меню пункт «Шаблоны сбора». Откроется страница со списком шаблонов сбора, где для пользователя отображаются только шаблоны, связанные с доступными данному пользователю тематиками (в соответствии со справочником «Доступы», см. п. 4.1.3.3.5).

В списке шаблонов доступны фильтры по полям «Тематика», «Статус».

Для перехода к паспорту шаблона из общего списка нужно нажать на пиктограмму «Паспорт шаблона» в последнем столбце строки нужного шаблона.

##### Создание шаблонов сбора

Для создания нового шаблона сбора необходимо:

1. в главном меню нажать пиктограмму «плюс» около пункта меню «Шаблоны сбора»;
2. в открывшейся форме заполнить поля (рисунок 80):
   1. «Название» – название шаблона сбора, которое будет отображено в названии заданий, которые получат пользователи;
   2. «Тематика» – тематика определяет доступ: список пользователей, имеющих доступ к данном сбору;
   3. «Тип сбора»;
   4. в разделе «Структура сбора»:

* «Показатель параметризации» – показатель типа «Справочник», которым параметризованы формы/отчеты/реестры;
* «Показатель определения прав доступа» – справочник, по которому будет проходить сбор, например, «Административная структура»;
* «Узлы» – те узлы, которые получат задание и с которых требуется собрать данные;
  1. в разделе «Периодичность»:
* «Показатель параметризации» – показатель типа «Период», которым параметризованы рекомендованные формы/отчеты/реестры;
* выбор одного из значений:
* «Из типа периода» – выбор типа периода для периодичных заданий: день/неделя/год и т. д. Следует обратить внимание, что сбор всегда начинается с текущего периода (если «Количество предыдущих периодов для генерации заданий» = 0, см. ниже). То есть если при настройке сбора был выбран 2016 год, а сейчас 2019 год, то задание все равно будет сформировано, начиная с 2019 года. Конкретный период важен только для недели (с какого дня недели сбор: с понедельника по воскресенье или с пятницы по четверг). При этом:
* флаг «Разовое задание» – позволяет создать задания только на текущий период из выбранного типа периода (задания сбора на следующие периоды сбора автоматически формироваться не будут);
* «Количество предыдущих периодов для генерации заданий» – по умолчанию 0. Данное поле требуется использовать, если нужно запустить сбор, начиная не с текущего периода, а с какого-либо прошлого. Особенность: сроки сбора – такие, как прописано в шаблоне;
* «Фиксированные периоды» – выбор конкретных периодов, за которые будут сформированы задания. Фиксированный период может быть выбран в прошлом.
* в разделе «Рекомендованные» (формы/отчеты/реестры) – указать те сущности, на которые будут вести ссылки из паспорта задания. В паспорте они будут расположены в алфавитном порядке (в названии формы может использоваться пробел перед названием). При этом:
* формы и реестры необходимо заполнить пользователю;
* отчеты можно посмотреть и/или экспортировать. Также отчеты могут быть использованы для межформенных проверок (например, для сравнения данных, введенных в форму, с прошлым периодом);

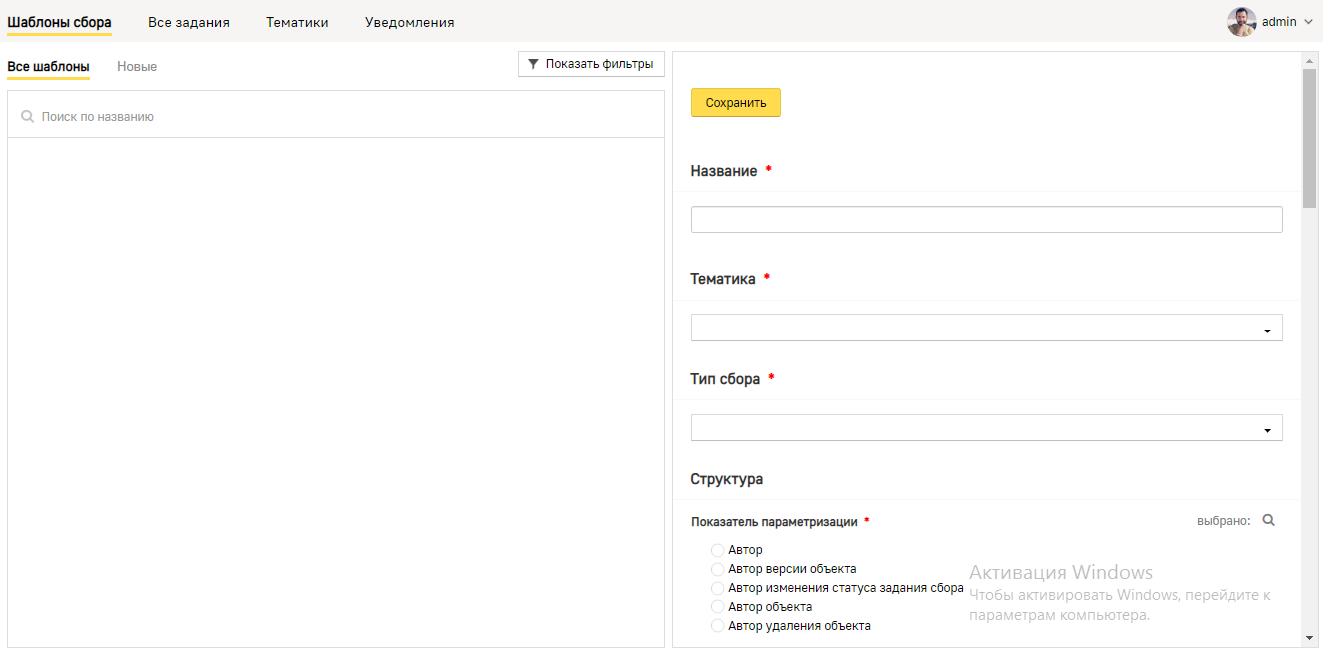


Рисунок – Создание шаблона сбора данных

* 1. нажать кнопку «Сохранить». После сохранения шаблона сбора станут доступны настройки рабочего процесса и возможность прикрепления файла с методическими указаниями по заполнению;
  2. в разделе «Рабочий процесс» настроить рабочий процесс сбора данных, выбрав данные из справочника «Рабочий процесс сбора» (см. п. 4.1.3.3.2). В таблице «Рабочий процесс» для любого перехода из статуса в статус можно указать уведомление (одинаковое для всех ролей). Настройка уведомлений включает поля: «Шаблон уведомления» и «Уведомляемые роли». Шаблон уведомления (название шаблона, тема, текст) настраивается в справочнике «Уведомления» (см. п. 4.1.3.3.7);
  3. указать межформенные проверки, которые будут выполняться для форм, указанных в данном шаблоне (рисунок 81). Для добавления межформенной проверки необходимо воспользоваться соответствующей таблицей.

Проверка может быть создана для любого перехода из статуса в статус. Обычно проверка необходима только для тех статусов, где в формы ввода вносятся изменения. Необходимо указать проверки для всех переходов, где это требуется по регламенту (например, из статуса «Новое» и из статуса «Новое просрочено» должно быть сделано две отдельные записи). Название формы/реестра/отчета указано справа от рекомендованной формы/реестра/отчета (после сохранения шаблона сбора).

Для задания какого-либо правила нужно указать, на какую ячейку оно накладывается. Ячейка задается следующим образом: сначала указывается код таблицы, затем через точку номер ячейки. Ячейки нумеруются так же, как в Microsoft Excel: строки цифрами, начиная с первой, а столбцы большими буквами латинского алфавита (A, B, C...) Например, если требуется первая ячейка первой строки в таблице с кодом «form\_1», то она запишется так:

form\_1.A1

Если требуется столбец, то он задается буквой и звездочкой, строка задается звездочкой и цифрой – номером строки:

form\_1.C\*

form\_1.\*1

Знаками равенства и неравенства можно указать необходимые правила, между ячейками можно указывать математические операции. Каждое новое правило пишется на новой строке:

form\_1.L1=form\_2.C1

form\_1.M1=form\_2.D1

form\_1.N1>=form\_3.R1+form\_2.P1+form\_6.D1+form\_1.B1

form\_1.C\*>form\_2.C\*+100

form\_1.\*1=1

Если требуется, чтобы ячейка (или столбец/строка) обязательно были заполнены, нужно написать условие с восклицательным знаком после знака сравнения вида:

f1.A3=!f1.A3

f2.B\*=!f2.B\*

Также работают условия вида:

f1.A9>=!f1.A11

где должно быть выполнено условие неравенства с обязательным требованием, что ячейки должны быть заполнены.

Допускается использование конструкции IF-THEN-ELSE (если условие 1 истинно, то выполняется условие 2, иначе условие 3). Пример:

if form1.C2 = "БАЛАНС"

then form2.A1 = form2.A1 + form2.A2 | "НЕ СХОДИТСЯ СУММА"

else form2.A2 = form2.A1 + form2.A2 | "СХОДИТСЯ"

В конструкции условия не работает сравнение с датой, написанной в проверке. Можно сравнивать только даты, содержащиеся в различных ячейках проверяемых форм и отчетов.

Оператора «И» (AND) в синтаксисе межформенных проверок нет. Для реализации оператора «И» необходимо два раза написать условие.

Пример:

form1.A\*=!form1.A\* | "Введите отсутствующие данные!"

form1.B\*=!form1.B\* | "Введите отсутствующие данные!"

В межформенных проверках нельзя присваивать значение ячейке, у пустых ячеек по умолчанию значение нуль.

При невыполнении проверки можно выводить комментарий-подсказку во всплывающем окне с ошибками. Для задания подсказки после условия в том же окне нужно написать вертикальную черту и текст в кавычках.

Если какая-то из строк содержит ошибку, неверное написание или если есть повторение среди кодов таблиц, при попытке сохранить появится сообщение с указанием ошибок.

Межформенные проверки выполняются в момент выполнения задания или при проверке. Если данные в ячейках таблиц не соответствуют межформенным проверкам, то задание не может быть выполнено;

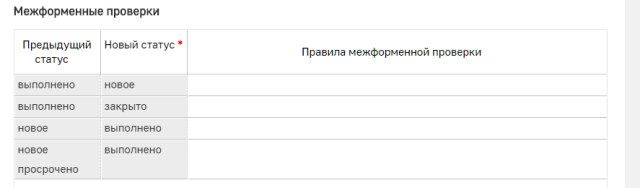


Рисунок – Межформенные проверки

* 1. если в каждом задании сбора по шаблону во всех статусах должен быть доступен для скачивания какой-либо документ (например, «Регламент сбора»), то в разделе «Методические указания» необходимо добавить эти файлы (рисунок 82). Для этого следует нажать пиктограмму «+» и выбрать файл в стандартном диалоговом окне. Для сохранения документа после прикрепления файла необходимо убедиться, что нет оранжевого треугольника, предупреждающего об ошибке. Для сохранения данных в ячейке необходимо перейти на другую ячейку;

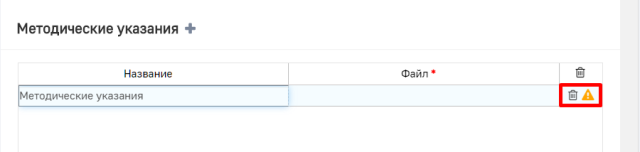


Рисунок – Добавление файлов в разделе «Методические указания»

* 1. нажать кнопку «Сохранить» для сохранения шаблона.

##### Редактирование шаблона сбора

После того, как шаблон сохранен в Системе и включен (т. е. по нему сформированы задания), действуют следующие правила редактирования:

* запрещено редактирование следующих данных в шаблоне (для их редактирования требуется отключить шаблон и создать новый):
* периодичность сбора – необходимо выключить шаблон и создать новый – с другим периодом;
* при изменении следующих данных требуется перезапуск шаблона после редактирования:
* изменение списка узлов сбора (удаление или добавление новых). После изменения узлов требуется перезапустить шаблон. Задания, которые были сформированы ранее (для удаленных узлов) останутся в Системе, но станут недоступными для редактирования (их можно просмотреть в списке «Все задания», с отметкой «Задание не активно») и в отчетах об исполнительской дисциплине. Задания для новых узлов будут сформированы, начиная с текущего периода;
* изменение поля «Значение события» в настройке рабочего процесса (как и в других таблицах в Системе, для сохранения данных в ячейке необходимо перейти на другую ячейку). После изменения поля требуется перезапустить шаблон;
* при изменении следующих данных перезапускать шаблон не требуется:
* список рекомендованных форм/отчетов/реестров (рекомендованные сущности появятся в заданиях сразу после сохранения шаблона);
* межформенные проверки.

##### Включение шаблона

После добавления новому шаблону присваивается статус «Черновик». Для того чтобы задания по шаблону начали формироваться, необходимо перевести статус в значение «Действующий». Для этого следует нажать на кнопку «Включить шаблон» в паспорте шаблона и подтвердить включение.

После включения шаблона в течение нескольких секунд (в зависимости от количества указанных узлов дерева сбора и сроков сбора) формируются задания сбора в соответствии с настройками шаблона. Посмотреть их можно, нажав на кнопку «Задания шаблона» в его паспорте.

##### Отключение шаблона

Для отключения шаблона следует нажать на кнопку «Отключить шаблон». Статус шаблона изменится с «Действующий» на «Черновик».

## Описание операций администрирования

### Настройка главного меню

#### Настройка меню системы для конкретного пользователя

Общий порядок настройки меню Системы для конкретного пользователя включает в себя следующие шаги:

1. создать группы показателей, форм, реестров, отчетов, дашбордов, доступные для предполагаемого пользователя (см. п. 4.1.1 «Справочники и работа с ними»).
2. создать роль, которой дать доступ на чтение (и запись) указанных выше групп (см. п. **Ошибка! Источник ссылки не найден.** «**Ошибка! Источник ссылки не найден.**»);
3. создать учетную запись пользователя, которому присвоить настроенную роль. При необходимости присвоить пользователю другие роли (см. п. **Ошибка! Источник ссылки не найден.** «**Ошибка! Источник ссылки не найден.**»);
4. в справочнике «Дерево меню» (см. п. 1)) настроить видимость элементов, уже имеющихся в Системе, или создать и настроить новые. Следует обратить внимание, что в дереве меню уже настроено меню для системных ролей, но оно может не подойти для конкретного пользователя;
5. проверить отображение меню для созданного пользователя, войдя в Систему под тестовой учетной записью, для которой заданы те же права доступа, что и для данного пользователя.

#### Справочник «Дерево меню»

В Системе необходимо настроить отображение и доступность пунктов главного меню в зависимости от прав пользователя с той или иной ролью. Настройки выполняются в системном справочнике «Дерево меню» (рисунок 83), который расположен в общем списке всех справочников Системы. Также доступ к справочнику можно получить, выбрав в главном меню пункт «Главное меню».

В качестве элементов справочника «Дерево меню» представлены все существующие пункты меню. Здесь, как в обычном справочнике, доступна возможность добавлять и удалять элементы. Соответственно, в главном меню можно создать ссылку на любой элемент Системы и настроить доступ пользователей к этой ссылке.

При развертывании Системы создается стандартное главное меню.

Поля «Выбранные роли», «Показывать» автоматически не заполняются, по умолчанию главному системному администратору (суперадмину) доступны все пункты меню. Главный системный администратор настраивает доступ к разделам меню в соответствии с функциональными задачами пользователей Системы.

#### Настройка элемента справочника

Для того чтобы настроить уже существующий элемент, необходимо:

1. нажать на нужное поле в строке справочника. Для настройки доступны следующие поля элемента справочника:

* «Название» – название текущего элемента меню;
* «Верхний узел меню» – узел верхнего уровня для элемента;
* «URL» – адрес страницы Системы, которую нужно отобразить в данном пункте меню;
* «Активно если (регулярное выражение)» – пункт меню становится активным (см. п. 4.2.1.4);
* «Выбранные роли» – роли, которым разрешено видеть данный пункт меню (флаги следует установить напротив ролей, которым будет доступен элемент);
* «Код пункта меню» – обозначение картинки, которая будет показана около наименования пункта меню. Картинки и коды расположены по ссылке https://fontawesome.com/v4.7.0/icons/;
* «Показывать» – разграничение прав доступа к элементу меню пользователей;
* «Порядковый номер» – номер пункта меню, влияющий на порядок отображения элементов в нем;
* «Ссылка на создание объекта?» – отметка, означающая, что данный пункт меню требуется отображать не словами в общем списке, а в виде знака «Плюс» около родительского элемента (например, «Новый отчет»);

1. после того, как элемент настроен, необходимо нажать кнопку «Сохранить» (рисунок 83).

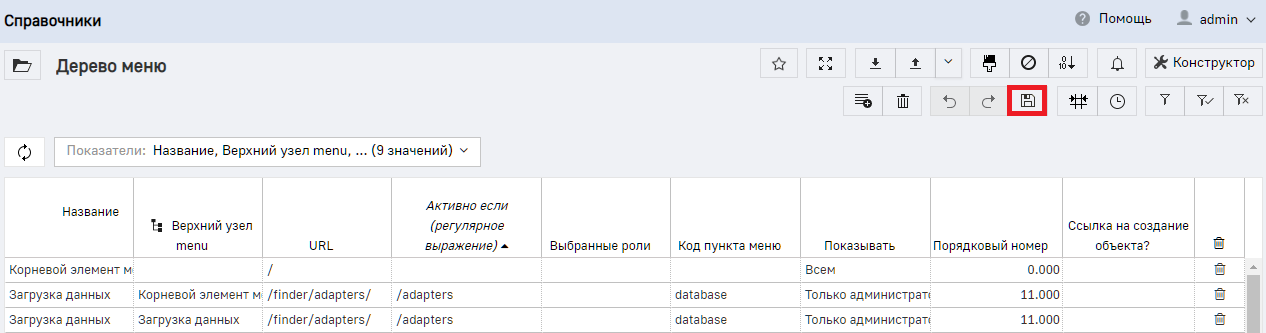


Рисунок – Справочник «Дерево меню», сохранение элемента

#### Регулярные выражения

Применение регулярных выражений позволяет пользователю понять, какой пункт меню открыт в текущий момент (рисунок 84).

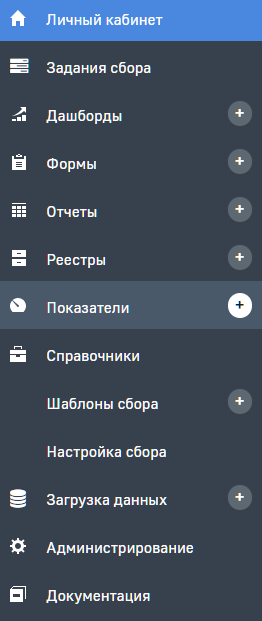


Рисунок – Пункт главного меню, который открыт в текущий момент

Общепринятые по умолчанию в Системе регулярные выражения, изначально присвоенные элементам меню:

^/(user|role|objects|logs|sysinfo|adapters).

Вертикальная черта между пунктами меню обозначает группировку объектов, при выборе которых данный элемент меню будет активен.

Если в URL объекта содержится больше одного слова:

/finder/registries/,

то при записи в окно регулярного выражения их нужно будет описать с обратным слешем (\ – символ экранирования).

Вид преображенных выражений:

^(\/finder\/registries/)

Если требуется активировать пункт меню только для объекта с определенным ID, то следует записать выражение:

(/ID)

Для активирования специально созданного объекта меню, позволяющего просмотреть последние задания сбора по организациям для контролера сбора, нужно в шаблоне сбора выяснить URL, который ведет к его последним заданиям (в шаблоне сбора нажать кнопку «Последние задания шаблона») и скопировать его в пункт URL и «Активно». В пункте «Активно» нужно будет его привести к виду:

(\/collect\/template\_tasks\/list\/-500\/N-задания/)

Если какой-либо элемент, содержащий в своем названии ключевые слова, должен отображаться только в одном пункте меню, а ключевые слова содержатся в нескольких URL (например, пользователю доступен раздел меню группы реестров и необходимо создать в меню еще один пункт, ссылающийся на другие группы реестров), то перед названием группы, которая должна отображаться отдельно, прописывается в основном пункте реестров установленный символ «?!». Он обозначает, что данную группу объектов пункт меню отображать не должен.

#### Создание нового элемента меню

Создание нового элемента рассмотрено ниже на конкретном примере. Например, пользователю нужно заполнять реестры, относящиеся к определенной группе реестров. Для этого необходимо:

1. создать новый пункт меню в корневом элементе;
2. указать название элемента, в поле URL записать номер группы реестров /registry/groups/#12345 (чтобы узнать номер группы, необходимо нажать на узел группы; номер будет отображаться в адресной строке браузера). Чтобы группа отображалась раскрытой, следует написать в виде: /registry/groups/12345/#12345;
3. задать регулярное выражение и выбрать роли, которым будет доступен новый пункт меню. Например, при заданных настройках «Роль» – «Сбор данных: участник», «Показывать» – «только выбранной роли», данный пункт меню не будет доступен никому, кроме участника сбора.

#### Особенности работы меню при отсутствии прав на доступ к элементу

При входе в Систему проверяется доступность личного кабинета по дереву меню (не учитывая корень дерева). Если страница «Личный кабинет» не доступна данному пользователю, то происходит переход на первый доступный в дереве меню узел (поиск ведется «в ширину», т. е. сначала проверяются все дочерние узлы корневого узла, потом все узлы второго уровня, потом все узлы третьего и т. д.).

#### Стандартный вариант главного меню системы

При развертывании Системы создается стандартное главное меню. Поля «Выбранные роли», «Показывать» автоматически не заполняются, по умолчанию администратору системы доступны все пункты меню. Администратор настраивает доступ к разделам меню в соответствии с функциональными задачами пользователей Системы.

### Работа с метаданными, данными, базой данных

#### Идентификационные номера объектов. Определение и использование

Каждому информационному объекту Система присваивает уникальный идентификационный номер (id объекта). Идентификационные номера объектов используются для задания формулами фильтров в заголовках строк или столбцов отчета.

Системные объекты имеют идентификаторы со знаком «минус». Например, системный показатель «Период» имеет id = -64, системный справочник «Варианты доступа» имеет id = -905.

Сведения о системных справочниках приведены в п. 4.1.1.3.

Определить идентификационный номер объекта в Системе можно следующими способами:

* перейти на страницу просмотра объекта и скопировать последние цифры из URL;
* если системным администратором включена соответствующая настройка, то в отчете/форме можно нажать правой кнопкой мыши на значение и в раскрывающемся списке выбрать «Номер объекта» (рисунок 85).

Допустим, требуется узнать идентификатор справочника «Группы отчетов». Для этого необходимо в главном меню выбрать пункт «Справочники» и выбрать нужный справочник. В строке URL на странице просмотра справочника net-dev.chtd.ru/relation/-206/, последние цифры (-206) и есть идентификатор справочника, то есть в данном примере id= -206.

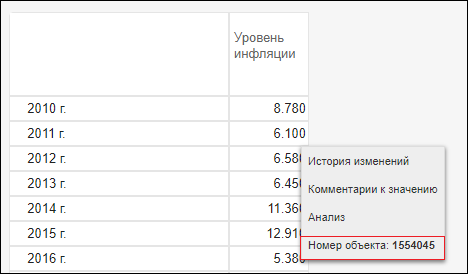


Рисунок – Идентификатор объекта

#### Граф метаданных

Граф метаданных – это визуализация объекта и его исходящих связей. Данная визуализация помогает увидеть и понять структуру связей между объектами Системы. Для того чтобы просмотреть граф метаданных определенного объекта, необходимо перейти по ссылке следующего вида:   
<url системы>/objgraph/<id объекта>/, где <url системы> – адрес главной страницы Системы, <id объекта> – идентификационный номер объекта.

#### Системный отчет «Блоки данных»

Администратору системы доступен системный отчет «Блоки данных» о текущем состоянии данных Системы. Для просмотра отчета необходимо в главном меню выбрать пункт «Администрирование» (рисунок **Ошибка! Источник ссылки не найден.**), затем на открывшейся странице выбрать вкладку «Блоки данных» либо перейти по адресу <url системы>/datablocks/, где <url системы> – адрес главной страницы Системы.

В отчет выводятся все информационные объекты, существующие в Системе в данный момент (рисунок 86), с учетом следующего:

* объекты одинаковой структуры (содержащие одинаковые показатели, одинаковые признаки и принадлежащие одному справочнику) группируются в один блок данных;
* для каждого блока данных в столбце «Всего объектов» указывается количество объектов, содержащихся в этом блоке;
* в столбце «Последний созданный объект» указывается идентификационный номер объекта, который создан последним из объектов этого блока;
* зеленая «галочка» в столбце «Системный» означает, что все показатели и признаки объектов этого блока являются системными;
* зеленая «галочка» в столбце «Аналитические признаки актуальны» означает, что в паспортах у всех показателей данного блока действительно заданы аналитические признаки, которые имеются у объектов этого блока;
* красный «крестик» в столбце «Аналитические признаки актуальны» означает, что хотя бы у одного объекта, относящегося к данному блоку, имеются аналитические признаки, не указанные в паспорте хотя бы одного показателя этого блока. В таком случае при наведении курсора на название этого показателя данного блока (в столбце «Показатели») во всплывающей подсказке указываются названия отсутствующих аналитических признаков.

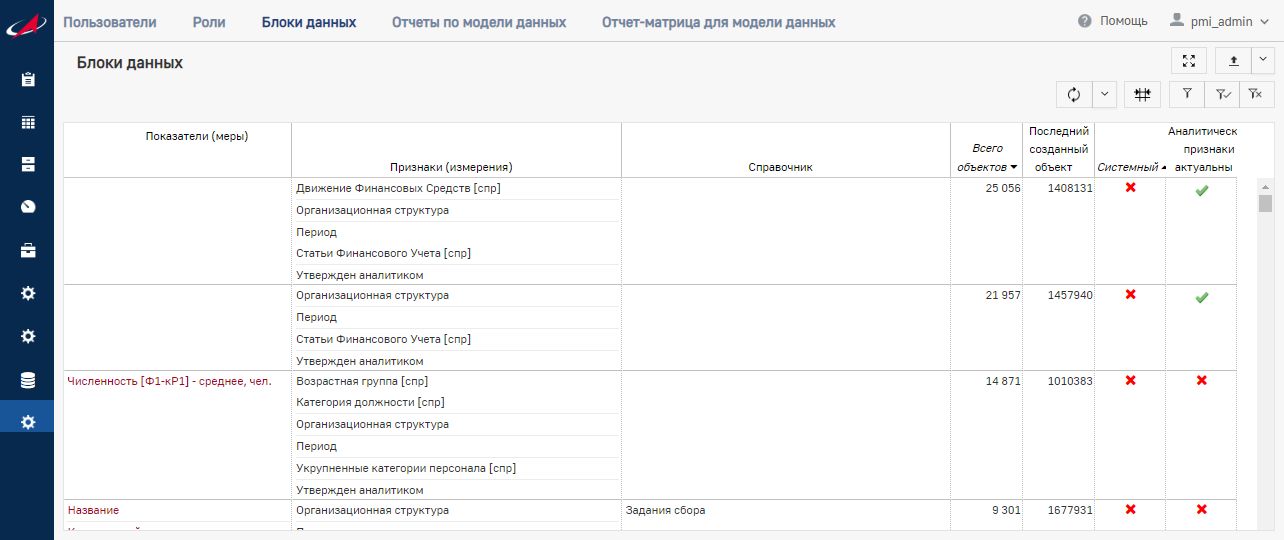


Рисунок – Отчет «Блоки данных»

#### Интерфейс исследования модели данных

Администратору системы доступен интерфейс исследования модели данных, который визуализирует взаимодействие показателей, отчетов, реестров, форм, дашбордов (формирование специальных системных отчетов по взаимосвязям основных сущностей Системы – показателей, отчетов, реестров, форм, дашбордов с точки зрения использования их метаданных). Для просмотра отчетов необходимо в главном меню выбрать пункт «Администрирование» (рисунок **Ошибка! Источник ссылки не найден.**), затем на открывшейся странице выбрать вкладку «Отчеты по модели данных».

На вкладке доступны отчеты: «Показатели», «Дашборды», «Отчеты», «Реестры», «Формы». Пример страницы отчета «Показатели» приведен на рисунке 87. Для перехода между сущностями (т. е. выбора отчета) следует нажать пиктограмму «папка» () слева от названия отчета.

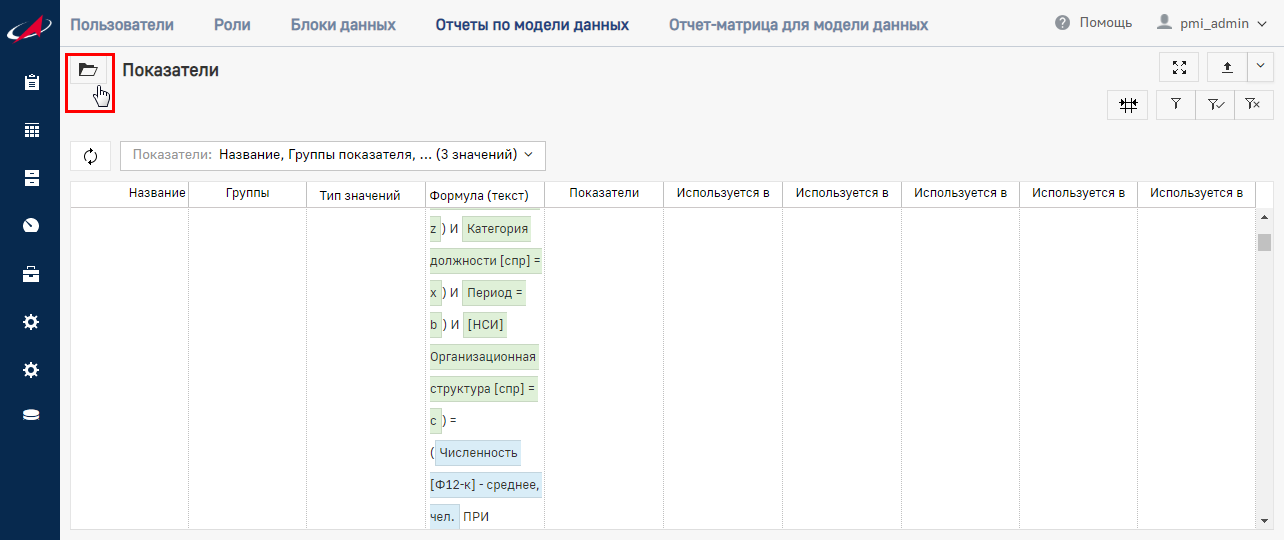


Рисунок – Отчет «Показатели» и пиктограмма выбора сущностей,   
по которым формируется отчет

Также в Системе доступен отчет-матрица для модели данных, который демонстрирует перечень показателей и их аналитических разрезов показателей деятельности организаций РКО. Для просмотра отчета следует перейти на вкладку «Отчет-матрица для модели данных». Пример отчета приведен на рисунке 88.

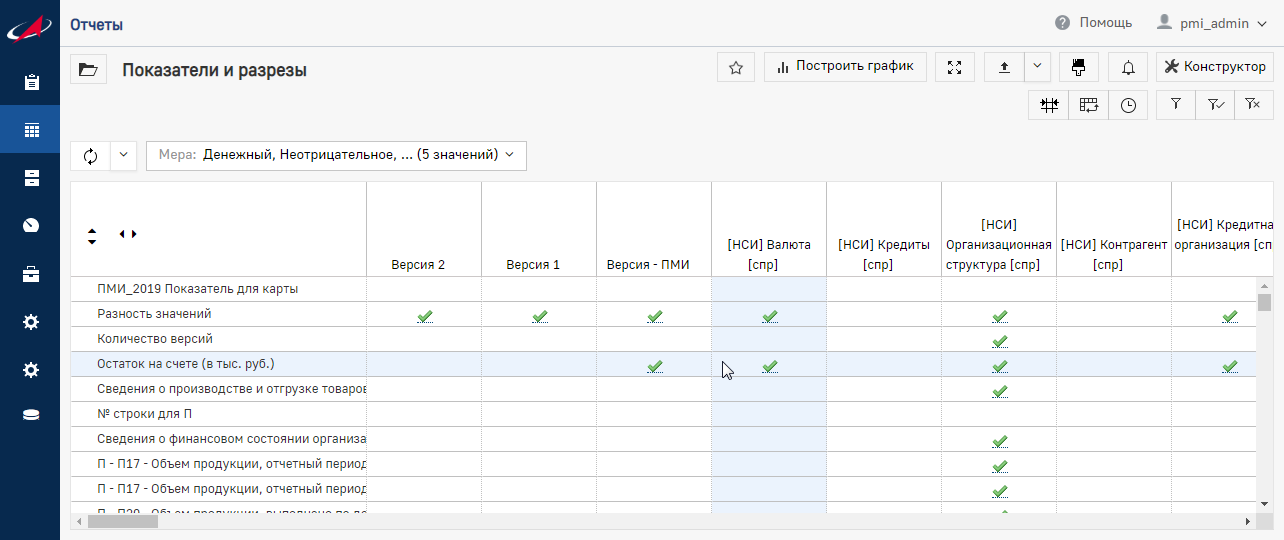


Рисунок – Отчет-матрица для модели данных

### Импорт и экспорт метаданных

#### Экспорт метаданных

Для того чтобы экспортировать данные из Системы в формат JSON, необходимо:

1. в главном меню выбрать пункт «Администрирование», далее «Импорт/экспорт»;
2. на странице (рисунок 89) отметить флагами нужные объекты;
3. нажать кнопку «Экспорт» ();
4. в диалоговом окне указать место сохранения файла и выполнить сохранение. Файл будет экспортирован в формате JSON.

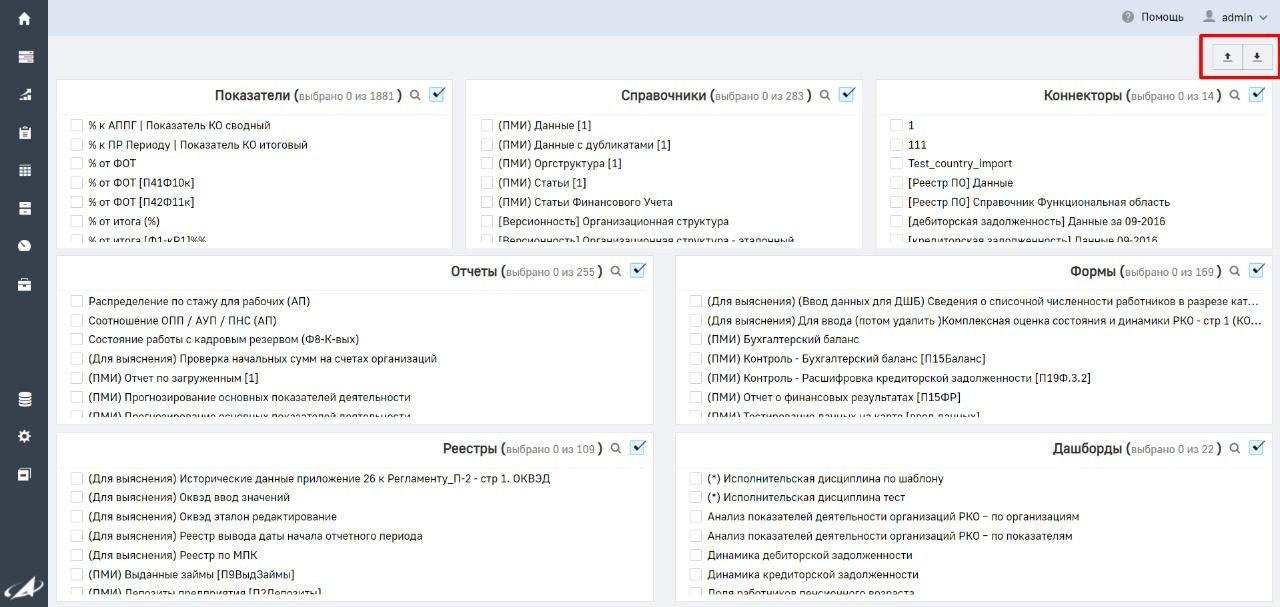


Рисунок – Страница импорта/экспорта данных и кнопки импорта/экспорта

#### Импорт метаданных

Для импорта метаданных из файла в формате JSON необходимо:

1. в главном меню выбрать пункт «Администрирование», далее «Импорт/экспорт»;
2. нажать кнопку «Импорт» ();
3. в диалоговом окне выбрать файл в формате JSON и открыть его. Метаданные из файла будут импортированы в Систему.

### Настройка стилей интерфейса системы

Администратору системы доступна возможность настройки внешнего вида Системы средствами пользовательского интерфейса. Для настройки внешнего вида приложения необходимо:

1. в правом верхнем углу страницы нажать ссылку с именем авторизованного пользователя и в раскрывшемся списке выбрать пункт «Настройки» (рисунок 90);

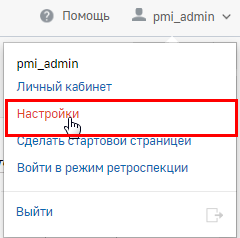


Рисунок – Переход к настройкам Системы

1. на открывшейся странице в разделе «Внешний вид приложения» нажать ссылку «Редактировать внешний вид приложения» (рисунок 91). Откроется страница редактирования внешнего вида приложения (/project/description/) (рисунок 92)

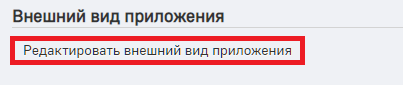


Рисунок – Переход к настройке внешнего вида приложения

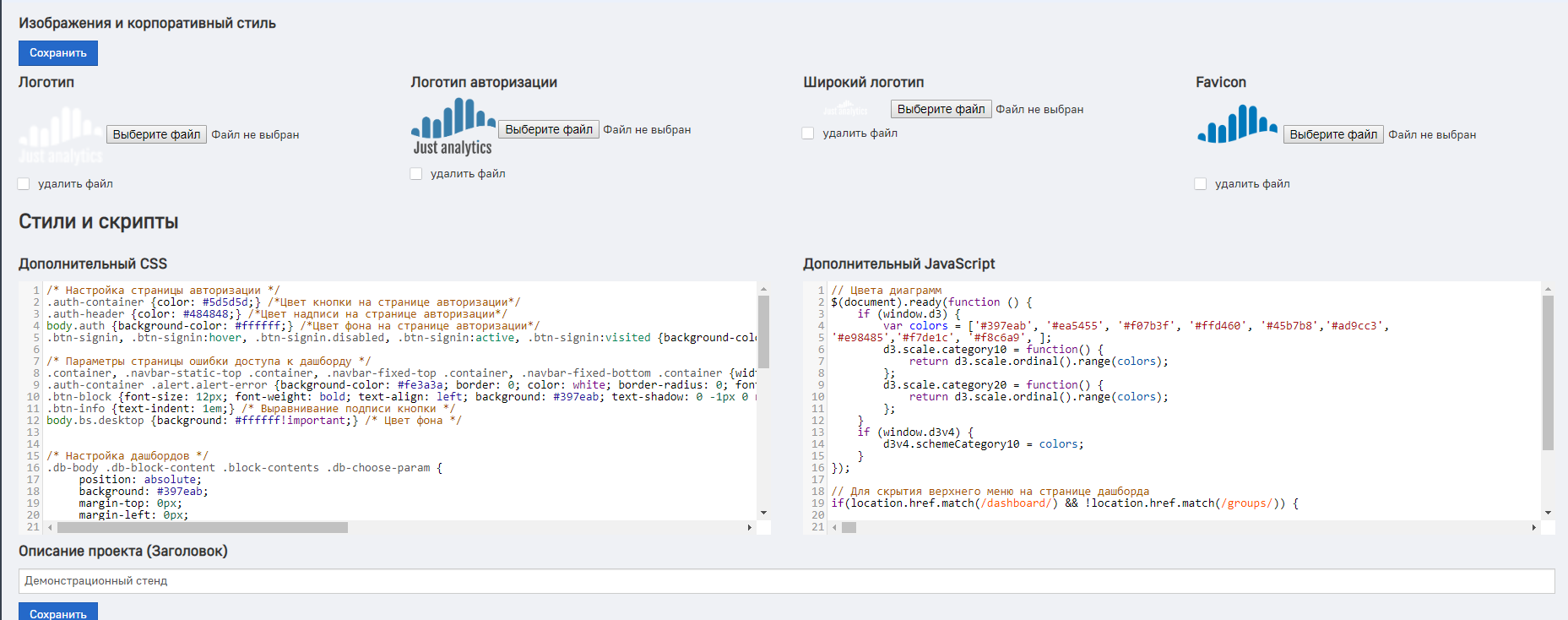


Рисунок – Страница редактирования внешнего вида приложения

1. на странице изменить необходимые настройки в полях:

* «Описание проекта (Заголовок)» – название Системы;
* «Логотип» – файл-изображение, которое будет логотипом Системы. Для выбора файла следует нажать кнопку «Выберите файл» и выбрать файл в диалоговом окне выбора файла. Рекомендуемый размер изображения: 100\*100 px;
* «Логотип авторизации» – файл-изображение, которое будет логотипом страницы авторизации; выбирается аналогично файлу логотипа;
* «Широкий логотип» – файл-изображение, которое будет логотипом Системы при открытом меню, выбирается аналогично файлу логотипа;
* «Favicon» – файл значка, отображаемого перед названием страницы во вкладке браузера, закладках; выбирается аналогично файлу логотипа;
* «Дополнительный css» – в поле возможна настройка стенда при помощи задания дополнительных стилей css. На любой странице Системы к элементу body страницы приписываются классы:

<body class="path-URL user-USER\_ID role-ROLE\_ID">...</>

Здесь:

* path-URL: URL – это текущий path, в котором символы «/» заменены на «-»;
* user-USER\_ID: USER\_ID – идентификатор текущего пользователя;
* role-ROLE\_ID: ROLE\_ID – идентификатор текущей роли.

Если необходимо изменить стили для конкретной страницы, раздела, конкретного пользователя или роли, то к этим данным возможно обратиться через соответствующие классы тега body. Любой класс (вида .className {}; ), который описывается в project/desciption, при необходимости можно сделать видимым только на определенной странице (или при авторизации под определенным пользователем). Для этого перед стилем пишется название класса (через пробел), которое будет определять где (для кого) необходимо применять эти стили:

.path-URL .className {

//стили пользователя

}

Примеры:

Размер и отступы заголовка стенда, верхнего меню:

#branding h1 a { height: 5px; padding-top: 10px; padding-left: 100px; font-size: 40px; } #top-menu { padding-left: 110px; }

На странице дашбордов необходимо скрыть шапку (класс .header\_wrap):

.path-dashboard .header\_wrap {display: none;}

То же самое, но конкретно для администратора:

.user-85 .header\_wrap {display:none;}

Таким образом, перед каждым стилем, который необходимо изменить, существует возможность указать тот объект, для которого это необходимо изменить: конкретный отчет, пользователь и т. д.

* «Дополнительный JavaScript» – возможность настроить Систему, используя собственные js-функции и методы, которые будут вставлены в тег <script> страницы, например, вставка текста на странице, задание анимации, изменение структуры HTML и т. д.

### Точки восстановления

В Системе можно вернуться к состоянию данных и метаданных на заданный момент времени. Для того чтобы вернуться к определенному моменту времени, он должен быть заранее зафиксирован, т. е. должна быть создана точка восстановления. При откате на точку восстановления происходят необратимые изменения – удаляются все изменения, произошедшие в Системе после создания точки восстановления.

#### Автоматическое создание точек восстановления

Перед началом работы в Системе создается начальная точка восстановления. Откат на эту точку означает полное очищение базы данных.

При каждой загрузке данных с помощью коннектора создается точка восстановления, позволяющая вернуться к состоянию данных и метаданных до этой загрузки. В названии точки восстановления будет отражено, перед загрузкой данных из какого источника она создана, а также время и дата создания точки.

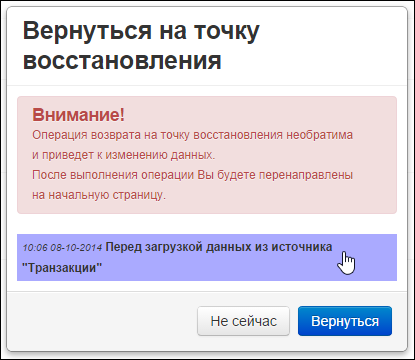


Рисунок – Сведения об автоматически созданной точке загрузки восстановления

#### Создание точки восстановления вручную

Для создания точки восстановления вручную необходимо:

1. в правом верхнем углу страницы нажать ссылку с именем авторизованного пользователя и в раскрывшемся списке выбрать пункт «Настройки» (рисунок 90);
2. на открывшейся странице в разделе «Восстановление» нажать на кнопку «Создать точку восстановления» (рисунок 94);

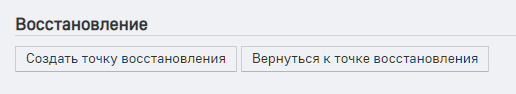


Рисунок – Кнопки выполнения операций с точками восстановления

1. в открывшемся окне:

ввести комментарий к точке восстановления (рисунок 95). Комментарий позволит в дальнейшем понять, к какому состоянию данных и метаданных будет совершен откат;

нажать кнопку «Создать». Появится сообщение об успешном создании точки восстановления.

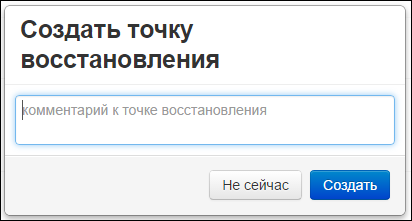


Рисунок – Создание точки восстановления

#### Откат на точку восстановления

Для того чтобы вернуться к состоянию данных на момент создания точки восстановления, необходимо:

1. в правом верхнем углу страницы нажать ссылку с именем авторизованного пользователя и в раскрывшемся списке выбрать пункт «Настройки» (рисунок 90);
2. на открывшейся странице в разделе «Восстановление» нажать на кнопку «Вернуться к точке восстановления» (рисунок 94);
   1. в открывшемся окне, где перечислены все точки восстановления (рисунок 96):

нажать на строку с нужной точкой восстановления;

при необходимости установить флаг «С сохранением истории» – при установке флага Система обеспечивает возможность перемещения по временным срезам данных Системы, с сохранением истории изменений, сделанных после точки среза (работу с данными Системы в разрезе временной оси);

нажать на кнопку «Вернуться».

ВАЖНО: Возврат на точку восстановления необратим!

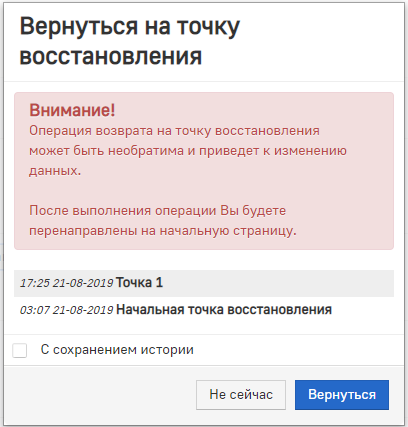


Рисунок – Выбор точки восстановления

#### Полная очистка базы данных

Для того чтобы полностью очистить базу данных, необходимо выполнить откат до точки восстановления «Начальная точка восстановления». Эта точка создается автоматически перед началом работы в Системе и откат к ней будет означать полное удаление всех данных и метаданных.

После отката данных на начальную точку восстановления откроется начальный экран Системы.

#### Резервное копирование и восстановление данных (подробности в Инструкции по резервному копированию)

### Формирование резервной копии

Выполняется по расписанию и в режиме, указанным в Таблице №1.

|  |  |
| --- | --- |
| **Расписание резервного копирования** | |
| Тип резервного копирования | Полный бэкап (дамп базы + архив каталога \_media) |
| Периодичность | 1 раз в день |
| День недели | Пн, Вт, Ср, Чт, Пт, Сб, Вс |
| Интервал времени | 01:00 – 02:00 |
| Время хранения копий | 1 неделя (7 дней) |
| Необходимость остановки системы | Нет |
| Примечания | Каталог файлов netdbapp\_db-YYYY-MM-DD.zip |

Каталог файлов: /home/netdbuser/backup/dumps/

**Восстановление системы из резервной копии**

**Порядок запуска серверов**

1. Выполнить старт (рестарт) сервера
2. После завершения загрузки операционной системы убедиться в нормальном запуске всех необходимых для приложения сервисов:

* веб-сервера Apache;
* сервера баз данных Postgres PRO;
* сервера сообщений RabbitMQ;
* сервера резидентной системы баз данных Redis;
* сервера запуска приложений supervisord.

выполнив в командной строке команду проверки статуса сервиса:

*systemctl status имя\_сервиса*

Для проверки статуса сервисов приложения необходимо выполнить:

*supervisorctl status*

Необходимые сервисы должны быть в состоянии RUNNING.

1. В том случае, когда в результате каких-либо сбоев приложение не может работать с базой данных, необходимо выполнить процедуру восстановления из дампа, описание которой  
   приведено ниже.

4. В каталоге хранения дампов баз данных (/home/netdbuser/backup/dumps/)  
найти последний по времени файл дампа. Файлы дампов формата 'zip' содержат в своем составе  
дамп базы данных и загруженные пользователями медиа-файлы.

5. От пользователя netdbuser выполнить команду:

*$ netdb\_load\_dump путь\_к\_последнему\_по\_времени\_файлу*

6. Выполнить команду netdb\_restart пользователем netdbuser в bash.

**Проверка работоспособности системы.**

Открыть в браузере ссылку msk1-surivk1.interrao.ru Увидеть стартовую страницу и интерфейс системы.

# Перенос настроек и тестовых данных

## Описание переноса настроек и тестовых данных

Выполнение переноса собственных настроек осуществляется выгрузкой настроек и данных во внешний файл из интерфейса системы-источника, либо из командной строки сервера приложения системы-источника и импортом полученного файла в систему-приемник.

Перенос собственных настроек с тестового стенда на продуктивный стенд и с продуктивного стенда на тестовый стенд осуществляется при помощи модуля экспорта/импорта метаданных.

Администратор модуля СУРиВК (АМК) на стенде-источнике переходит к странице модуля экспорта-импорта метаданных (/modules/metadata-import-export/), где выбирает сущности с реализованными настройками (показатели, справочники, коннекторы, реестры, формы, отчеты, дашборды или шаблоны сбора) и экспортирует их в файл метаданных, нажимая на кнопку «экспорт метаданных». Система формирует файл метаданных автоматически и предлагает сохранить его на АРМ пользователя.

Сформированный файл метаданных может быть импортирован на стенд-получатель. Для этого Администратор модуля СУРиВК / АМК на стенде-получателе переходит к странице модуля экспорта-импорта метаданных и импортирует файл метаданных, нажимая на кнопку «импорт метаданных».

Предусмотрено обновление тестовых данных в зоне настройки, зоне тестирования и контроля качества.

Перенос данных осуществляется администратором Системы. Из командной строки сервера приложения системы-источника формируется файл дампа БД с данными, и затем полученный файл импортируется в систему-приемник. Инструкция по формированию и импорту файла дампа БД с данными приведена в инструкции администратора Системы.

Применение обновлений прикладного ПО, поставляемых производителем, осуществляется путем копирования файлов (дистрибутивов, конфигураций и т.д.) с рабочего места администратора Системы в зону разработки и выполнения команд по обновлению системы. После завершения адаптации собственных настроек и разработок к изменениям производителя ПО и выполнения контроля качества, осуществляется применение обновлений прикладного ПО, поставляемых производителем, последовательно в зоне тестирования и контроля качества и в зоне постоянной эксплуатации путем копирования файлов с рабочего места администратора.

Создание и изменение функциональности Системы (преимущественно отчетов) осуществляется встроенными средствами ПО без изменения программного кода ПО Триафлай.

Для передачи файлов с разработками и данных между зонами организован единый транспортный каталог.

Дополнительных сетевых взаимодействий, относительно перечисленных в других разделах настоящего документа для выполнения перечисленных процедур не требуется.

# Аварийные ситуации

## Действия в случае сбоев в работе программного обеспечения

В случае возникновения аварийных ситуаций, связанных с программным обеспечением Системы, необходимо действовать в рамках внутренних регламентов предприятия. При неполадках, вызванных неисправностью технических средств или кабельных систем, необходимо известить технический персонал, обслуживающий сервер или рабочее место пользователя.