

О проекте

Как правило, платформа «1С:Предприятие 8» является не единственной системой в комплексах систем автоматизации деятельности разного рода организаций, которые могут включать в себя платформы электронной коммерции, хранилища данных, web-порталы, АСУ и прочие. Обеспечение взаимодействия между разными системами часто является нетривиальной задачей, требующей для своего решения владением кроссплатформенными навыками и знанием механизмов и протоколов интеграции. Система Snotone реализует один из возможных способов интеграции 1С с внешними системами, построенными на реляционных СУБД, обеспечивая между ними и 1С онлайн синхронизацию данных.

Система обеспечивает решение следующих задач:

- выгрузка (по расписанию, на основе 1С событий, вручную) произвольных данных из 1С во внешнюю СУБД (MySQL, PostgreSQL, Oracle - в разработке);
- вызов процедур 1С из внешних приложений.

Ключевые характеристики системы:

- простота и гибкость настройки выгрузки данных;
- развитый инструментарий управления выгрузкой;
- надежность и прозрачность работы системы.

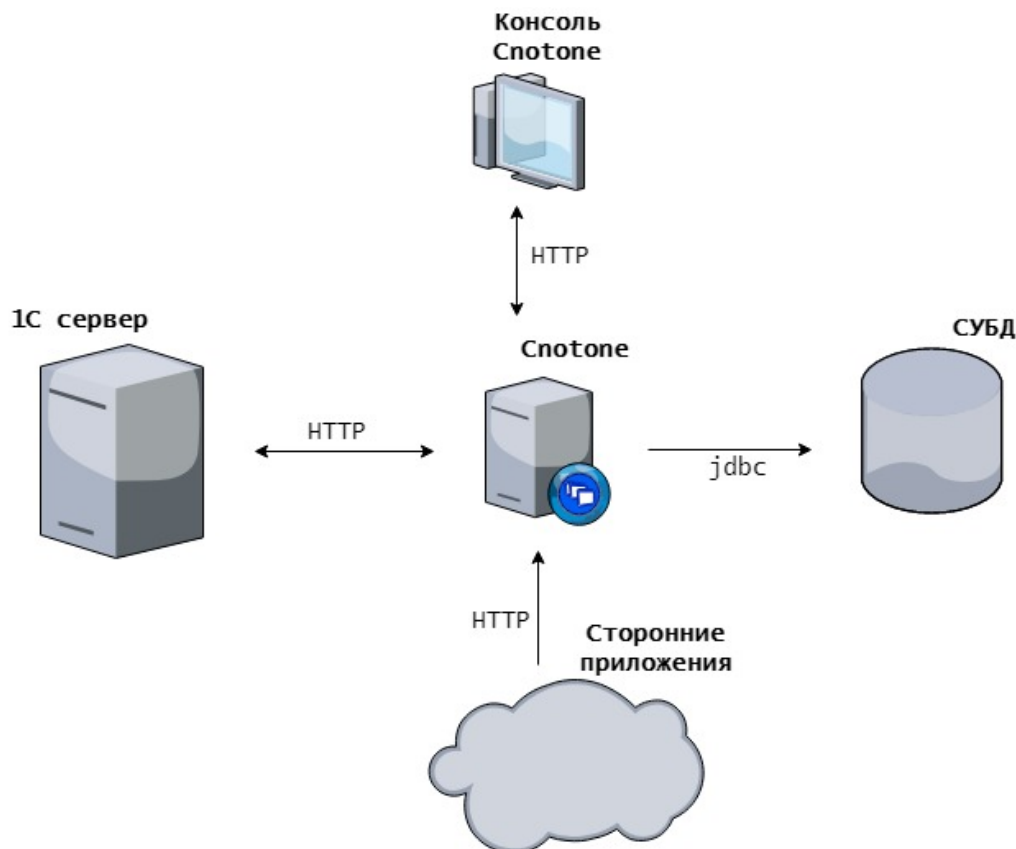
В обзорном ролике по адресу <https://youtu.be/a2Z2GGr-AkA> показаны базовые возможности системы - настройка выгрузки документа "РеализацииТоваровИУслуг" с табличной частью "Товары" во внешнюю СУБД при проведении в 1С.

Архитектура

Система включает в себя следующие компоненты:

- Серверное java приложение (cnotone-runtime), обслуживающее запросы как со стороны 1С, так и сторонних приложений.
- Одностраничное веб-приложение – консоль управления выгрузкой и загрузкой данных из/в 1С (cnotone-консоль).
- Конфигурация 1С, включающая в себя два справочника, модуль (cnotone-модуль) и http-сервис.

Взаимодействие между всеми компонентами системы осуществляется по http-протоколу посредством POST/GET запросов.



Упрощенно, сценарии работы системы.

1. Администратор системы в интерфейсе cnotone-консоли вводит настройки выгрузки данных (метаданные обмена), включает и выключает выгрузку, просматривает журналы работы системы, выполняет разные административные процедуры.
2. При работе в 1С, на основе введенных настроек в п.1, cnotone-модуль выполняет отправку данных в cnotone-runtime. Отправка инициируется либо на основе механизма подписок на события, либо вручную вызовом процедур cnotone-модуля. Данные передаются в формате json. Состав отправляемых реквизитов, полнота ссылочных данных, вычисляемые, системные и дополнительные реквизиты и прочие параметры выгрузки определяются в метаданных, настраиваемых в cnotone-консоли (п.1).
3. Cnotone-runtime принимает запросы из 1С и на основе настроенных метаданных определяет в какие таблицы/столбцы базы данных (mysql/postgresql/oracle) необходимо сохранять полученные объекты 1С, выполняет сохранение посредством jdbc, записывает информацию о событии в журнал, производит вызов дополнительных скриптов или хранимых процедур СУБД.

4. Сторонние приложения отправляют в snotone-runtime POST-запросы с параметрами вызова функций 1С в формате json. Snotone осуществляет журналирование вызова и транслирует, синхронно или асинхронно на основе механизма очереди, вызов в 1С. Snotone-модуль преобразует полученное json сообщение в структуру 1С и выполняет вызов указанной в параметрах функции 1С.

Установка и запуск

Предустановленное ПО и начальные требования

1. Java. Cnotone работает под jvm версии 8 и выше, проверить установлена ли Java на вашем сервере можно командой `java -version` в командной строке, если не установлена, ее необходимо установить - Java ^[1].
2. 1C. Для работы системы необходима 1C - серверная версия 8.3 или выше, с установленным модулем расширения веб-сервера, с поддержкой http-сервисов.
3. Apache http server. Http-сервисы 1C функционируют на основе модуля расширения веб-сервера 1C через веб-сервер Apache ^[2] либо Microsoft IIS ^[3]. О http-сервисах 1C и установке и настройке Apache и публикации http-сервисов 1C изложено, например, здесь: HTTP-сервисы для тех, кто ничего не понимает в WEB ^[4] и здесь - Настройка веб сервера Apache + 1C (Пошаговое руководство) ^[5]

Настройка конфигурации 1C

1. Выполнить объединение вашей конфигурации 1C с поставленной конфигурацией (файл - cnotone_1Cv8.cf) с помощью стандартного инструментария 1C, подробнее см. Сравнение и объединение конфигураций, механизм. ^[6]
2. Установить и настроить Apache (или IIS), выполнить в 1C публикацию http-сервиса Cnotone. После запуска веб-сервера и публикации сервиса проверить корректность его установки можно запросом в браузере: `http://{apache-host}:{apache-port}/{1c-context-root}/hs/cnotone/getSystemInfo`.

При успешной установке в ответе на запрос будет получен json с информацией о версии и конфигурации 1C такого вида:

```
{
  "success": true,
  "message": "OK",
  "body": "Конфигурация: Управление торговлей, редакция 11.2, версия 11.2.2.94; Платформа: 8.3.13.1644"
}
```

Установка Cnotone

Структура папок

Необходимо создать папки на жестком диске:

- `C:\cnotone\` - корневая папка, может быть любой;
- `C:\cnotone\config` - файлы конфигурации ;
- `C:\cnotone\logs` - логи;
- `C:\cnotone\db` - папка для внутренней БД.

В корень кладутся основной исполняемый файл - `Cnotone.jar`, утилита для установки сервиса - `service.bat` и `nssm.exe` (Non-Sucking Service Manager ^[7]), в папку `config` — `cnotone-config.properties` и `logback-cnotone.xml`.

Конфигурационные параметры

Необходимо ввести следующие параметры в файле *cnotone-config.properties*:

- cnotone.host;
- cnotone.port;
- jetty.context.path;
- hsql.db.location;
- connection.lc.http.url;
- connection.lc.admin.login;
- connection.lc.admin.password;
- connection.db.engine;
- connection.db.driverClass;
- connection.db.jdbcUrl;
- connection.db.name;
- connection.db.user;
- connection.db.password.

Подробнее о том, как их заполнять см. Конфигурационные параметры, остальные параметры можно оставить по умолчанию. Также, в параметрах журналирования при необходимости нужно изменить пути к логам – см.Журналирование, если установка выполнена в папку по умолчанию (*c:\cnotone*) – это делать не обязательно.

Установка сервиса

Cnotone может запускаться в виде отдельного процесса из командной строки:

```
java.exe -server -XX:+UseG1GC -Xms1G -Xmx3G  
-Dlogback.configurationFile="C:\cnotone\config\logback-cnotone.xml"  
-DcnotoneConfig="C:\cnotone\config\cnotone-config.properties" -jar Cnotone.jar
```

но в ОС Windows предпочтительней создать отдельный сервис (службу). Сервис можно установить с помощью специальной утилиты *nssm.exe* (включена в поставку), необходимо отредактировать файл *service.bat*, изменив в нем пути к java jre, конфигурационным файлам *cnotoneConfig* и *logback.configurationFile*, а также при необходимости установить параметры использования памяти *jvm -Xms* и *-Xmx*. После внесения изменений запустить *service.bat* файл под учетной записью администратора. Выполнить запуск сервиса. При успешном запуске в основном лог-файле *cnotone-main.log* должны появиться строки вида:

```
INFO main r.c.s.CnotoneContextListener - Cnotone контекст успешно инициализирован  
INFO main o.eclipse.jetty.server.Server - Started @34686ms
```

Проверка соединений в Cnotone

После успешной установки и запуска Cnotone открыть в браузере консоль управления, введя url *http:// {cnotone.host}:{cnotone.port}/{jetty.context.path}* (по умолчанию *http://localhost:8080/cnotone*), дождаться загрузки приложения. После загрузки выбором соответствующих пунктов в главном меню по очереди проверить соединения:

- Проверить соединение Cnotone -> 1C
- Проверить соединение 1C -> Cnotone
- Проверить соединение Cnotone -> БД

Если при проверках не было сообщений об ошибках, значит система готова к работе.

Примечания

- [1] <https://java.com/ru/download/>
 - [2] https://ru.wikipedia.org/wiki/Apache_HTTP_Server
 - [3] https://ru.wikipedia.org/wiki/Internet_Information_Services
 - [4] <https://infostart.ru/public/516810/>
 - [5] <https://infostart.ru/public/646384/>
 - [6] http://v8.1c.ru/overview/Term_000000291.htm
 - [7] <https://nssm.cc/>
-

Консоль управления

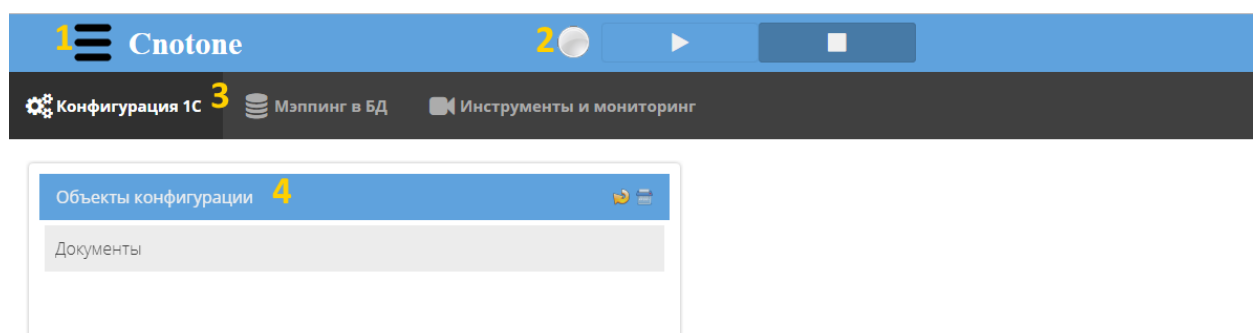
Первый запуск

Cnotone-консоль - это одностраничное веб-приложение, доступное через браузер, URL развертывания приложения задается в конфигурационном файле *cnotone-config.properties*, например параметры

- cnotone.host = 127.0.0.1
- cnotone.port = 8080
- jetty.context.path = /cnotone

определяют URL - `http://127.0.0.1:8080/cnotone`.

При первом запуске приложения cnotone-консоль имеет следующий вид:

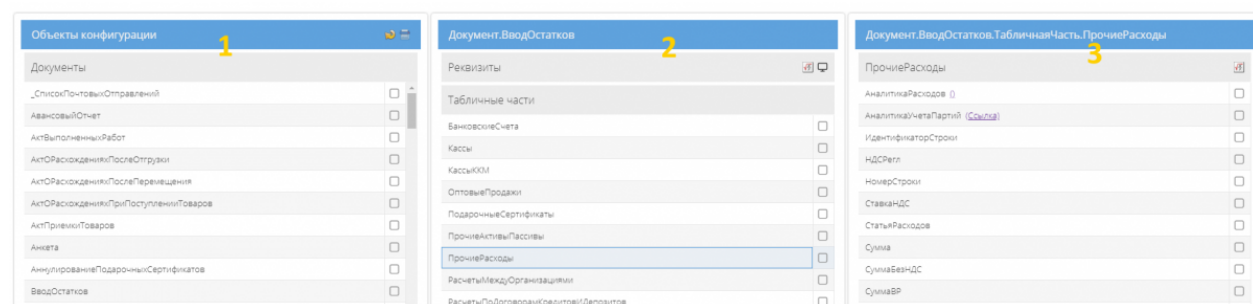


1. Главное меню
2. Статусный индикатор и кнопки запуска/остановки выгрузки. Индикатор меняет цвет в зависимости от статуса работы системы: зеленый - выгрузка активна, серый - выгрузка остановлена, красный — выгрузка остановлена из-за ошибки, - при наведении курсора на индикатор выводится информация об ошибке.
3. Панель основных закладок
 1. Конфигурация 1С. Здесь задаются объекты для выгрузки из 1С.
 2. Мэппинг БД. Здесь определяется в какие объекты БД будут выгружаться данные.
 3. Инструменты и мониторинг. Просмотр журналов выгрузки и загрузки, инструменты пакетной загрузки данных, инструменты контроля корректности выгрузки.
4. Рабочая панель

При первом запуске системы необходимо загрузить метаданные конфигурации из 1С — *Главное меню* -> *Загрузка (обновление) конфигурации 1С*, в зависимости от сложности конфигурации и производительности системы загрузка может выполняться несколько минут (в среднем 3-5). После успешной загрузки страница обновляется и можно приступить к настройке выгрузки на закладке *Конфигурация 1С*.

Панель «Конфигурация 1С». Выбор объектов выгрузки

Панель *Конфигурация 1С* разделена на 3 рабочих области:



1. Объекты конфигурации. Список всех объектов конфигурации 1С доступных для выгрузки. В текущей версии система Cnotone поддерживает выгрузку **Документов, Справочников, Регистров сведений и Регистров Накопления**.
2. Реквизиты и табличные части объекта конфигурации. Рабочая область открывается (обновляется) при выборе (клике) на объект конфигурации.
3. Реквизиты табличной части объекта конфигурации. Рабочая область открывается (обновляется) при выборе (клике) на табличную часть из предущего списка.

Настройка выгрузки объекта

Для включения объекта в выгрузку необходимо выбрать объект на панели, проставить галочку в чекбоксе, появится модально окно для выбора типов событий, при которых необходимо выполнять сохранения объекта. Список включает в себя стандартный набор событий, на которые возможна подписка в 1С:

- ПередЗаписью
- ПриЗаписи
- ПередУдалением
- ОбработкаПроведения
- ОбработкаУдаленияПроведения

Плюс, дополнительно два чекбокса:

- Вручную - сохранение объекта выполняется не на основе механизма подписок, а из кода 1С посредством вызова соответствующей функции модуля Cnotone (*Cnotone.СохранениеВручнуюОбъекта*, *Cnotone.СохранениеВручнуюРегистра*);
- Асинхронное сохранение – выгрузка экземпляров объектов будет выполняться асинхронно: cnotone-runtime сразу возвращает ответ на http-запрос, сохранение данных в БД происходит в фоновом режиме.

После выбора объекта конфигурации на второй панели необходимо выбрать реквизиты и табличные части, которые будут выгружаться. Проставить галочки. Аналогичным образом на третьей панели выбрать реквизиты табличных частей для выгрузки.

Ссылочные реквизиты

Реквизиты объекта 1С, имеющие простой тип (строка, число, дата и пр.) при сохранении в БД будут автоматически приведены к соответствующему типу столбца в БД, но часть реквизитов объекта могут иметь тип ссылки на другой объект (**Документ** или **Справочник**), в этом случае необходимо определить состав реквизитов для выгрузки этого ссылочного типа. Для этого нужно нажать на гиперссылку рядом с наименованием реквизита (по умолчанию она имеет вид (*Ссылка*), после выбора реквизитов – их наименования через запятую), откроется модальное окно:

Выбор реквизитов ссылки
✕

Реквизит: "Контрагент"

☐ По ссылке
 ☐ Наименование типа
 ☒ По значению

Справочник.Контрагенты	
ДополнительнаяИнформация	<input type="checkbox"/>
ИНН	<input checked="" type="checkbox"/>
ИмяПредопределенныхДанных	<input type="checkbox"/>
КПП	<input checked="" type="checkbox"/>
КодПоОКПО	<input checked="" type="checkbox"/>
НДСпоСтавкам4и2	<input type="checkbox"/>
Наименование	<input checked="" type="checkbox"/>
НаименованиеПолное	<input type="checkbox"/>
ОбособленноеПодразделение	<input type="checkbox"/>

Записать
Отмена

Варианты выбора:

- По ссылке – сохраняется уникальный идентификатор объекта (GUID());
- Наименование типа – сохраняется полное наименование типа, например **Документ.РеализацияТоваровУслуг** (опция может быть полезна при сохранении составных реквизитов);
- По значению – выбирается список реквизитов ссылочного типа, которые будут сохраняться в отдельные столбцы БД в ту же таблицу, что и простые реквизиты основного объекта.

Вычисляемые реквизиты

Часто возникает ситуация, когда вместе с основными реквизитами объекта необходимо выгружать дополнительные данные по объекту, которые можно получить программным путем (в коде модуля 1С) на основании данных объекта. В системе Snotone такая возможность реализована через механизм *вычисляемых реквизитов*. Для создания реквизита такого типа на панели инструментов нужно кликнуть по иконке *Добавить вычисляемый реквизит*:

Реквизиты
✓
🖨

ОснованиеДата
✕

ОснованиеНомер
✕

Добавить вычисляемый реквизит

Откроется модальное окно, в котором нужно ввести наименование реквизита, тип, а также формулу – наименование общего модуля и функции 1С, которая будет производить вычисление.

Создание вычисляемого реквизита
✕

Имя:

ПечатнаяФормаСчета

Тип данных:

Строка
▼

Формула:

ВыгрузкаДанных.ФормированиеПечатнойФормы

Записать

Отмена

Для примера, мы хотим выгружать печатную форму документа **СчетаНаОплатуКлиенту**, которую в дальнейшем внешняя система (интернет-магазин) сможет отправлять по почте клиенту или предоставлять для скачивания в личном кабинете покупателя. В spotone-консоли создается вычисляемый реквизит **ПечатнаяФормаСчета**, а в общий модуль 1С **ВыгрузкаДанных** добавляется функция **ФормированиеПечатнойФормы**, которая имеет один аргумент – экземпляр объекта выгружаемого типа:

Функция ФормированиеПечатнойФормы (Источник) Экспорт

```

ОбработкаСсылка                                     =
Справочники.ДополнительныеОтчетыИОбработки.НайтиПоРеквизиту("ИмяОбъекта","СчетНаОплатуПокупателю");
ИмяФайла = ПолучитьИмяВременногоФайла();
ДвоичныеДанные = ОбработкаСсылка.ХранилищеОбработки.Получить();
ДвоичныеДанные.Записать(ИмяФайла);
ОписаниеЗащиты = Новый ОписаниеЗащитыОтОпасныхДействий;
ОписаниеЗащиты.ПредупреждатьОбОпасныхДействиях = Ложь;
ВнешняяОбработка = ВнешниеОбработки.Создать(ИмяФайла, Ложь, ОписаниеЗащиты);
МассивОбъектов = Новый Массив;
МассивОбъектов.Добавить(Источник.Ссылка);
ОбъектыПечати = Новый СписокЗначений;
ТабличныйДокумент = ВнешняяОбработка.Печать(МассивОбъектов, Неопределено,
ОбъектыПечати, Неопределено);
СоздатьКаталог("C:\1documents\\"+Формат(Источник.Дата,"ДФ=уууу")+"\bill");
ТабличныйДокумент.Записать("C:\1documents\\"+Формат(Источник.Дата,"ДФ=уууу")+"\bill\bill_"
+ Формат(Источник.Дата,"ДФ=ууууMMddHHmmss") + "_" + Источник.Номер + ".xls",
ТипФайлаТабличногоДокумента.XLS);
РезультатXls = Base64Строка(Новый
ДвоичныеДанные("C:\1documents\\"+Формат(Источник.Дата,"ДФ=уууу")+"\bill\bill_"
+ Формат(Источник.Дата,"ДФ=ууууMMddHHmmss") + "_" + Источник.Номер + ".xls"));
Возврат РезультатXls;

```

КонецФункции

Системные реквизиты

Иконка рядом с иконкой **вычисляемого реквизита**. При клике появляется модальное окно с возможностью выбора системных реквизитов, которые будут также сохраняться в таблицу БД вместе с остальными реквизитами объекта:

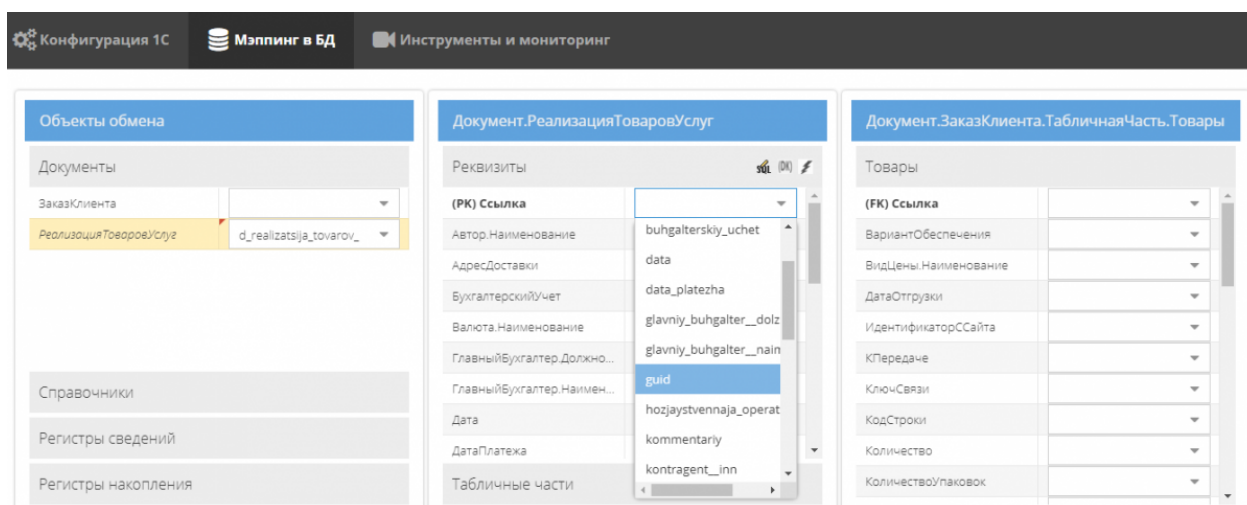
- **Дата и время изменения** - дата и время события, при котором выполнялось сохранение объекта;
- **ИдентификаторБД1С** - константа, идентификатор базы данных 1С (результат вызова *СтрокаСоединенияИнформационнойБазы()*);
- **ТекущийПользователь** - пользователь 1С, выполнявший действие (запись/проведение и пр.) с объектом;

Панель «Конфигурация 1С». Настройка мэппинга в БД

Основные принципы сохранения данных из 1С в базу данных (мэппинг):

- Экземпляр объекта 1С (документ, справочник, строка регистра) сохраняется в таблицу БД;
- Реквизит объекта сохраняется в столбец соответствующей таблицы;
- Строка табличной части сохраняется в таблицу БД;
- Реквизит табличной части сохраняется в столбец соответствующей таблицы;
- При сохранении документа и справочника всегда сохраняется ссылка на объект - его уникальный идентификатор (GUID), для документов и справочников по умолчанию этот столбец считается первичным ключом;
- При сохранении набора записей регистра накопления, всегда сохраняется GUID регистратора – документа;
- При сохранении табличной части, всегда сохраняется GUID родительского объекта;
- Перед сохранением объекта, если в настройках мэппинга задан «Ключ удаления» (подробнее см. ниже), то выполняется удаление соответствующей строки в таблице по этому ключу удаления;
- Перед сохранением объекта всегда выполняется очистка его табличных частей по GUID;
- Перед сохранением набора записей регистра накопления выполняется очистка по ключу – GUID-у документа-регистратора, движение по которому сохраняется;
- При сохранении объекта, в случае, если объект с таким же первичным ключом уже существует в БД, (синтаксис *ON CONFLICT ... DO UPDATE SET* в PostgreSQL и *ON DUPLICATE KEY UPDATE* в MySQL) то выполняется *UPDATE*, иначе *INSERT* данных.

Ввод настроек сохранения данных выполняется на закладке «Мэппинг в БД».



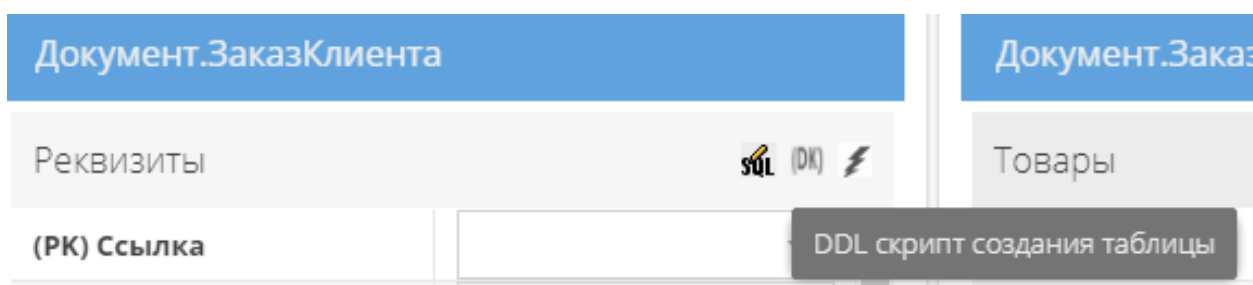
Подобно закладке «Конфигурация 1С» панель разделена на 3 рабочие области:

- **Объекты обмена** - список объектов, включенные в выгрузку;
- **Реквизиты и Табличные части**, включенные в выгрузку;
- **Реквизиты табличных частей**, включенные в выгрузку.

В отличие от закладки *Конфигурация 1С* на этой панели показаны только те объекты, которые включены в выгрузку данных (проставлены галочки). Для каждого из объектов выгрузки (**документов, справочников, регистров**) а также их табличных частей необходимо выбрать таблицу выгрузки, а для каждого из реквизитов соответственно — столбец. Списки доступных таблиц и столбцов загружаются из справочников метаданных - *information_schema.tables/columns* - соответственно. Наименование базы данных и схемы, в которую будут выгружаться данные, задается в конфигурационном файле *cnotone-config.properties* параметрами *connection.db.*, подробнее см. в разделе Конфигурационные параметры.

Автоматическое создание таблиц и привязка к ним

В случае, если таблиц базы данных, в которые будет осуществляться выгрузка объекта 1С, еще нет, есть возможность автоматически их создать и привязать к ним объект 1С с его реквизитами и табличными частями. Для этого нужно кликнуть на иконку **SQL** на заголовке панели реквизитов объекта:



Появится модальное окно, с автоматически сгенерированным .sql DDL скриптом для создания всех таблиц (и основного объекта и табличных частей), скрипт при желании можно отредактировать и запустить кнопкой **Выполнить скрипт на целевой БД**. После успешного выполнения скрипта нажатием кнопки **Проставить автоматический мэппинг** задается привязка для вновь созданных таблиц и объектов конфигурации 1С.

Ключ удаления

Вторая иконка на заголовке панели реквизитов открывает модальное окно для задания столбцов, по которым будет выполняться удаление старой строки в БД перед сохранением объекта.

Ключ удаления

☒ Удалять старый объект перед сохранением

"d_REALIZATSIJA_TOVAROV_USLUG", ключ...

GUID	<input checked="" type="checkbox"/>
ADRES_DOSTAVKI	<input type="checkbox"/>
AVTOR_NAIMENOVANIE	<input type="checkbox"/>
BUHGALTERSKIY_UCHET	<input type="checkbox"/>
DATA	<input checked="" type="checkbox"/>
DATA_PLATEZHA	<input type="checkbox"/>
GLAVNIY_BUHGALTER_DOLZHNOST	<input type="checkbox"/>
GLAVNIY_BUHGALTER_NAIMENOVANIE	<input type="checkbox"/>

ЗаписатьОтмена

Пример на скриншоте задает ключ, по которому будет выполняться удаление:

delete from d_REALIZATSIJA_TOVAROV_USLUG where GUID=? and DATA=?

Для **документов** и **справочников** элементами ключа могут быть произвольные реквизиты объекта, для **регистров сведений** и **регистров накопления** соответствующие реквизиты должны быть включены в **Отбор** при сохранении набора записей регистра.

Триггеры

Третья иконка открывает модальное окно для задания триггеров, выполняемых до и после сохранения объекта.

Триггеры

"d_REALIZATSIJA_TOVAROV_USLUG"

☒ Перед записью

sql-скрипт
вызываемый
перед записью
объекта:

call trgRealizatsiya_Before(&GUID);

☒ После записи

☒ Выполнять асинхронно

sql-скрипт
вызываемый
после записи
объекта:

call trgRealizatsiya_After(&GUID);

Записать

Отмена

Триггеры выполняются до (и/или) после сохранения всего объекта (основного объекта и всех его табличных частей) для документа или справочника, либо до (и/или) после сохранения всего набора записей регистра. В тексте скриптов можно ссылаться на идентификатор сохраняемого объекта (или регистратора для регистра) через переменную **@GUID** и на все остальные сохраняемые реквизиты объекта посредством синтаксиса **@НаименованиеРеквизита**. Можно задать выполнение триггеров асинхронным (в отдельном пуле потоков), проставив соответствующую галочку.

Публикация изменений

После выполнения любых изменений в настройках выгрузки они не активируются сразу же, для применения в `spotone-runtime` их необходимо опубликовать, кнопка **Опубликовать изменения** в правом верхнем углу консоли:

конфигурация обмена данными изменена...

Опубликовать изменения

Отменить изменения

В случае, если последние изменения были ошибочными их можно полностью откатить, нажав на кнопку **Отменить изменения**, что вернет метаданные к последнему опубликованному состоянию.

Остановка и запуск выгрузки

После того, как все метаданные выгрузки будут введены, можно активировать выгрузку в `snotone-runtime` кнопкой запуска в верхней статусной панели экрана.



При необходимости выгрузку можно выключить соседней кнопкой **стоп**.

Внимание! При остановке выгрузки отсутствует какая-либо отложенная выгрузка, соответствующие события **ПриЗаписи** и пр. просто будут игнорироваться в `snotone`-модуле 1С.

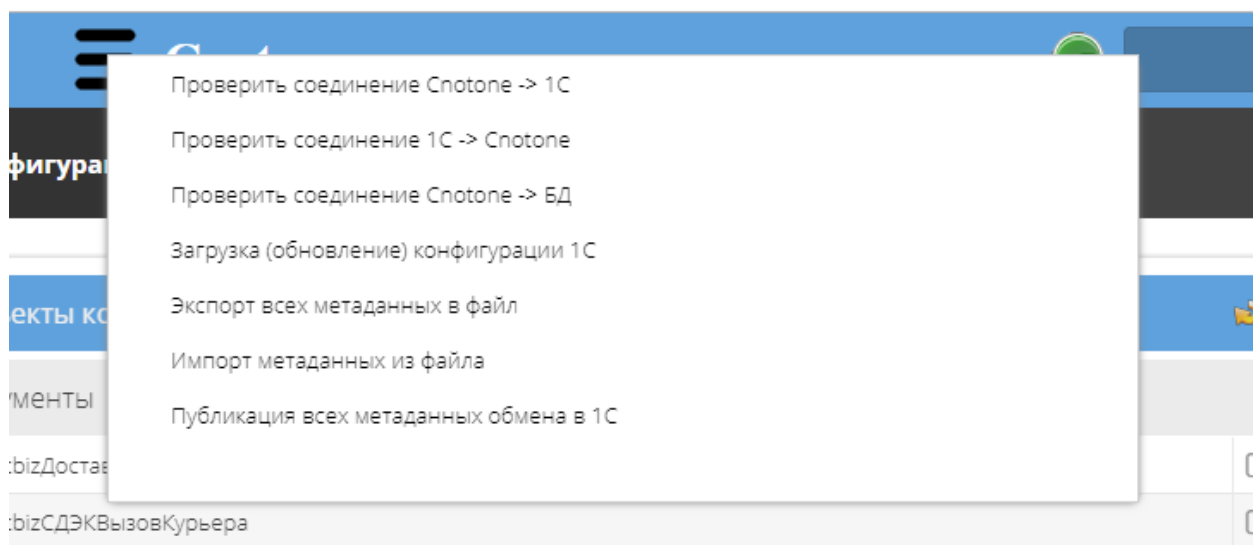
Внимание! Остановка выгрузки не влияет на выполнение входящей синхронизации, подробнее см. в Входящая синхронизация

Рядом с кнопками управления расположен статусный индикатор, показывающий цветом статус работы системы:

- зеленый - выгрузка выполняется;
- серый - выгрузка остановлена;
- красный - выгрузка остановлена из-за ошибки (отсутствует какое-либо из соединений, либо в системе произошел сбой), при наведении мыши на индикатор выводится причина ошибки.

Главное меню

В левом верхнем углу консоли находится главное меню системы:



- **Проверить соединение Cnotone -> 1С** - выполняется ping запрос к http-сервису 1С, при успешном ответе выводится сообщение с описанием версии платформы и конфигурации 1С;
- **Проверить соединение 1С -> Cnotone** - выполняется ping запрос к 1С с ожиданием ping-ответа (для проверки соединения от 1С к `snotone`), при успешном ответе выводится полное время пинга;
- **Проверить соединение Cnotone** - выполняется проверка соединения к БД, при успешном ответе выводится сообщение с версией СУБД;
- **Загрузка (обновление) конфигурации 1С** - выполняется полное обновление метаданных из 1С, это действие необходимо выполнять при первичной загрузке метаданных, а также после плановых обновлений конфигураций 1С;

- **Экспорт всех метаданных в файл** - все введенные настройки выгрузки выгружаются в файл .json, используется для бэкапа настроек либо для переноса их на другой сервер;
- **Импорт метаданных из файла** - импорт метаданных выгрузки из файла .json, полученного в предыдущем пункте меню, все предыдущие данные стираются;
- **Публикация всех метаданных обмена в 1С** - все метаданные выгрузки публикуются в 1С, может понадобиться после импорта из файла либо в случае если cnotone-справочник метаданных в 1С был вручную изменен средствами 1С;

Конфигурационные параметры

Конфигурационные параметры системы задаются в файле `.properties` (по умолчанию `cnotone-config.properties`), полный путь к которому необходимо передавать при запуске `Cnotone.jar` в параметре `-DcnotoneConfig`.

Настройки Cnotone веб-сервера

Параметр	Описание	Вариант заполнения/значение по умолчанию
cnotone.host	Хост, на котором запущен Cnotone	127.0.0.1
cnotone.port	Порт	8080
jetty.context.path	Контекст URL Cnotone	/cnotone

Системные настройки

Параметр	Описание	Варианты заполнения (значения по умолчанию)
use.password.authentication	Вход в cnotone-консоль защищен паролем	true/false
cnotone.admin.password	Пароль для входа в cnotone-консоль	
session.invalidate.time.min	Время сбрасывания сессии в минутах	120
hsql.db.location	Путь к внутренней базе Cnotone	C:/cnotone/db/hsql-cnotone

Настройки соединения с 1С

Параметр	Описание	Вариант заполнения/значение по умолчанию
connection.1c.http.url	URL по которому опубликован 1С Cnotone http сервис	[1]
connection.1c.usePasswordAuthentication	Использовать аутентификацию при обращении к 1С	true/false
connection.1c.admin.login	Логин пользователя 1С для вызова сервиса	Администратор
connection.1c.admin.password	Пароль пользователя 1С	

Настройки соединения с СУБД

Параметр	Описание	Вариант заполнения/значение по умолчанию
connection.db.engine	Тип СУБД	MYSQL/POSTGRESQL/ORACLE
connection.db.driverClass	jdbc драйвер подключения к БД	com.mysql.jdbc.Driver/org.postgresql.Driver/oracle.jdbc.OracleDriver
connection.db.jdbcUrl	URL jdbc подключения к БД	jdbc:mysql://127.0.0.1:3306/db_1c?useUnicode=yes&characterEncoding=utf8&allowMultiQueries=true&autoReconnect=true /jdbc:postgresql://127.0.0.1:5432/db_1c?useUnicode=yes&characterEncoding=utf8&allowMultiQueries=true&autoReconnect=true /jdbc:oracle:thin:scott/tiger@127.0.0.1:1521:orcl
connection.db.name	Имя базы данных	
connection.db.schema	Схема базы данных (для postgresql)	

connection.db.user	Пользователь подключения к БД	
connection.db.password	Пароль подключения к БД	
connection.db.preferredTestQuery	Пинговый запрос для проверки подключения	select 1/select 1 from dual
async.saving.task.pool.size	Число потоков, обрабатывающих асинхронное сохранение данных при выгрузке и асинхронный вызов триггеров	16

Настройки журналирования

Параметр	Описание	Вариант заполнения/значение по умолчанию
db.sync.log	Сохранять в БД информацию о событиях при выгрузке объектов из 1С, подробнее см. в Журналирование	true/false
db.sync.log.data	Сохранять в БД полные данные объекта при выгрузке из 1С, подробнее см. в Журналирование	true/false
db.syncin.log	Сохранять в БД информацию о входящих вызовах в 1С, подробнее см. в Журналирование	true/false
db.syncin.log.data	Сохранять в БД полные данных входящих вызовов в 1С, подробнее см. в Журналирование	true/false
log.1c.enabled	Выполнять логирование в файл в 1С	true/false
log.metadata1CFileName	Лог изменения Cnotone метаданных в 1С	C:\nотone\logs\1c\1c_metadata.log
log.metadata1CLevel	Уровень лога метаданных	ERROR/WARN/INFO/DEBUG/TRACE
log.sync1CFileName	Лог выгрузки объектов из 1С	C:\nотone\logs\1c\1c_sync.log
log.sync1CFileLevel	Уровень лога выгрузки объектов из 1С	ERROR/WARN/INFO/DEBUG/TRACE
log.syncin1CFileName	Лог входящих вызовов в 1С	C:\nотone\logs\1c\1c_sync_in.log
log.syncin1CLevel	Уровень лога входящих вызовов в 1С	ERROR/WARN/INFO/DEBUG/TRACE

Лицензия

Параметр	Описание	Вариант заполнения/значение по умолчанию
license.user.name	На кого зарегистрирован продукт	не заполнять для пробного периода
license.serial.key	Серийный ключ, TRIAL для пробного периода	8080

Примечания

[1] <http://localhost:1111/My1C/hs/cnotone>

Журналирование

Подсистема журналирования предназначена для мониторинга процесса синхронизации данных, отладки и оперативного устранения ошибок. Журналирование выполняется в трех контекстах:

- запись отправляемых и поступающих сообщений в базу данных;
- файловые логи работы системы на стороне 1С;
- файловые логи системы в cnotone-runtime и cnotone-консоли;

Запись событий в базу данных

Все события обрабатываемые системой: события выгрузки объектов из 1С и входящие вызовы функций могут записаны в лог в таблицы базы данных. По умолчанию лог отключен, включается установкой параметров в конфигурационном файле (см. Конфигурационные параметры) `db.sync.log = true` - для выгрузки из 1С и `db.syncin.log = true` - для входящих вызовов. Перед включением параметров необходимо создать в той же базе данных, в которую выгружаются данные из 1С, таблицы `sync_log` и `syncin_log` (см. #Приложение 1. Скрипты создания таблиц для журналирования в БД). Включением параметров `db.sync.log.data = true` и `db.syncin.log.data = true` активируется полная запись входящих и исходящих запросов, включая тело сообщений (json - данные). Данные журналов доступны в cnotone-консоле на вкладке **Инструменты и мониторинг**, подзакладки - **Журнал исходящей синхронизации** и **Журнал входящей синхронизации** соответственно.

Дата/время	GUID	Событие	Тип данных	Результат	Данные
2019.03.14 07:11...	67b94730-43ce...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 07:11...	67b94730-43ce...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 07:11...	d93ea078-44a4...	ПриЗаписи	Документ.РеализацияТоваров/услуг	OK	{ "НаименованиеТипа": "Документ.Реа...
2019.03.14 07:11...	67b94730-43ce...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 07:06...	59397ae2-43d5...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 07:06...	59397ae2-43d5...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 07:06...	33fa093-44a6-1...	ПриЗаписи	Документ.РеализацияТоваров/услуг	OK	{ "НаименованиеТипа": "Документ.Реа...
2019.03.14 07:06...	59397ae2-43d5...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 06:41...	67b94730-43ce...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 06:41...	67b94730-43ce...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 06:41...	d93ea078-44a4...	ПриЗаписи	Документ.РеализацияТоваров/услуг	OK	{ "НаименованиеТипа": "Документ.Реа...
2019.03.14 06:41...	67b94730-43ce...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 06:21...	59397ae2-43d5...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...
2019.03.14 06:21...	59397ae2-43d5...	ПриЗаписи	Документ.ЗаказКлиента	OK	{ "НаименованиеТипа": "Документ.Зак...

Для сортировки или фильтрации по столбцу необходимо навести мышку на его заголовок, нажать на стрелку и выбрать сортировку или ввести фильтр. Для фильтрации по подстроке используется метасимвол %, например `%строка%поиска%`.

Журнал входящей синхронизации

Результат	Данные	Асинхронно
ОК	{ "functionName": "СинхронизацияССайтом.ЗаписатьТовар", "async": true, "functionParams": { "ID": "10134", "Parent": "" } }	<div><div>Sort Ascending</div><div>Sort Descending</div><div>Columns</div><div>Filters</div></div> <div><div>10134%</div></div>

Дата/время	Функция
2019.03.06 23:37:52	Сайтом.ЗаписатьЗаказКлиентаССайта
2019.03.06 23:37:48	Сайтом.ЗаписатьПартнера
2019.03.06 21:07:21	Сайтом.ЗаписатьТовар
2019.03.06 21:07:19	СинхронизацияССайтом.ЗаписатьТовар
2019.03.06 21:04:09	СинхронизацияССайтом.ЗаписатьТовар
2019.03.06 20:58:50	СинхронизацияССайтом.ЗаписатьТовар
2019.03.06 20:58:46	СинхронизацияССайтом.ЗаписатьТовар
2019.03.06 19:13:07	СинхронизацияССайтом.ЗаписатьЗаказКлиентаССайта
2019.03.06 19:13:05	СинхронизацияССайтом.ЗаписатьПартнера
2019.03.06 19:10:55	СинхронизацияССайтом.ЗаписатьЗаказКлиентаССайта
2019.03.06 19:10:52	СинхронизацияССайтом.ЗаписатьПартнера
2019.03.06 19:05:20	СинхронизацияССайтом.ЗаписатьЗаказКлиентаССайта
2019.03.06 19:05:18	СинхронизацияССайтом.ЗаписатьПартнера
2019.03.06 19:00:53	СинхронизацияССайтом.ЗаписатьЗаказКлиентаССайта

« < | Page 1 of 26 | > »

Today

Общие принципы логирования в файл

Для любого лог-файла в системе можно настроить один из 5 стандартных уровней логирования:

Настройки Уровни логирования

Уровень	Описание
TRACE	Подробное описание происходящего, полный текст входящих и исходящих сообщений
DEBUG	Фиксирование контрольных точек выполнения кода, вывод отладочной информации - параметров и результат вызова функций/процедур
INFO	Общая информация о работе системы, фиксирование действий пользователя в консоли
WARN	Предупреждения о некорректных настройках или данных и прочих ошибочных ситуациях, которые не влияют на корректную работу системы
ERROR	Ошибки, которые влияют на корректную работу системы, сигнализирующие о сбое или о неправильных настройках

Уровень отображения с более высокой детализацией показывает все сообщения уровней более низкой детализации, например DEBUG выводит также сообщения уровней INFO,WARN,ERROR. А уровень WARN выводит также ERROR.

Для production-систем обычно выставляются уровни WARN или INFO, в среде разработки или при выявлении ошибок в production-е выставляются TRACE или DEBUG.

Файловые логи 1C

Логи 1C настраиваются в *cnotone-config.properties* (подробнее см. Конфигурационные параметры), для каждого из логов прописываются наименование файла и уровень логирования. Для включения логирования в 1C необходимо установить флаг *log.1c.enabled=true*. На каждые сутки создается отдельный лог-файл каждого типа с именем формата *"ИмяФайлаИзНастроек_ДатаЛога"*.

Cnotone-модуль 1C пишет в три лога

- *log.metadata1CFileName* - лог изменения Cnotone метаданных в 1C. Отображаются изменения в настройках выгрузки опубликованные из cnotone-консоли а также события обновления метаданных;
- *log.sync1CFileName* - лог выгрузки объектов из 1C. Отображают события формирования данных выгрузки и отправки их в cnotone-runtime, а также события включения/остановки выгрузки;
- *log.syncin1CLevel* - лог входящих вызовов в 1C. Отображает вызовы функций 1C из сторонних систем, отправленные из cnotone-runtime.

Файловые логи cnotone-runtime и cnotone-консоли

Логи cnotone-runtime используют стандартную java-библиотеку Logback ^[1], конфигурация логирования cnotone настраивается в файле *logback-cnotone.xml* (входит в поставку, путь к файлу - параметр *-Dlogback.configurationFile* при запуске приложения), при необходимости можно изменить пути к файлам под тегами *<file>* соответствующего лога, например

```
<appender name="MAIN" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>C:/cnotone/logs/cnotone-main.log</file>
```

и уровни логирования, атрибут *level* под тегом *logger* соответствующего лога, например :

```
<logger name="TRACER" level="TRACE" additivity="false">
```

Cnotone-runtime пишет в 7 логов:

- *cnotone-main.log* - основной лог, в него пишется все из всех логов;
- *cnotone-errors.log* - лог сообщений об ошибках, пишутся только ошибки;
- *cnotone-trace.log* - лог трассировки входящих и исходящих json-сообщений, пишется только при включенной трассировке (TRACE - уровень);
- *cnotone-tasks.log* - лог пакетных заданий по загрузке данных и заданий на контроль данных;
- *cnotone-sync.log* - лог выгрузки данных из 1C, пишутся события выгрузки, пинг-запросы, события включения/остановки выгрузки;
- *cnotone-webui.log* - лог действий пользователя по изменению метаданных в cnotone-консоли;
- *cnotone-metadata-update.log* - лог публикации метаданных в 1C и обновления метаданных из 1C (аналогичен логу *log.metadata1CFileName* в 1C)

Приложение 1. Скрипты создания таблиц для журналирования в БД

MySql

```
CREATE TABLE sync_log (  
    log_id BIGINT(20) NOT NULL AUTO_INCREMENT,  
    event VARCHAR(50) ,  
    guid VARCHAR(50),  
    type_name VARCHAR(500),  
    event_time DATETIME,  
    info LONGTEXT,  
    result VARCHAR(4000),  
    PRIMARY KEY (log_id),  
    INDEX sync_log_ix1 (event_time),  
    INDEX sync_log_ix2(guid),  
    INDEX sync_log_ix3(result(255))  
);  
  
CREATE TABLE syncin_log (  
    log_id BIGINT(20) NOT NULL AUTO_INCREMENT,  
    function_name VARCHAR(50),  
    event_time DATETIME,  
    info LONGTEXT,  
    result VARCHAR(4000),  
    async TINYINT(4),  
    PRIMARY KEY (log_id),  
    INDEX syncin_log_ix1 (event_time)  
);
```

PostgreSql

```
CREATE SEQUENCE sync_log_seq;  
CREATE TABLE sync_log  
(  
    log_id bigint NOT NULL DEFAULT nextval('sync_log_seq'::regclass),  
    event character varying(50),  
    guid character varying(50),  
    type_name character varying(500),  
    event_time timestamp,  
    info text,  
    result character varying(4000),  
    CONSTRAINT sync_log_pkey PRIMARY KEY (log_id)  
);
```

```
CREATE SEQUENCE syncin_log_seq;  
CREATE TABLE syncin_log  
(  
    log_id bigint NOT NULL DEFAULT nextval('syncin_log_seq'::regclass),  
    function_name character varying(200),  
    event_time timestamp,  
    info text,  
    result character varying(4000),  
    async boolean,  
    CONSTRAINT syncin_log_pkey PRIMARY KEY (log_id)  
);
```

Примечания

[1] <https://logback.qos.ch/>

Пакетная выгрузка и контроль данных

Пакетная выгрузка

Основная функциональность Cnotone обеспечивает выгрузку данных из 1С на основе механизма подписок на события. В некоторых случаях требуется выполнить пакетную выгрузку данных: например, полную выгрузку данных какого-то справочника, или документы за какой-то определенный интервал времени. Пакетная выгрузка в Cnotone реализована в cnotone-консоле на вкладке **"Инструменты и мониторинг"** подзакладка **"Задания на выгрузку"**, возможна одноразовая выгрузка а также выгрузка по расписанию. Выгрузка возможна только для тех объектов 1С, которые включены в выгрузку на вкладке **"Конфигурация 1С"** и для которых настроен мэппинг в БД.

Для одноразовой выгрузки заполняются поля формы:

- * **Метатип данных** – документ, справочник, регистр;
- * **Тип данных** – тип 1С;
- * **Условие на запрос данных** – условие фильтрации данных в 1С sql-запросе, например, условие на скриншоте задает запрос по которому будут выгружаться данные: *Выбрать * из Справочник.Партнеры где Наименование подобно "А%"* . Если не задать никакого условия, то будут выгружены все данные (справочника "Партнеры" в данном случае) .

По кнопке "Выполнить" проверяется корректность запроса и выводится количество объектов, которые будут загружены и запрашивается подтверждение на выгрузку.

Дополнительно, для настройки производительности выгрузки можно изменить параметры по умолчанию:

- * **Количество записей в одном пакете** – компромиссное значение – 100, для больших объектов его разумно уменьшить, для маленьких (записей регистра) – увеличить;
- * **Количество потоков, принимающих данные** – число параллельных потоков, обслуживающих сохранение данных в БД, нормальное значение – 1, имеет смысл ставить больше (1-10), если процесс сохранения данных в БД имеет пропускную способность ниже получения данных из 1С;
- * **Количество потоков, отправляющих данные** – число параллельных потоков в 1С, отправляющих данные, может существенно ускорить выгрузку, но нужно применять с осторожностью, т.к. есть риск существенно снизить

общую производительность 1С сервера на время выгрузки, используется механизм фоновых заданий 1С.

При выставлении параметра **Количество потоков, отправляющих данные** необходимо заполнять параметр **Условие в запросе на поток**, допустим у нас 6 потоков, тогда условие может быть такого типа: *и (ВЫРАЗИТЬ(Месяц(ДатаРегистрации)/2 КАК ЧИСЛО(12, 0))+1)=&threadNum*, в этом случае мы будем иметь 6 запросов (на каждый поток по запросу соответственно) вида:

*Выбрать * из Справочник.Партнеры где Наименование подобно "A%" и (ВЫРАЗИТЬ(Месяц(ДатаРегистрации)/2 КАК ЧИСЛО(12, 0))+1)=1*

*Выбрать * из Справочник.Партнеры где Наименование подобно "A%" и (ВЫРАЗИТЬ(Месяц(ДатаРегистрации)/2 КАК ЧИСЛО(12, 0))+1)=2*

....

По кнопке **Выполнить** выгрузка данных начинается сразу же. По кнопке **и (или) Запланировать** можно запланировать выполнение задания по расписанию, расписание задается **cron**-выражением.

При выборе подзакладки **Список заданий** открывается форма со списком заданий на выгрузку:

Наименование	CRON	Статус	Старт	Прогресс	Финиш	Сообщение
РегистрСведений.ОстаткиТоваро...		Выполняется	2019.03.16 19...	25%		
Справочник.Партнеры		Отменено	2019.03.16 18...	0%	2019.03.16 18...	{ОбщийМоду...

Для активных заданий показывается статус - процент выгрузки, для запланированных - расписание, для законченных - время и результат выполнения. Активные задания можно отменить кнопкой **стоп**. Переключатель **Только активные** скрывает уже выполненные задания. Кнопка **Скопировать задание** создает копию выбранного задания на подзакладке **Новое**.

Контроль данных

Внимание! Функционал, описанный в разделе **Контроль данных** находится на этапе опытной эксплуатации и поставляется по принципу "как есть", т.е. полностью корректная работа согласно описанию не гарантируется.

В любой 24/7-системе возможны дефекты работы, связанные с программными ошибками, аппаратными сбоями, нестабильностью каналов связи и пр. Система Cnotone позволяет в автоматическом режиме контролировать корректность выгружаемых данных и, по возможности, исправлять обнаруженные ошибки. Контроль корректности выгрузки данных реализован в cnotone-консоле на вкладке **"Инструменты и мониторинг"** подзакладка **"Задания на проверку"**. Проверка выполняется на основе введенных пользователем двух sql-запросов, один - для 1С, второй - для целевой БД выгрузки, предполагается, что при корректной выгрузке данных результаты запросов должны совпадать. Например, для проверки выгрузки документов "ЗаказКлиента" за 2019 год реквизиты формы нужно заполнить следующим образом (см.

скриншот):

The screenshot shows the 'Инструменты и мониторинг' (Tools and monitoring) section of the Cnotone application. The 'Добавить новое задание на проверку' (Add new task for checking) form is displayed. It includes the following fields and values:

- Тип задания:** Типизированное
- Наименование:** (empty)
- Метатип данных:** Документ
- Тип данных:** ЗаказКлиента
- Запрос к 1С:** Выбрать количество(*) из Документ.ЗаказКлиента где Дата >= ДАТАВРЕМЯ(2019,01,01) и Дата < ДАТАВРЕМЯ(2019,12,31)
- Запрос к БД:** select count(*) from d_ZAKAZ_KLIENTA where data>= TO_DATE(CONCAT('2019','-', '01','-', '01'), 'YYYY-MM-DD') and data< TO_DATE(CONCAT('2019','-', '12','-', '31'), 'YYYY-MM-DD')
- Реквизит "дата" в 1С:** Дата
- Столбец "дата" в БД:** data
- Расписание (cron - примеры):** 0 0 0/3 * * ?
- Дата начала контролируемого интервала:** 01/01/19
- Дата окончания контролируемого интервала:** 31/12/19
- Выполнять перезагрузку данных при обнаружении ошибок:** ☒
- Интервал проверки/перезагрузки:** День

Buttons at the bottom: Сформировать запросы, Проверить, and Добавить.

Задание на скриншоте каждые 3 часа (cron-выражение - 0 0 0/3 * * ?) выполняет запросы:

Выбрать количество(*) из Документ.ЗаказКлиента где Дата >= ДАТАВРЕМЯ(2019,01,01) и Дата < ДАТАВРЕМЯ(2019,12,31)

```
select count(*) from d_ZAKAZ_KLIENTA where data>= TO_DATE(CONCAT('2019','-', '01','-', '01'), 'YYYY-MM-DD') and data< TO_DATE(CONCAT('2019','-', '12','-', '31'), 'YYYY-MM-DD')
```

В случае, если результаты запросов не совпадают, запускается процесс исправление данных, по следующему алгоритму:

- * от **Даты окончания контролируемого интервала** до **Даты начала контролируемого интервала** (т.е. хронологически в обратном порядке) с шагом **Интервал проверки/перезагрузки** = *День*;
- * выполняются запросы, аналогичные основному, но на каждую дату, т.е. на первом шаге будут выполнен запрос в 1С – Выбрать количество(*) из Документ.ЗаказКлиента где Дата=ДАТАВРЕМЯ(2019,12,31), на втором – Выбрать количество(*) из Документ.ЗаказКлиента где Дата=ДАТАВРЕМЯ(2019,12,30) и т.д. Аналогичные запросы на каждом шаге будут также выполнены на целевой БД выгрузки;
- * в случае, если на каком-то шаге результаты запроса не будут совпадать, то запускается процесс выгрузки данных на ошибочную дату;
- * после успешной выгрузки данных проверяется еще раз эта же дата, если данные на эту дату исправлены, то выполняется проверка всего интервала, иначе завершение процесса исправления с сообщением об ошибке.
- * если результат проверочного запроса по всему интервалу корректен, то процесс исправления завершается успешно;
- * иначе продолжается проверка/исправление других дат до **Даты начала контролируемого интервала**.

Описание полей формы:

- * **Тип задания.** Произвольное – запрос не привязан к конкретному типу данных, исправление данных недоступно; Типизированное – запрос привязан к конкретному типу данных, возможна настройка исправления данных;

- * **Наименование** – наименование, которое будет отображаться в списке заданий на соседней вкладке;
- * **Метатип данных** – документ, справочник, регистр;
- * **Тип данных** – тип 1С;
- * **Запрос к 1С, Запрос к БД** – запросы, равенство результатов которых сравнивается при выполнении задания;
- * **Выполнять перезаливку данных при обнаружении ошибок** – флаг, включающий исправление данных при обнаружении ошибки;
- * **Реквизит "дата" в 1С, Столбец "дата" в БД** – реквизит, по которому будет выполняться обратный проход при исправлении данных;
- * **Расписание (cron)** – расписание выполнения задания;
- * **Даты начала контролируемого интервала и Даты окончания контролируемого интервала** – интервал проверки и исправления данных;
- * **Интервал проверки/перезаливки** – час/день/месяц

Кнопки:

- * **Сформировать запросы** – только для *типизированного* задания, автоматически на основе метаданных и значений полей **Дата начала контролируемого интервала** и **Дата окончания контролируемого интервала** формируется контролируемые запросы в 1С и в БД;
- * **Проверить** – выполнение и вывод результатов контролируемых запросов по введенным настройкам;
- * **Добавить** – добавление планового задания на контроль данных.

При выборе подзакладки **Список заданий** открывается форма со списком заданий на выгрузку:

Аналогично форме **Список заданий** на панели **Задания на выгрузку** доступны кнопки:

- * **Только активные** – переключатель, скрывающий неактивные задания;
- * **Скопировать задание** – создает копию выбранного задания на подзакладке **Новое**;

и для каждого из заданий в первом и последнем столбцах таблицы доступны кнопки:

- * **Выполнить задание сейчас** – запускает проверку и исправление;
- * **Выключить/Включить** – останавливает (делает неактивным)/запускает плановое выполнение задания;

* **Удалить** – удаляет задание;

при выполнении исправления данных в столбце **Посл.дата** отображается последняя проверенная/исправленная дата, а также доступна дополнительная кнопка **Остановить**, останавливающая процесс исправления данных.

Входящая синхронизация

Основной функционал системы Cnotone связан с выгрузкой данных из 1С во внешнюю реляционную базу данных. Для обратной загрузки данных в 1С в системе не предусмотрен какой-либо специальный механизм на основе метаданных, упрощающий процесс настройки загрузки. Поддержка входящей синхронизации в 1С реализована в виде отдельного http-сервиса, принимающего REST-запросы с json-body, в котором передаются параметры вызова функции 1С. Пошагово процесс вызова:

```
* внешняя система отправляет http POST сервису cnotone по адресу http://
/{cnotone.host}:{cnotone.port}/{jetty.context.path}/rest/
sync-in/call-function' (по умолчанию http://localhost:8080/
cnotone/rest/sync-in/call-function') с телом json, в котором
передаются параметры вызова в следующем формате:
```

```
{
  "functionName" : "ИмяМодуля.Имя1СФункции",
  "async" : true/false -асинхронный/синхронный вызов,
  "functionParams" : json-object параметров функции
}
```

```
* cnotone-runtime принимает вызов, записывает в логи (см.
Журналирование Журнал входящей синхронизации), сразу возвращает ответ
и ставит вызов в очередь если async=true, отправляет вызов в 1С
cnotone-модуль если async=false;
* cnotone-модуль преобразует json-object параметров в структуру 1С
(json-object -> структура 1С, json-array -> массив 1С), и
вызывает функцию 1С с единственным аргументом – этой структурой;
* после выполнения функции результат успешного вызова или сообщение об
ошибке передается обратно в cnotone-runtime, где выполняется
журналирование и, в случае синхронного вызова, передача http-ответа
внешней системе.
```

Код вызова из системы, написанной на php, может выглядеть примерно следующим образом:

```
$partner = array(
  "lName"=>"Петров",
  "fName"=>"Сергей",
  "sName"=>"Иванович",
  "phone"=>"+79261234567",
  "email"=>"petrov@mail.ru",
  "address"=>"Москва, Петровка 16-37",
  "inn"=>"348119982323",
  "passport"=>"4500 123456",
  "country"=>"РФ")
);
```

```
$tp='{ "functionName":"СинхронизацияССайтом.ЗаписатьПартнера","async":false,"functionParams":'.json_encode($p
JSON_UNESCAPED_UNICODE)."}';

try {
    $ch = curl_init($SERVER_1C_CNOTONE_URL);
    curl_setopt($ch, CURLOPT_HEADER, 1);
    curl_setopt($ch,CURLOPT_NOBODY,0);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $tp);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: text/plain'));
    ob_start();
    $out = curl_exec($ch);
} catch (Exception $e) {
}
```

Для корректного преобразования json в 1С-структуру наименования полей json-объекта должны удовлетворять требованиям, предъявляемым к наименованию реквизитов 1С структуры (отсутствовать пробелы в наименовании и прочее).