

Использование COM-объектов в обработке 1С

Последние изменения: 2024-03-26

Клеверенс: Настройка подключения *



Тип интерфейса подключения к базе Mobile SMARTS: REST API Компонента Mobile SMARTS

Укажите тип интерфейса взаимодействия с базой Mobile SMARTS для получения, отправки данных в подключенную базу.

REST API - обмен посредством HTTP запросов в формате OData. Компонента Mobile SMARTS - внешняя компонента доступа разработанная компанией Клеверенс.

Описание методов и свойств COM-объекта "TerminalConnector".

Описание методов и свойств остальных COM-объектов из пространств имен "Cleverence.Common", "Cleverence.Infrastructure" и "Cleverence.Warehouse"

Общая информация

1. Обмен данными между 1С и Mobile SMARTS может инициировать только 1С. Реализовано это через возможность работать в 1С с внешними компонентами. У Mobile SMARTS специально для 1С разработана внешняя компонента, представленная файлами «Cleverence.Infrastructure.dll», «Cleverence.MobileSMARTS.ComConnector.dll», «Cleverence.Warehouse.TerminalConnector.dll». Для того чтобы была возможность использовать в 1С внешние компоненты, необходимо их зарегистрировать в операционной системе.
2. Обмен при помощи COM-объектов возможно и **в глобальном, и в локальном ядре**. Код обмена практически идентичный. Но компонента обязательно должна быть зарегистрирована на той стороне, где выполняется обмен (соответственно, клиент 1С или сервер 1С)
3. В ядре используются следующие COM-объекты:
 1. **COMОбъект("Cleverence.Warehouse.StorageConnector")** — объект, с помощью которого можно взаимодействовать с Mobile SMARTS.
 2. **COMОбъект("AddIn.Cl.TerminalConnector")** — объект, посредством которого также осуществляется взаимодействие с Mobile SMARTS. Позволяет выполнять соединение с Mobile SMARTS, выгружать справочник номенклатуры и дополнительные таблицы, выгружать документы Mobile SMARTS для исполнения на терминале, загружать завершенные документы Mobile SMARTS в 1С. Отличие от Cleverence.Warehouse.StorageConnector заключается в том, что для обмена данными используются объекты 1С (массивы, списки значений, таблицы значений и др.), что существенно упрощает написание кода обмена в 1С. Фактически, это еще один слой абстракции, высокоуровневая обертка над объектом StorageConnector, т.к. из TerminalConnector можно явно получить StorageConnector методом TerminalConnector.ПолучитьОбъектСоединенияСБазой().
 3. **COMОбъект("Cleverence.Warehouse.Document")** — сущность, необходимая для создания документа Mobile SMARTS, заполнения его реквизитами и выгрузки в Mobile SMARTS и обратно — для получения документа Mobile SMARTS с сервера Mobile SMARTS и загрузки в 1С.
 4. **COMОбъект("Cleverence.Warehouse.DocumentItem")** — строка табличной части документа MS, добавляется в табличную часть «СтрокиПлан» (DeclaredItems) для последующей выгрузки документа Mobile SMARTS.

5. **COMОбъект("Cleverence.Warehouse.Table")** — дополнительная табличная часть документа MS, то есть не «СтрокиПлан» и не «СтрокиФакт». Используется для выгрузки документа из 1С в MS.
6. **COMОбъект("Cleverence.Warehouse.Row")** — строка дополнительной табличной части документа MS. Создается, заполняется данными и добавляется в табличную часть методом Add()
7. **COMОбъект("Cleverence.Warehouse.DocumentDescriptionCollection")** — коллекция, содержит элементы «DocumentDescription», используется в [онлайн-режиме](#) для получения списка документов из 1С в методе «ПолучитьСписокДокументов()» и отображения их на ТСД
8. **COMОбъект("Cleverence.Warehouse.DocumentDescription")** — элемент коллекции «DocumentDescriptionCollection», содержит Ид, Имя, ТипДокумента1С и т.д.

Механизм проверки и поддержания постоянного соединения с базой MS

Для того чтобы постоянно поддерживать соединение с [базой Mobile SMARTS](#), необходимо чтобы объекты StorageConnector и TerminalConnector были инициализированы в течение всего времени, пока запущена обработка-ядро и осуществляется обмен с Mobile SMARTS. Если обмен происходит на клиенте 1С, то эти объекты сохраняются в переменные локального ядра COMЧастнойБазыMS / COMВременнойБазыMS / COMБазыMS. Если же обмен с Mobile SMARTS происходит на сервере 1С, то между серверными вызовами никакие данные на сервере сохранить нельзя, т.е. сервер 1С работает в режиме «без состояния» («stateless») - серверный вызов отработал и уничтожил все локальные переменные. Но нам необходимо, чтобы COM-объекты существовали все время, пока открыта обработка, поэтому данные объекты помещаются во временное хранилище на сервере 1С и находятся там на все время существования обработки-ядра, в переменной ядра «АдресаПеременных». Но возможны ситуации, когда соединение с базой Mobile SMARTS по различным причинам разрывается, в т.ч. потому что объекты StorageConnector и TerminalConnector уничтожаются из временного хранилища, т.к., в общем случае, временное хранилище не предназначено для хранения COM-объектов. Также возможна ситуация, что объекты будут уничтожены из временного хранилища, если серверные вызовы будут обрабатываться не одним, а разными рабочими серверами 1С, физически расположенными на разных компьютерах, входящих в состав кластера серверов 1С.

Поэтому, чтобы при каждом обмене с Mobile SMARTS не возникало ошибок из-за отсутствия соединения, во всех методах, где осуществляется обмен, вызывается проверка соединения с базой Mobile SMARTS. Проверка включает в себя несколько этапов:

1. Созданы ли COM-объекты StorageConnector и TerminalConnector и существуют ли они еще во временном хранилище (для глобального ядра). Для этих целей предназначен метод ядра «КомЯдро_ПолучитьСтруктуруПодключенияБазыСМАРТС()», который проверяет существование объектов StorageConnector и TerminalConnector, в случае их отсутствия, создает их, помещает во временное хранилище, в переменную ядра «АдресаПеременных» (для глобального ядра) или в переменные ядра «COMЧастнойБазыMS / COMВременнойБазыMS / COMБазыMS» (для локального ядра), а в переменную ядра булевого типа «КОМсоздан» сохраняет результат об успешности создания. Если же, по каким-то причинам, не удалось создать объекты, то переменные ядра, хранящие эти объекты, очищаются, а флагу «КОМсоздан» присваивается значение Ложь. Дополнительно существует метод, который только возвращает значение переменной ядра КОМсоздан, но не создает сами объекты — «ЛокЯдро_ПроверитьСозданиеКОМОбъекта()», используется только в главной форме, в нескольких методах, более нигде.
2. Если предыдущий этап пройден успешно, пытаемся подключиться к конкретной базе по строке подключения, это делается в методе «КомЯдро_ПодключитьсяКСерверуSMARTS()», если необходима авторизация по токenu, то по логину и паролю получаем токен и авторизуемся по токenu. При необходимости, можно во входном параметре явно указать, что нужно переподключиться к базе Mobile SMARTS.
3. Далее, метод «КомЯдро_ПодключитьсяКБазеSMARTS()» — если не заполнена структура описания базы (Ид базы, СтрокаПодключения, Логин, Токен, АвторизацияВключена), то получаем базу Mobile SMARTS по строке подключения, затем получаем структуру описания базы. Если авторизация включена, но не заполнен токен, тогда вызываем «КомЯдро_ПодключитьсяКБазеSMARTSHaСервереЗавершение()». Если авторизация не включена, снова подключаемся к базе и авторизовываемся по токenu, затем вызываем «КомЯдро_ПодключитьсяКБазеSMARTSHaСервереЗавершение()».

4. «КомЯдро_ПодключитьсяКБазеSMARTSНаСервереЗавершение()» — метод используется в том числе из формы авторизации, где вводятся логин, пароль и флаг «ЗапомнитьАвторизацию», поэтому у него название как у обработчика оповещения. В методе опять получаем структуру описания базы, проверяем соединение, получаем ID и строку подключения базы и сохраняем все в кэш, в переменную ядра «АдресаПеременных».
5. КомЯдро_ПроверитьСоединениеССерверомSMARTS() — вызывает КомЯдро_ПодключитьсяКСерверуSMARTS() (пункт 2), затем КомЯдро_ПодключитьсяКБазеSMARTS() (пункт 3)
6. Точка входа, которая вызывается во всех методах, где есть обмен с Mobile SMARTS — «ЛокЯдро_ПроверитьСоединениеССерверомSMARTS()», логика его работы такова, что в этот метод почти всегда передают обработчик оповещения, в котором и происходит обмен с MS (например, выгрузка справочников, документов, сохранение/загрузка настроек и т.д.). Если все проверки в данном методе не будут выполнены, то метод вернет «Ложь» и в обработчике оповещения не будет выполняться обмен с MS.

Например, когда мы из формы обмена документами хотим удалить документа, то запускаем метод проверки соединения с сервером Mobile SMARTS, а после этого, в обработчике оповещения проверяем флаг «СерверДоступен», и, если сервер недоступен, то не выполняем обработчик:

```

&НаКлиенте
□ Процедура КомПанель_ДокументыMSУдалитьДокумент (Команда)
    ОП = Новый ОписаниеОповещения ("КомПанель_ДокументыMSУдалитьДокументПослеПодключения", ЭтаФорма);
    _ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS (, , ТипБазы, , , ОП);
- КонецПроцедуры

&НаКлиенте
□ Процедура КомПанель_ДокументыMSУдалитьДокументПослеПодключения (СерверДоступен, ДопПараметры) Экспорт
    Если НЕ СерверДоступен = Истина Тогда
        Возврат;
    КонецЕсли;

```

Примеры вызовов из других форм, где выполняется обмен с Mobile SMARTS:

.ОбменДокументами.Форма.Модуль(181, 15)	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ОбменДокументами.Форма.Модуль(200, 15)	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ОбменДокументами.Форма.Модуль(219, 15)	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ОбменДокументами.Форма.Модуль(238, 15)	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ФормаСписка_БизнесПроцессы.Форма.Мо...	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ..., ОП);
.ФормаСписка_ОтборыСправочников.Форм...	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ФормаСписка_ОтборыСправочников.Форм...	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ФормаСписка_ОтборыДокументов.Форма...	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);
.ФормаСписка_ОтборыДокументов.Форма...	_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS(, ТипБазы, ..., ОП);

Итого, стек вызовов методов для проверки и поддержки постоянного соединения с базой Mobile SMARTS выглядит следующим образом:

```

ЛокЯдро_ПроверитьСоединениеССерверомSMARTS()
    -> КомЯдро_ПроверитьСоединениеССерверомSMARTS()
-> КомЯдро_ПодключитьсяКБазеSMARTS() (пункт 3)
-> КомЯдро_ПодключитьсяКСерверуSMARTS() (пункт 2)
-> КомЯдро_ПолучитьСтруктуруПодключенияБазыSMARTC() (пункт 1)
    -> КомЯдро_ПодключитьсяКБазеSMARTSНаСервереЗавершение() (пункт 4)

```

то есть метод «ЛокЯдро_ПроверитьСоединениеССерверомSMARTS()» как раз и вызывается во всех методах, где происходит взаимодействие с Mobile SMARTS.

Подключение к удаленной базе по строке подключения

Используя COM-объекты можно подключаться к базе, расположенной на любом компьютере в сети, а не только к той, которая расположена локально, т.е. на том же компьютере, где и COM-объекты. Для этого достаточно знать абсолютный путь до базы, включающий в себя имя / IP-адрес компьютера с базы, порт базы и Ид базы, например:

- `http://server_name:10501/c65ccd10-6daa-496e-92aa-80043476f2e8`
- `http://8.8.8.8:10501/c65ccd10-6daa-496e-92aa-80043476f2e8`

Прочее

1. Начиная с версии платформы Mobile SMARTS 3.2 для обмена с базой Mobile SMARTS появилась возможность использовать интерфейс **REST API**. Его использование предпочтительнее, т.к. он работает быстрее, стабильнее, чем внешняя компонента, нет необходимости регистрировать внешнюю компоненту в системе и даже устанавливать платформу на клиенте или на сервере 1С, т.к. все взаимодействие происходит посредством HTTP-запросов. REST API более простой для освоения, есть возможность отлаживать запросы посредством таких программ как ARC, Postman, Swagger, в то время как работу через COM можно отлаживать только в отладчике 1С.
2. В онлайн-режиме обмен данными производится в формате XML, например, на вход обработчика «ПолучитьДокумент» приходит строка в формате XML, содержащая информацию о текущей сессии Mobile Smarts, а методом «StorageConnector.ЗагрузитьОбъектИз_XML()» преобразуется в COM-объект «Cleverence.Warehouse.ServerSession», с которым можно работать, используя его программный интерфейс (например, GetDeviceInfo() и т.д.). Аналогично - используя метод «StorageConnector.СохранитьОбъектВ_XML()» мы преобразуем COM-объект «Cleverence.Warehouse.Document» в XML-строку и возвращаем результат из 1С на сервер Mobile SMARTS.

Не нашли что искали?



Задать вопрос в техническую поддержку

Методичка по созданию и обмену документами Mobile SMARTS при самостоятельной интеграции с учетной системой

Последние изменения: 2024-03-26

Возможные средства интеграции

При установке на ПК любой конфигурации Mobile SMARTS вы можете воспользоваться одним из четырех вариантов организации обмена данными между учетной системой пользователя и сервером платформы Mobile SMARTS.

К чему подключаемся



Выберите способ обмена с вашей товароучетной системой

- Обмен с базой данных «1С: Предприятие»
Вы можете обмениваться со своей базой на платформе «1С: Предприятие» при помощи обработок, которые позволяют получать данные из учетной системы и загружать проверенные на мобильном устройстве документы обратно в 1С. Список поддерживаемых конфигураций, а также возможность подключения к произвольной или нетиповой базе 1С - в следующем окне.
- Обмен данными через файлы Excel или CSV
Вы можете заполнять справочники системы, выгружать на устройство и обратно документы из файлов Excel или текстовых файлов в формате CSV.
- Обмен данными через собственную интеграцию на базе OLE/COM
Вы можете разработать интеграцию с продуктом, используя API на основе COM объектов.
- Обмен данными через REST API
Вы можете разработать интеграцию с продуктом, используя REST API.

Помощь

Назад

Далее

Прервать

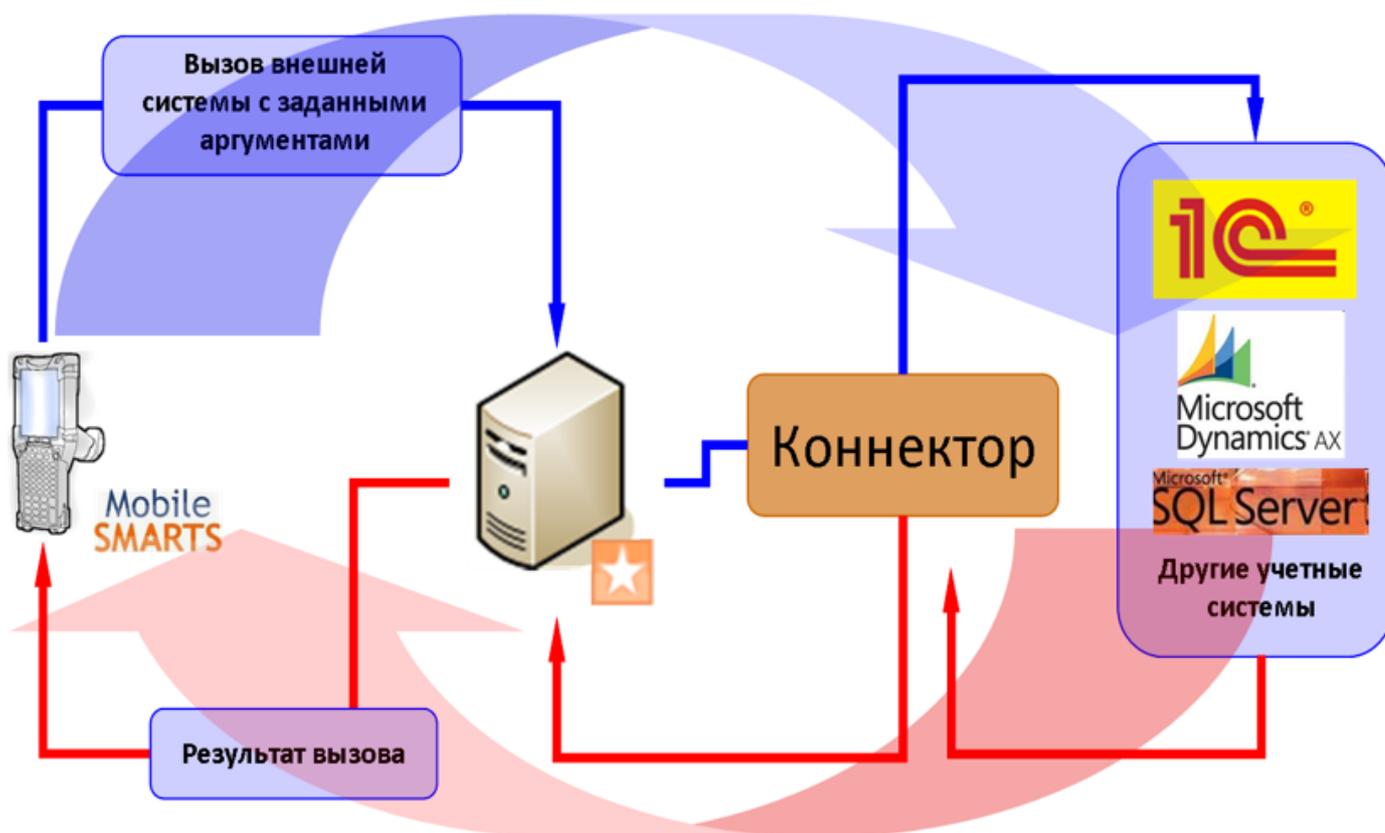
Рассмотрим отличия данных вариантов обмена и принципов их работы более подробно.

Обмен с базой данных «1С: Предприятия» и через собственную интеграцию на базе OLE/COM

Эти два варианта схожи по своей сути, т.к. для организации обмена данными между учетной системой пользователя и Mobile SMARTS применяются две специальные COM-компоненты. Установка и регистрация компонент выполняется автоматически при установке Mobile SMARTS из дистрибутива. В первом случае это COM-компонента, предназначена специально для работы с «1С:Предприятием» и некоторыми другими конфигурациями продуктов 1С (в зависимости от продукта). Во втором случае это неспециализированная

универсальная компонента, которая может быть использована с любой учетной системой. Подробнее об этом можно узнать [здесь](#).

В обоих случаях обращение к внешней системе происходит следующим образом:

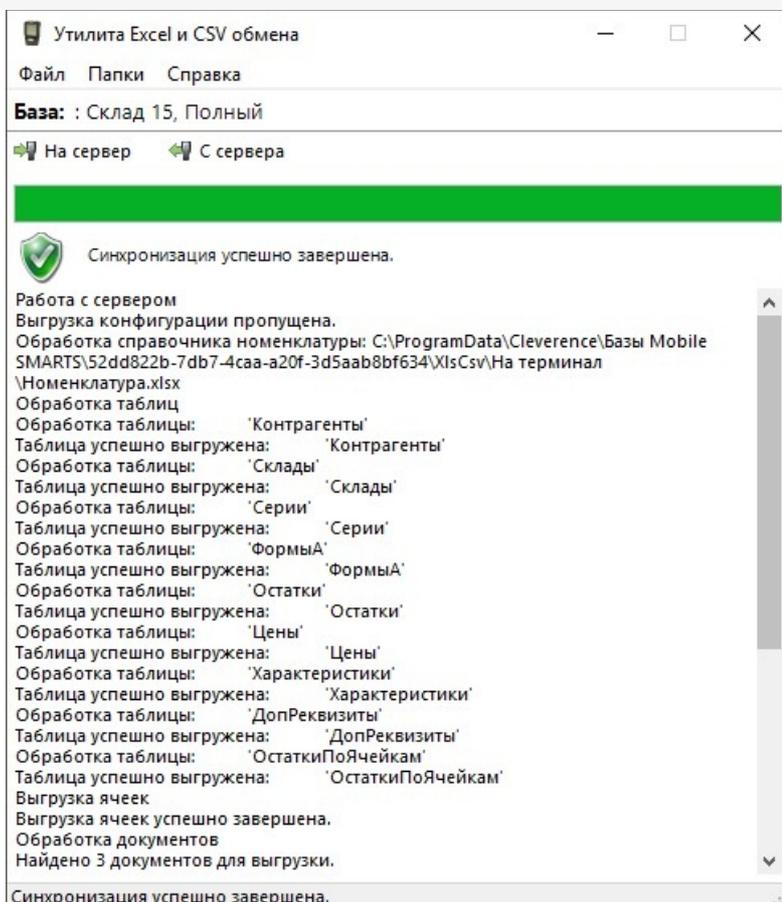


Обмен через Excel/ CSV файлы

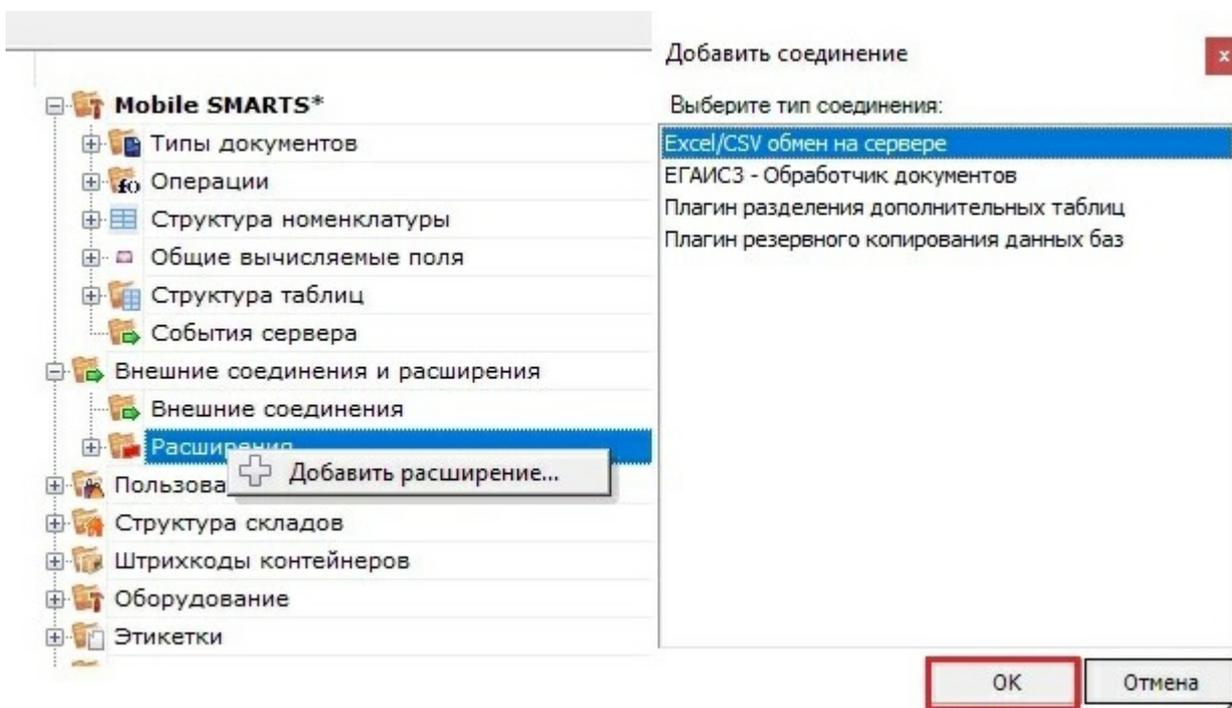
Помимо разнообразных вариантов интеграции с учётными системами, платформа Mobile SMARTS содержит в себе инструменты для интеграции на файлах CSV и Excel.

После установки платформы и развёртывания базы данных Mobile SMARTS становятся доступны следующие варианты обмена в формате CSV и Excel:

- [утилиты Excel и CSV обмена](#);



- использование специального [Excel/CSV коннектора](#).



В обоих случаях обмен происходит следующим образом:

1. Создаются файлы-шаблоны, содержащие заголовки полей, по которым будут разбираться файлы, содержащие данные. Эти файлы должны иметь такой же формат, в котором предполагается обмен.
2. В специальную папку обмена (по умолчанию «На терминал») помещаются файлы формата Excel или CSV, который содержат в себе номенклатуру, документы, таблицы и т. п. При этом структура этих файлов должна соответствовать созданному ранее шаблону.
3. Производится синхронизация с мобильным устройством или базой данных Mobile SMARTS (в зависимости от режима работы базы). При этом на устройство или сервер выгружается номенклатура и документы

из папки «На терминал», а обратно загружаются обработанные документы.

- Обработанные документы при загрузке тоже разбираются по соответствующим файлам-шаблонам и конвертируются в формат Excel или CSV.

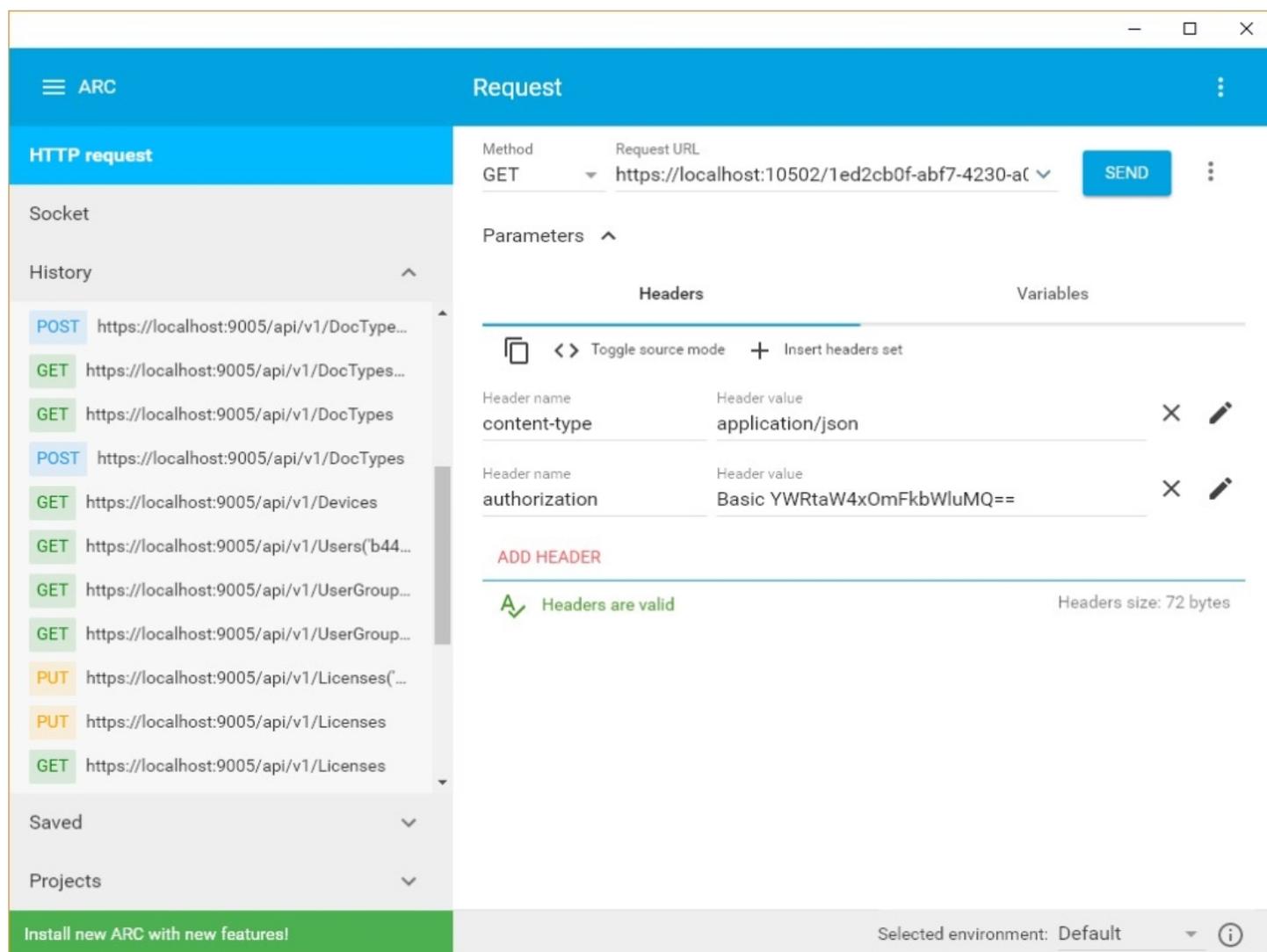
Подробнее о данном типе обмена можно узнать [здесь](#).

Обмен данными через REST API

API (Application Programming Interface) — интерфейс для программирования приложений, позволяет удаленно обращаться к функциям приложения и выполнять какие-либо действия в нем. Чаще всего используется для получения данных для мобильных приложений, интернет порталов и интеграции с внешними системами (например, интернет-магазинами).

В качестве API в нашей системе используется REST API на основе протокола OData, в котором собраны наиболее популярные техники работы с данными через HTTP запросы (GET, POST, PUT, PATCH). Более подробную информацию по работе с OData (4.0) можно найти по [ссылке](#).

Удобством использования REST является возможность проверки получаемых данных в обычном браузере, либо использованием стандартных приложений для выполнения HTTP запросов, например Advanced REST client (ARC).



В нашей системе также предусмотрена возможность просматривать структуру API и выполнять простые запросы, используя [Swagger](#).

Подробнее о данном методе обмена вы можете узнать по [ссылке](#).

Описание структуры документа и его основных переменных

В качестве примера будет рассматриваться документ «Приход на склад».

Документ «Приход на склад» используется в следующих бизнес-процессах:

- Приобретение товаров и услуг (проверка документа на ТСД)
- Приобретение товаров и услуг (на основании «Заказа поставщику»)
- Приходный ордер на товары (на основании «Заказа поставщику»)
- Приходный ордер на товары (на основании «Поступления товаров»)
- Приходный ордер на товары (проверка документа на ТСД)

Сам экземпляр документа «Приход на склад» физически представляет собой XML файл, хранящийся на сервере Mobile SMARTS, с данными в полях, определяемых конфигурацией сервера Mobile SMARTS для данного типа документа.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <Document xmlns:clr="http://schemas.cleverence.ru clr" barcode="123456789012398" createDate="2021-06-24T01:46:48.292
3   <Fields count="4">
4     <FieldValue fieldName="Поячейкам">
5       <Value clr:Type="Boolean">False</Value>
6     </FieldValue>
7     <FieldValue fieldName="КонтрольКолва">
8       <Value clr:Type="Boolean">False</Value>
9     </FieldValue>
10    <FieldValue fieldName="КонтрольЯчеек">
11      <Value clr:Type="Boolean">False</Value>
12    </FieldValue>
13    <FieldValue fieldName="ПоказыватьБП">
14      <Value clr:Type="String">Нет</Value>
15    </FieldValue>
16  </Fields>
17  <States count="1">
18    <DocumentState finished="False" finishedDate="2021-06-24T07:19:27.5984900-07:00" inProcess="True" inProcessD
19  </States>
20  <Tables/>
21  <DeclaredItems>
22    <DocumentItem uid="1c7ebef4-ea6a-4b57-a036-89855b3dd866" productId="697b6490-3453-4a86-b3a4-d05172213c81" pa
23    <Fields count="2">
24      <FieldValue fieldName="ИдЕдиницыИзмерения">
25        <Value clr:Type="String">266e3c86-0aa2-4291-8b9e-bde72f2b6809</Value>
26      </FieldValue>
27      <FieldValue fieldName="Цена">
28        <Value clr:Type="Decimal">22600</Value>
29      </FieldValue>
30    </Fields>
31  </DocumentItem>
32  <DocumentItem uid="06e08b48-550b-45cd-9dcf-16bae9f4fcfb" productId="f834dd58-13f0-4061-8b03-6f2d9f135cbf" pa
33  <Fields count="2">
34    <FieldValue fieldName="ИдЕдиницыИзмерения">
35      <Value clr:Type="String">db206d48-e210-4a6b-b80a-75b7c1ec6ad0</Value>
36    </FieldValue>
37    <FieldValue fieldName="Цена">
38      <Value clr:Type="Decimal">36900</Value>
39    </FieldValue>
40  </Fields>
41  </DocumentItem>
42  <DocumentItem uid="f8924e3e-bdf5-4929-b095-99be03455d5e" productId="549df872-379c-45bf-9b22-363cded95790" pa
43  <Fields count="2">
44    <FieldValue fieldName="ИдЕдиницыИзмерения">
45      <Value clr:Type="String">9a4426a0-2bfd-4098-a12e-1ed74748668a</Value>
46    </FieldValue>
47  </Fields>

```

В разрезе обмена данными, в случае файлового обмена, данный документ представляет собой файл в формате XLS, CSV, XML и т. д., который загружается/выгружается из учетной системы и обратно через папки обмена, находящиеся по пути «...\\XlsCsv» в корневой директории вашей базы. В общем случае для файлового обмена в XlsCsv предусмотрены следующие папки с указанным содержимым:

Имя папки/ подпапки
Описание

На терминал
Содержит файлы Excel и CSV, предназначенные для отправки на терминал.
На терминал/Архив
Архив успешно конвертированных файлов Excel и CSV. Если файл «пропал», его можно найти здесь.
С терминала
Сюда складываются файлы с терминала после конвертации их в Excel или CSV по шаблон.
Templates
Папка с файлами шаблонов конвертации. Для «Магазин 15» или «Склад 15» готовые шаблоны добавляются в папку автоматически при установке.
Templates/Upload
Содержит шаблоны, по которым разбираются файлы для терминала.
Templates/Download
Содержит шаблоны, по которым формируются готовые файлы с терминала. Для получения документов в определенном виде нужно положить сюда файл шаблона с именем типа документа, для которого предназначен шаблон.

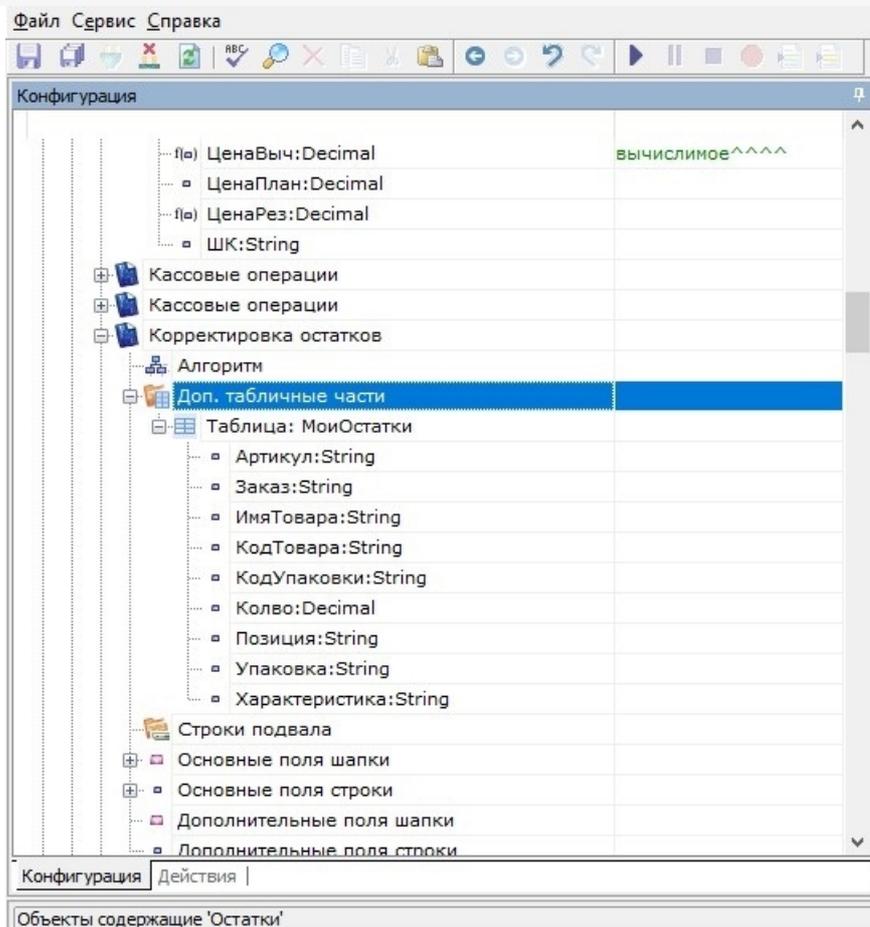
В случае использования других вариантов обмена, например COM-коннектора, данные папки и шаблоны не используются, так как данные передаются непосредственно в учетную систему, а шаблоны обмена данными и участвующие в ней поля определены в обработке/конфигурации со стороны самой учетной системы.

Структура и содержание документа «Приход на склад»

В общем смысле любой документ Mobile SMARTS содержит две основные логические части — «шапку» и «табличную часть» документа.

Первая, так называемая «шапка документа», содержит поля с общей информацией о самом документе и правилах его обработки. С точки зрения конфигурации Mobile SMARTS и программирования в ней — это набор полей данных, принадлежащих объекту {Document} и доступных в конфигурации через сам объект {Document} (например {Document.Name} и т. п.).

Вторая, «табличная» часть документа, содержит данные о самих товарах — декларативную (плановую) часть {Document.DeclaredItems} и текущую (фактическую) часть {Document.CurrentItems}, которые с точки зрения конфигурации и программирования, являются коллекцией специального типа данных для описания товаров — DocumentItemCollection и тоже принадлежат конкретному документу. Также табличная часть может содержать дополнительные произвольные табличные данные типа RowCollection и/или вычисляемые поля — это могут быть таблицы характеристик, остатков или серий товара, вычисляемые поля для отображения в списках и т. п.



Документ «Приход на склад» с точки зрения обмена данными

Обратите внимание, что при обмене данными между учётной системой и Mobile SMARTS используются далеко не все поля, которые существуют у данного типа документа в конфигурации Mobile SMARTS.

The screenshot displays a configuration window for a document type named «Приход на склад». The left pane shows a tree view of fields, including basic fields (Id, Name, Appointment, etc.) and additional fields (Additional fields). The right pane shows a script for the «Алгоритм» (Algorithm) section, which includes logic for filling default values, selecting warehouses, and setting document properties based on whether it was created on a PDA.

Additional fields configuration:

- НастройкиСохранены: Boolean (Указывает)
- ТабакКакОбычный: Boolean
- ПредставлениеИмениДокумента: String
- ВыбранныйБПнаТСД: String
- ПоказыватьБП: String
- ИмяБП: String
- КонтрольКолва: Boolean
- ИдКонтрагента: String
- Контрагент: Object
- ИдСклада1С: String
- Склад: Object
- КонтрагентВведенНаТСД: Boolean
- СкладВведенНаТСД: Boolean
- НастройкаБизнесПроцесса: String

Algorithm Script:

```

if Document.CreatedOnPDA
    f() ЗаполнениеПоУмолчанию (ВидЗаполнения, DocumentType =
    {Document.DocumentTypeName}, ИмяСклада = {GlobalVars.СкладПоУмолчанию
    [Document.DocumentTypeName] != "" ? GlobalVars.СкладПоУмолчанию
    [Document.DocumentTypeName]: GlobalVars.СкладПоУмолчанию["LastChanged"]},
    ИдСклада = {GlobalVars.ИдСкладаПоУмолчанию[Document.DocumentTypeName] !=
    "" ? GlobalVars.ИдСкладаПоУмолчанию
    [Document.DocumentTypeName]: GlobalVars.ИдСкладаПоУмолчанию
    ["LastChanged"]}, Зона = {GlobalVars.ИдЗоныПоУмолчанию
    [Document.DocumentTypeName] != "" ? GlobalVars.ИдЗоныПоУмолчанию
    [Document.DocumentTypeName]: GlobalVars.ИдЗоныПоУмолчанию
    ["LastChanged"]})

f() СкладыИКонтрагенты()
if Document.РаботаенСячейками == true
    if Document.Склад != null && Document.Склад.Адресный == true
        ПомещениеСклада = select first (*) from Склады where Item.ИдРодителя ==
        Document.ИдСклада1С
        if ПомещениеСклада != null
            Меню: [(Помещение != null: Выберите помещение...)]
            [Все равно продолжить]
        f() (Склад, ВыходПокнопке) = Выбор адресных складов (ItemsCollection =
        {Склады}, SelectedItem = {Document.Склад}, ТолькоОШК = {False}, ВыборПоШК
        = {True}, ТекстЗаголовка = <b color="Steelblue">Выбор склада</b>, ...)
        if Склад != null
            Document.ИдСклада1С = Склад.Ид;
            Document.ИмяСклада = Склад.Наименование;
            Document.СкладВведенНаТСД = true;
            if Document.CreatedOnPDA
                GlobalVars.ИдСкладаПоУмолчанию["LastChanged"] = Склад.Ид;
                GlobalVars.ИдСкладаПоУмолчанию[Document.DocumentTypeName] =
                Склад.Ид;
                GlobalVars.СкладПоУмолчанию["LastChanged"] = Склад.Наименование;
                GlobalVars.СкладПоУмолчанию[Document.DocumentTypeName] =
                Склад.Наименование;
                GlobalVars.Мод = GlobalVars.Мод == false;
            if Склад.Адресный == true
                if (Document.CreatedOnPDA == false) Document.ПоЯчейкам = True;
                else
                {
                GlobalVars.РежимРаботыПоЯчейкам[Document.DocumentType.Name] =
                "перед товаром";
                GlobalVars.РежимЯчейки[Document.DocumentType.Name] = true;
                GlobalVars.Мод = GlobalVars.Мод ? false: true;
                }
            if (Document.CreatedOnPDA == false) Document.ПоЯчейкам = False;

```

При этом в табличной части шапки документа типа «Приход на склад» при выгрузке из учетной системы присутствуют следующие поля документа:

Название поля
Расшифровка
Тип данных
{Name}
Название документа в учётной системе.
String
{Barcode}
Номер/Штрихкод документа
String
{ВыдаватьАвтоматически}
Автоматическое назначение документа
Boolean
{ПоЯчейкам}
Работа по ячейкам
Boolean

{ИсполняемыйНаСервере}
Коллективная работа с документом на сервере
Boolean

{КонтрольКолва}
Контроль планового количества товара
Boolean
{КонтрольЯчеек}
Контроль ячеек
Boolean
{РежимУпаковок}
Режим упаковок («коробки», «палеты», «коробки или палеты»)
String

При выгрузке в обратном направлении (из Mobile SMARTS в учетную систему) набор полей шапки меняется.

Название поля
Расшифровка
Тип данных
{Name}
Название документа в учётной системе.
String
{UserName}
Имя пользователя, завершившего документ
String
{DeviceId}
Идентификационный номер устройства, на котором был завершён документ
String
{DeviceIP}
IPv4/IPv6 адрес устройства на котором был завершён документ
String

Количество существующих и используемых в конфигурации полей намного больше, чем участвующих в обмене с учетной системой. Они нужны для обеспечения работы внутренней логики конфигурации Mobile SMARTS и их набор от конфигурации к конфигурации различается.

Во случае выгрузки из учетной системы на сервер Mobile SMARTS используются поля таблицы с товарной плановой частью документа {Document.DeclaredItems}. В случае выгрузки уже обработанного и завершённого документа из Mobile SMARTS в учётную систему используется тот же перечень полей, но со смешанными данными из плановой и фактической частей документа {Document.CombinedItems}. Ниже представлено описание полей документа «Приход на склад», участвующих в обмене с учетной системой (для стандартной конфигурации «Склада 15»).

Название поля
Смысл переменной
Тип данных
{НомерСтрокиДокумента}
№ строки
Int32
{ШК}
Штрихкод
String
{Product.Marking}
Артикул
String
{Product.Name}
Название
String
{ИмяХарактеристики}
Имя характеристики
String
{ИдХарактеристики}
Ид Характеристики
String
{ИдСерии}
Ид Серии
String
{ИдЕдиницыИзмерения}
Ид Единицы Измерения
String
{ИмяСерии}
Серия
String
{ИмяНеизвестного}
Описание
String
{DeclaredQuantity}
Заявленное количество
Decimal

{CurrentQuantity}
Принятое количество
Decimal
{ЦенаПриемки}
Цена приемки
Decimal
{АлкоНаим}
Название алко
String
{АлкоОбъем}
Объем
Decimal
{АлкоКрепость}
Крепость
Decimal
{Производитель}
Производитель
String
{ПроизвИНН}
ИНН производителя
String
{ПроизвКПП}
КПП производителя
String
{АлкоКод}
Алко код
String
{АлкоПДФ}
Поле для хранения PDF417 (существует только как синоним, второе наименование поля {Марка})
-
{АлкоСН}
Серийный номер алко
String
{СН}
Серийный номер
String

{ДатаРозлива}
Дата розлива
DateTime
{АлкоЕстьВФормах}
Наличие в «формах А» (имеется описание производителя/ импортера и диапазона серийных номеров акцизов)
Boolean
{КодФормыА}
Код «формы А»
String
{МаркаИСМП}
Марка ИСМП («01» + GTIN + «21» + CH)
String
{Марка}
Контрольная марка товара
String
{ИдКоробки}
Штрихкод коробки
String
{SSCC}
Штрихкод палеты
String
{FirstStorageId}
Ячейка хранения
String
{МаркаБлока}
Марка сигаретного блока
String
{МаркаСтандарт}
Марка в формате со скобками для управляющих кодов
String
{GS1Марка}
Марка в формате GS1
String
{ПолнаяМаркаHRI}
Марка в формате HRI
String

{ПолнаяМаркаBase64}
Марка в формате Base64
String

Также в конфигурации в документе «Приход на склад» присутствуют некоторые поля, не используемые при обмене, но необходимые для обеспечения работы внутренней логики конфигурации.

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Как отобразить данные из «1С: Предприятия» на мобильном устройстве без исправления кода основной обработки «Клеверенса»

Последние изменения: 2024-03-26

Клиенты «Клеверенса» часто обращаются в техническую поддержку с вопросами:

- можно ли отобразить дополнительную информацию о товаре на мобильном устройстве и как это сделать?
- как изменить определенный параметр сразу в нескольких местах?

Наш ответ — можно, и в этой статье мы подробно расскажем, как.

1. [Как отобразить дополнительную информацию о товаре на ТСД?](#)
 - [Краткая формулировка проблемы](#)
 - [Развернутое решение проблемы](#)
 - [Алгоритм решения проблемы](#)
 - [Действия в панели управления Mobile SMARTS](#)
 - [Действия в конфигураторе «1С: Предприятия»](#)
2. [Как изменить определенный параметр сразу в нескольких местах?](#)
 - [Решение проблемы](#)

Как отобразить дополнительную информацию о товаре на ТСД?

Проблема. Как показать и заполнить произвольное поле в структуру номенклатуры?

Краткий ответ. Для решения данной проблемы можно использовать различные варианты: например, добавить дополнительное поле номенклатуры товара в справочнике номенклатуры Mobile SMARTS.

Для выгрузки номенклатуры в Mobile SMARTS чаще всего используется учетная система «1С: Предприятие», в которой уже есть встроенные метаданные для большинства справочников и регистров продукта на платформе Mobile SMARTS (например, «Склад 15»). Поэтому для включения новых метаданных нужно только внести изменения в СКД (система компоновки данных).

Это делается один раз и для онлайн, и для офлайн сценариев интеграции Mobile SMARTS и 1С (т.е. в любом варианте работы используются одни и те же СКД).

Изменения вносятся в файл [интеграционной обработки](#), который по умолчанию расположен в папке базы продукта «\Обработки 1С\Интеграционные обработки».

Развернутое решение проблемы

Рассмотрим подробнее, как добавить новый реквизит в справочник номенклатуры и выгрузить его на ТСД.

Цель: выгрузить реквизит номенклатуры «ГоденДо» в поле «СрокГодности» и отобразить его на ТСД.

Исходные данные:

- в конфигурацию 1С добавлен реквизит номенклатуры «ГоденДо» (данные взяты в качестве примера, т.к. на практике срок годности может храниться, например, в регистре);
- в конфигурацию Mobile SMARTS добавлено дополнительное поле «СрокГодности» в структуре номенклатуры.

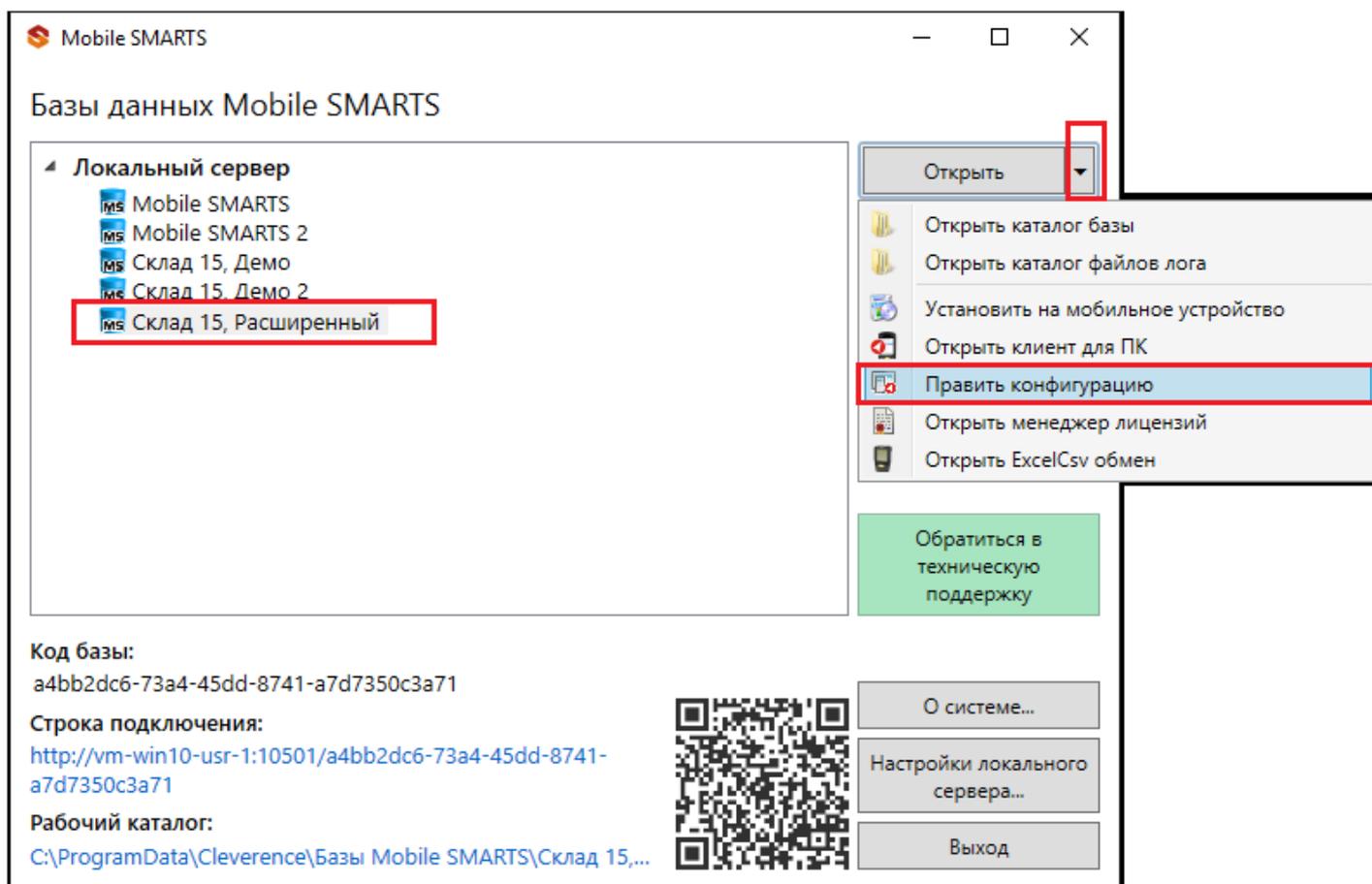
Далее необходимо открыть [основную обработку обмена КлеверенсТСД](#) в «1С Предприятие». Если при первом развертывании продукта Mobile SMARTS все настройки базы были распакованы правильно, то к основной обработке подцепится интеграционная соответствующей версии и конфигурации.

Если интеграционная обработка не подключена, не имеет поддержки текущей конфигурации или не поддерживает конкретную версию конфигурации, то потребуется [подключить её самостоятельно](#).

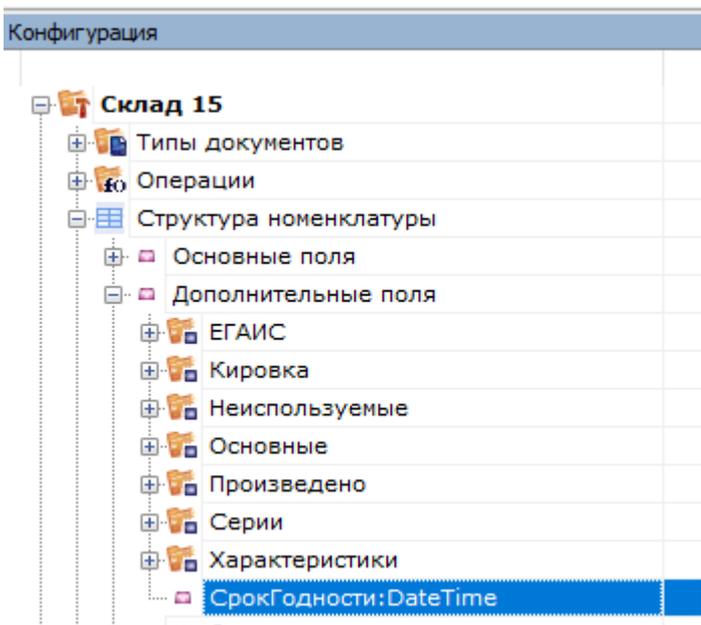
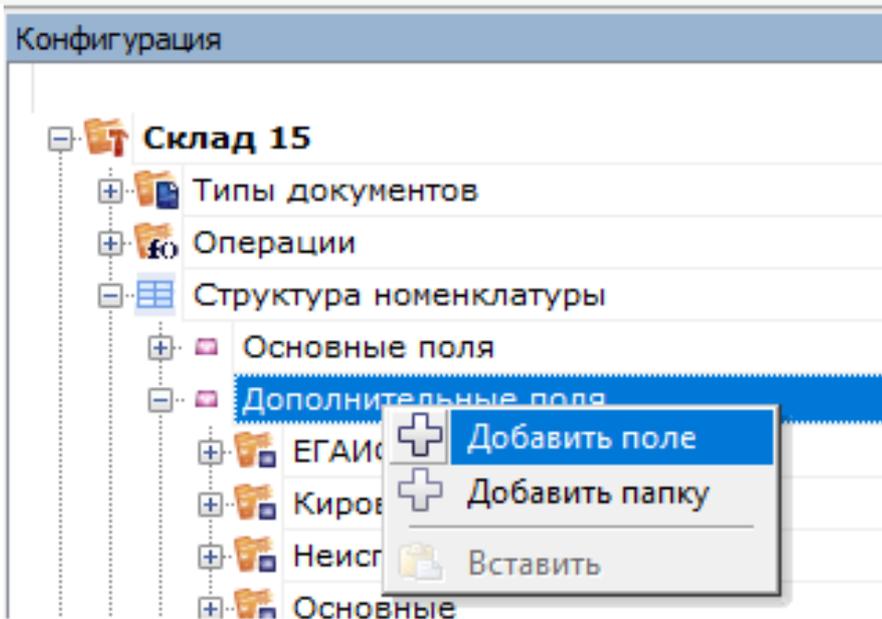
Алгоритм решения проблемы

Действия в панели управления Mobile SMARTS

1. Запустите [менеджер базы данных](#) и откройте [панель управления Mobile SMARTS](#) («Открыть» --> «Править конфигурацию»), которая позволяет редактировать конфигурацию (продукта Mobile SMARTS).



2. В левой части окна конфигурации разверните узел «Структура номенклатуры» --> «Дополнительные поля» --> правым щелчком мыши вызовите выпадающее меню --> «Добавить поле» --> введите наименование поля и его тип.



Имя поля должно в точности совпадать с именем, которое указано в шаблоне номенклатуры, включая регистр СИМВОЛОВ.

3. Сохраните изменения в конфигурации.

Mobile SMARTS - [Склад 15, Демо 4] - <http://win-3an96cq4nr.v>

Файл Сервис Справка



Конфигурация

Как вывести дополнительное поле в списке номенклатуры и выгрузить, например с помощью утилиты Excel/CSV описано в статье «[Добавление дополнительного поля номенклатуры Excel/CSV](#)».

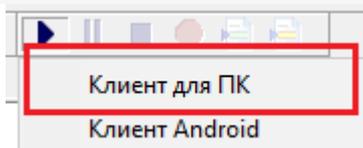
4. В конфигураторе отладки разверните узел «Данные» --> «Номенклатура» и запустите обновление измененных данных в номенклатуре. На экране появится выгруженная номенклатура, среди которой найдите нужный товар с добавленным полем и проверьте во вкладке «Поля» корректность его отображения.

The screenshot shows the configuration tool interface. On the left, the 'Склад 15' (Warehouse 15) tree is expanded to 'Номенклатура' (Inventory). The main window displays a list of inventory items. The selected item is 'Кондиционер ELEKTA' (Air conditioner ELEKTA) with ID 'dee6e1a4-55bc-11d9-848a...'. The 'Поля' (Fields) tab is open, showing a list of fields. The 'СрокГодности' (Expiration Date) field is highlighted. The 'Упаковки' (Packaging) tab is also visible, showing a list of packaging options.

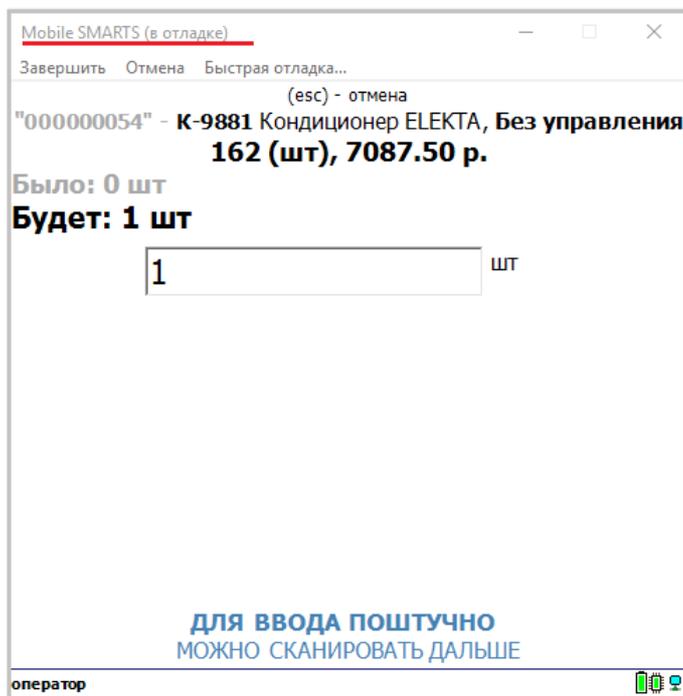
5. Запустите обмен с сервером на терминале сбора данных, откройте [клиент для ПК](#) в менеджере баз и проверьте точность отладки.

Отладка кода — это этап разработки, на котором обнаруживают, локализуют и устраняют ошибки. Для оперативного проведения отладки в Mobile SMARTS применяется [встроенный отладчик](#).

6. Далее можно проверить, как добавленное поле («СрокГодности») отображается на ТСД (например, в верхней части документа). Для этого, не закрывая выгруженную номенклатуру, запустите режим отладки в панели управления (пункт «Клиент для ПК» или кнопка F5).

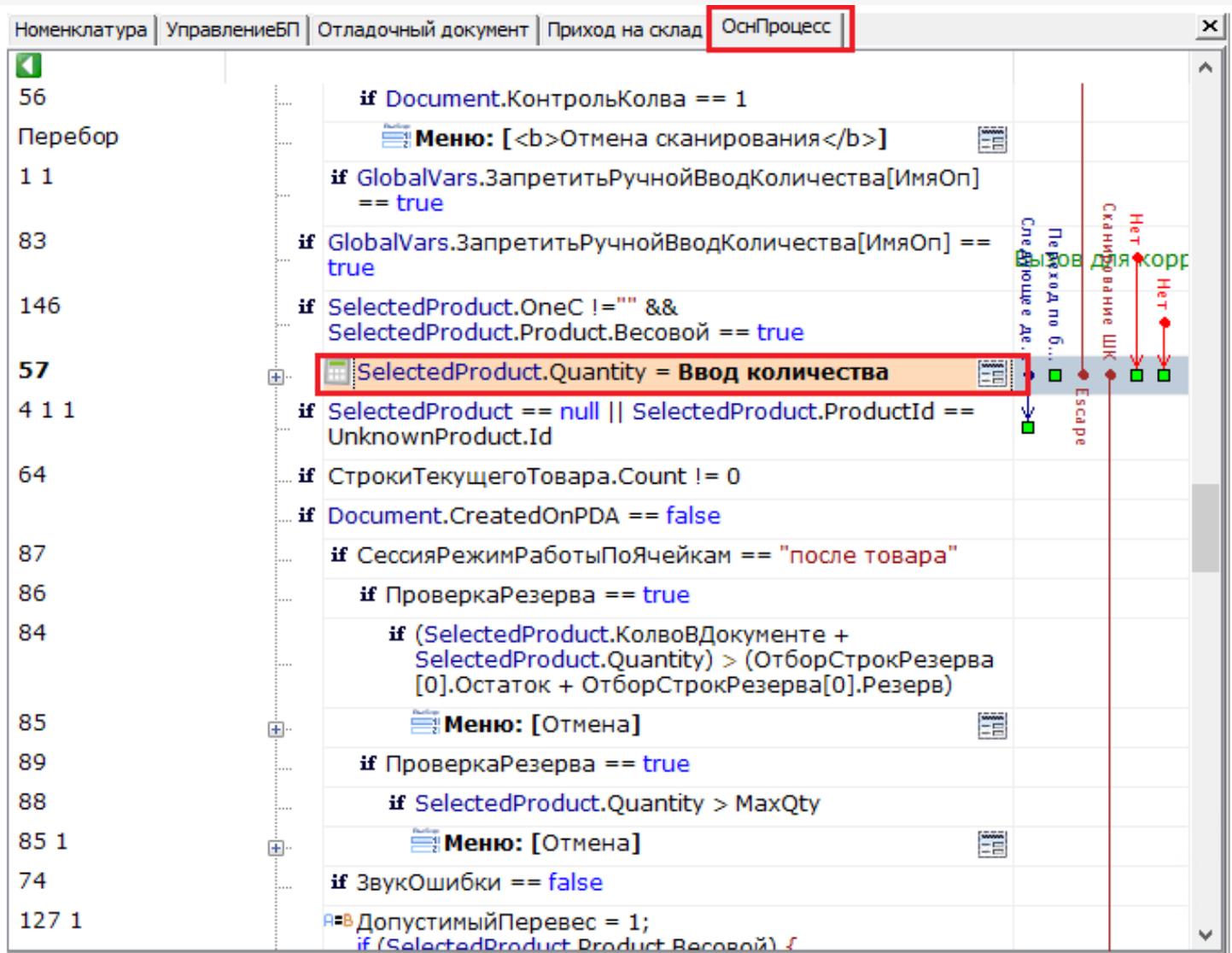


7. После запуска отладчика на экране ПК появится окно эмулятора ТСД с надписью в заголовке окна «В отладке». Рекомендуется использовать конфигурацию из панели управления, либо перед подключением отладчика нажать кнопку «Обмен с сервером» в клиенте Mobile SMARTS для ПК.

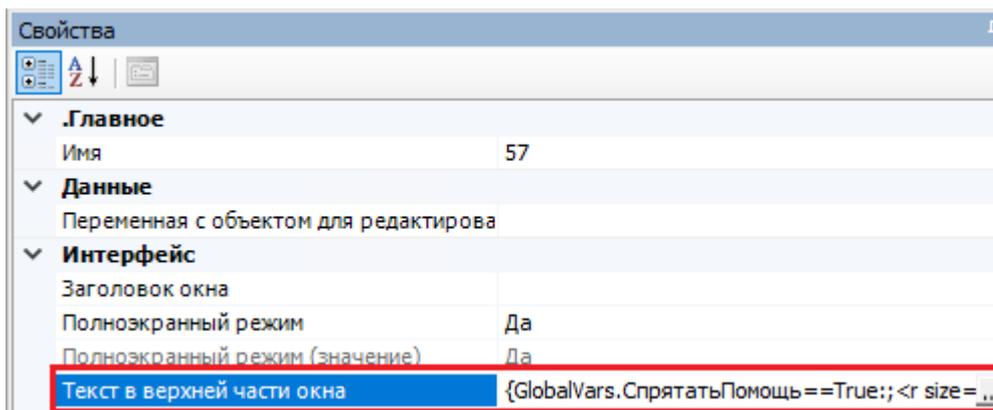


Во время выполнения операций на клиенте для ПК отладчик останавливается на действиях, требующих ввода данных. В панель управления подсвечивается красным тип документа, в котором ведётся работа, а жёлтым — действие, которое в данный момент выполняется в клиенте Mobile SMARTS для ПК.

8. Выберите элемент, который нужно изменить (в данном примере — это поле ввода количества) и внесите изменения в алгоритм конфигурации, чтобы в верхней части выводилось новое поле «СрокГодности» («ГоденДо»).



9. Во вкладке «Свойства» найдите пункт «Текст в верхней части окна», откройте его, нажав на кнопку с двумя точками, и задайте новое действие в алгоритме конфигурации.



10. Введите в поле редактирования следующую строку {SelectedProduct.Packing.СрокГодности} (для удобства ввода можно развернуть панель переменных внизу окна «Редактор строк»).

```
{GlobalVars.СпрятатьПомощь==True;;<r size="-2">(esc) - отмена</r>}
{SelectedProduct.стрВШапкахОкон}
{SelectedProduct.Packing.СрокГодности}
{SelectedProduct.Packing.ИмяСерии:<b size="-1" align="center" color="DarkSlateGray">Серия: (0
{SelectedProduct.ЦенаСкладВводКолва}
<b size="+2" align="left" color="DarkGray">Было: {SelectedProduct.КолвоВДокументе :0.####}
<b size="+4" align="left" color="Black">Будет: {SelectedProduct.КолвоВДокументе + CurrentActio
</b>
```

Шаблон: {SelectedProduct.Packing}

		{Document}	Текущий док
		{Products}	Объект для и
		{Buffer}	Буфер строк
		{ScannedBarcode}	Отсканирова
		{SSCCPallet}	Паллета (SSC
		{FirstStorage}	1-ое место хр
		{SecondStorage}	2-ое место хр
		{SelectedProduct}	Выбранная п
		{Product}	Продукт
		{Packing}	Упаковка

11. Нажмите «ОК», если все выполнено правильно — в верхней части окна отобразится новое значение («СрокГодности» --> «ГоденДо»). Далее потребуется изменить СКД в конфигураторе 1С.

Mobile SMARTS (в отладке)

— □ ×

Завершить Отмена Быстрая отладка...

(esc) - отмена

"2000000025865" - **К-9881 Кондиционер ELEKTA, Без
управления**

10.02.2022 0:00:00

162 (шт), 7087.50 р.

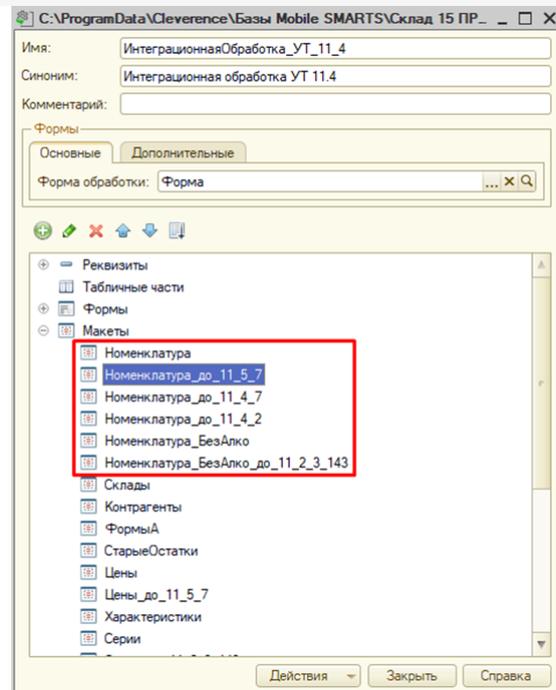
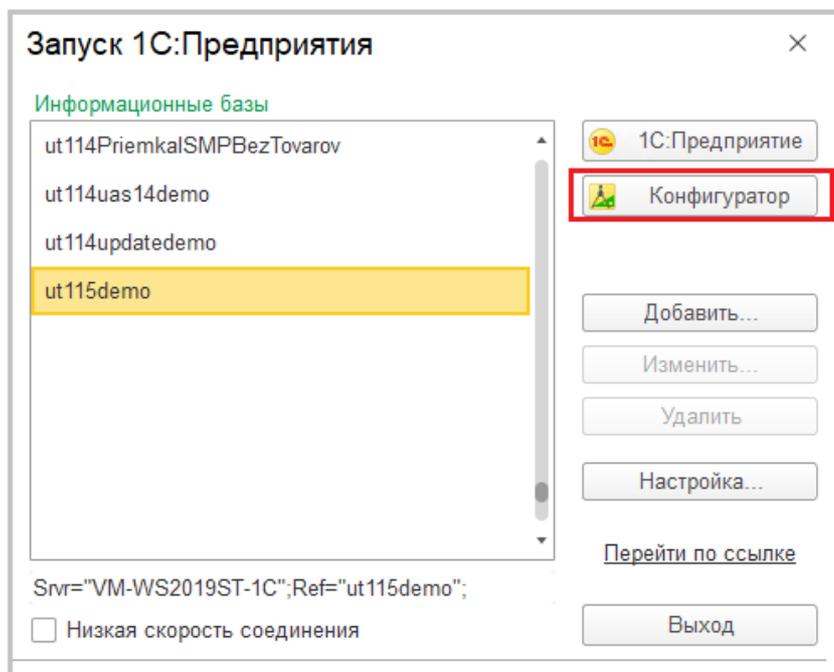
Было: 0 шт

Будет: 1 шт

шт

Действия в конфигураторе «1С: Предприятия»

1. Откройте интеграционную обработку в конфигураторе 1С и найдите макет, который используется для выгрузки номенклатуры.



2. Чтобы определить, какой макет используется для выгрузки номенклатуры, откройте окно «Обмен справочников и печати» на главной форме основной обработки и найдите имя макета в соответствующей строке.

← → Клевренс: Настройка обмена справочников

Сохранить и закрыть Сохранить Сохранить в файл... Восстановить из файла...

Выгружать номенклатуру документа при выгрузке документа

Разбить номенклатуру для выгрузки по: позиций (0 - не разбивать) ?

Установить поиск по наименованию и артикулу ?

Использовать шаблоны весового товара

Выбрать все Снять все Добавить Удалить Выгрузить выбранные Очистить выбранные

Выгружать	Наименование	Выгрузка	Предпросмотр и п...	Очистка справоч...	Способ выгрузки	Имя макета ИО	Прои
<input checked="" type="checkbox"/>	Номенклатура	Выгрузка...	Предпросмотр и п...	Очистка справоч...	По умолчанию	Номенклатура_до_11_5_7	
<input checked="" type="checkbox"/>	Характеристики номе...	Выгрузка...	Предпросмотр и п...	Очистка справоч...	По умолчанию	Характеристики	
<input checked="" type="checkbox"/>	Контрагенты	Выгрузка...	Предпросмотр и п...	Очистка справоч...	По умолчанию	Контрагенты	
<input checked="" type="checkbox"/>	Склады (места хран...	Выгрузка...	Предпросмотр и п...	Очистка справоч...	По умолчанию	Склады	

Этот способ не является универсальным, т.к. в некоторых случаях имя макета может программно подменяться в методе ПолучитьМакетСКД модуля интеграционной обработки. В данной статье по умолчанию считаем, что имя не подменено.

3. Откройте макет, который используется для выгрузки номенклатуры, и добавьте в текст запроса новое поле для выгрузки в Mobile SMARTS.

```

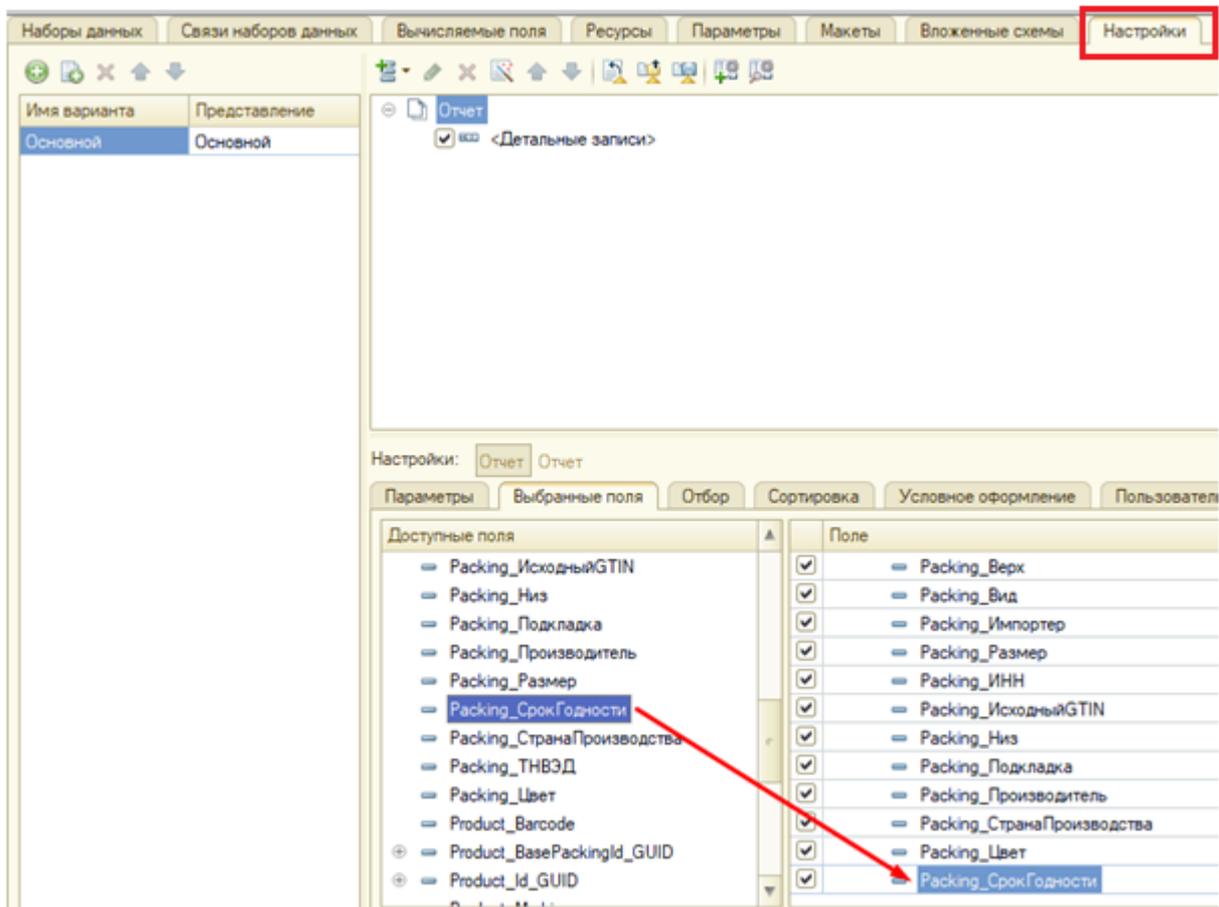
КОНЕЦ КАК ИспользоватьХарактеристики,
Номенклатура.ВидНоменклатуры.ИспользованиеХара:
ВЫБОР
    КОГДА Номенклатура.ВидНоменклатуры.Владелец
    ТОГДА Номенклатура.ВидНоменклатуры.Вла:
    ИНАЧЕ Номенклатура.ВладелецХарактеристик
КОНЕЦ КАК ВладелецХарактеристик
, Номенклатура.ГоденДо КАК ГоденДо
ПОМЕСТИТЬ ВТ_Номенклатура
ИЗ
    Справочник.Номенклатура КАК Номенклатура
ГДЕ
////////////////////////////////////
ВЫБРАТЬ РАЗРЕШЕННЫЕ РАЗЛИЧНЫЕ
ВТ_Номенклатура.Номенклатура                КАК Product_Id_GUID,
ВТ_Номенклатура.Наименование                КАК Product_Name,
ВТ_Номенклатура.Код                          КАК Product_Barcode,
ВТ_Номенклатура.ЭтоГруппа                    КАК Product_ЭтоГруппа,
ВТ_Номенклатура.Артикул                      КАК Product_Marking,
ВТ_Номенклатура.Весовой                      КАК Product_Весовой,
ВТ_Номенклатура.ЕдиницаИзмерения            КАК Product_BasePackingId_GUID,
ВТ_Номенклатура.Маркировка                  КАК Product_Маркировка,
...
""                                             КАК Packing_ИдСерии_GUID
, ВТ_Номенклатура.ГоденДо                    КАК Packing_СрокГодности
ИЗ
    ВТ_ШК_Упаковки КАК ВТ_ШК_Упаковки

```

Итоговое имя поля для таблицы номенклатуры должно иметь следующий вид: <Префикс>_<ИмяПоляMobileSMARTS>, где <Префикс> — это Product или Packing, в зависимости от того, будет относиться новое поле к позиции номенклатуры в целом или к ее отдельным упаковкам.

Объектом, представляющим товар в Mobile SMARTS является Product (Товар), каждый товар содержит коллекцию упаковок Packing (Упаковка).

4. Добавьте новое поле в состав выбранных полей схемы компоновки данных на вкладке «Настройки».



5. Сохраните интеграционную обработку и перевыберите ее в основной обработке.

← → **Клеверенс: Склад 15, Расширенный**

Торговый объект: ×

НАСТРОЙКИ

Выгрузить справочники

Открыть окно обмена справочников и печати

Открыть окно обмена документами

Скрыть настройки

[Информация по работе с обработкой](#)

Подключенная база Mobile SMARTS:
Склад 15, Расширенный (master УТ115), версия: 1.3.2.242

Настроить или изменить подключение к базе Mobile SMARTS подключения к базе Mobile SMARTS. Подключенная база Mobile SMARTS обязательный элемент для работы текущей обработки.

Путь к обработке интеграции:
Справочник: "Дополнительные отчеты и обработки", Обработка

Выбрать интеграционную обработку для её использования в выбранной базе Mobile SMARTS.

[Ссылка на базу знаний](#)

Если используется обработка из файла на диске, то перевыберите файл как показано на скриншоте ниже. Если используется обработка из справочника дополнительных отчетов и обработок, воспользуйтесь одним из способов, описанных в [инструкции](#).

6. Если вы работаете в режиме оффлайн, [выгрузите справочник номенклатуры](#) на устройство заново.

Операция вывода номенклатуры товара в справочнике номенклатуры Mobile SMARTS прошла успешно!

Как изменить определенный параметр сразу в нескольких местах?

Проблема. У каждого типа данных в Mobile SMARTS есть собственный набор полей, но что если нужно изменить определенный параметр сразу в нескольких местах номенклатуры?

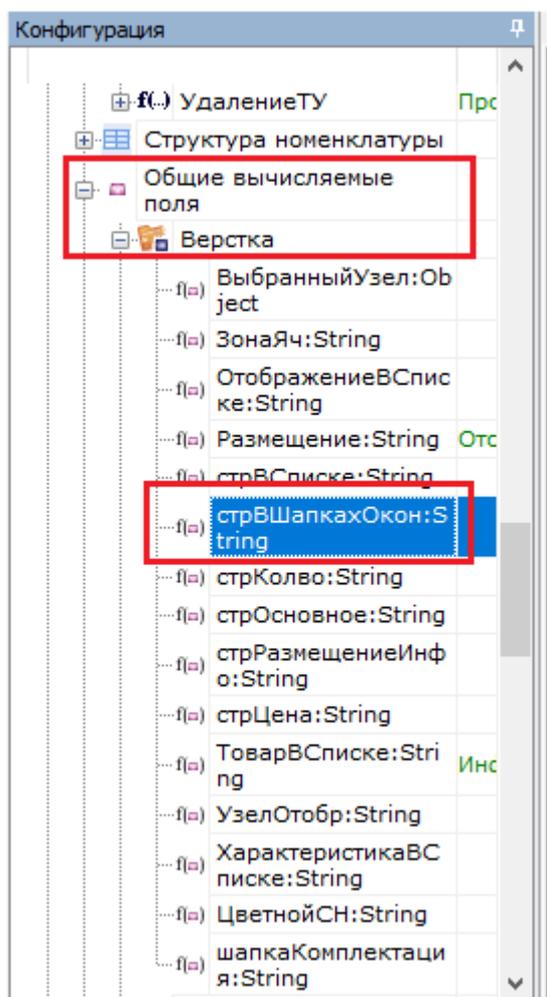
Решение проблемы

Не нужно заходить в каждый параметр и править его отдельно, т.к. в Mobile SMARTS есть возможность добавить/исправить общее вычисляемое поле сразу для всех типов данных.

В данном примере, чтобы изменения отображались в вычисляемых шаблонах в верхнем поле или других строках ТСД, нужно добавить новое значение в общем вычисляемом поле {SelectedProduct.стрВШапакахОкон}.

Общие вычисляемые поля представляют собой дополнительные поля, которые появляются сразу у всех объектов (типов данных), и значения которых задаются шаблонами.

1. Для этого в конфигураторе отладки разверните узел «Общие вычисляемые поля» --> «Верстка» --> «стрВШапкахОконНоменклатура».



2. Далее в окне «Свойства» выберите пункт «Шаблон значения» как показано ниже:

Свойства	
.Главное	
Uni	strVShapkahOkon
Общее	
Имя поля	стрВШапкахОкон
Синоним	
Тип поля	String
Шаблон значения	<b color="DarkGray">"{Item.Маркировка == "Обувь"
Шаблон значения вычисляется	Каждый раз
Поиск	
Индексировать для поиска	Нет
Индексировать для поиска по части строки	Нет
Уникальный ключ	Нет
Прочее	
ЕапAI	None
Комментарий	

3. Добавьте в редакторе строк новое значение {Item.Packing.СрокГодности}.

```
<b color="DarkGray">"{Item.Маркировка == "Обувь" && Item.GS1Марка! = "";  
{GO.GetBarcodeData(Item.GS1Марка).GS1.GTIN: <b size="-1">(0)</b> - };{Item.ШК: <b size="-1">  
(0)" </b> - } }</b>{Item.стрОсновное}{Item.CH: , <b>№(0)</b>}  
{Item.Packing.СрокГодности}
```

4. Сохраните отладку и заново запустите эмулятор ТСД.

По данному алгоритму в дальнейшем можно менять конфигурации вывода полей в разных частях окон на ТСД.

Не нашли что искали?



Задать вопрос в техническую поддержку

Ядро и формы обработки 1С

Последние изменения: 2024-03-26

Ядро

Ядро обработки представляет собой программный интерфейс для взаимодействия между товарно-учетными системами на базе 1С и программами «Магазин 15»/ «Склад 15» на платформе Mobile SMARTS.

Ядро полностью отвязано (абстрагировано, независимо) от конкретной конфигурации 1С. Для интеграции с конкретной конфигурацией предназначена отдельно поставляемая интеграционная обработка, содержащая в себе запросы для [выгрузки справочников](#) и обработчики [загрузки документов](#) из Mobile SMARTS в 1С.

Предполагается, что вносить изменения в ядро могут ТОЛЬКО разработчики «Клеверенс». Для интеграции ядра с конфигурацией 1С нужно вносить изменения в [интеграционную обработку](#). Поэтому всегда используется одна и та же версия ядра, а разница лишь в интеграционных обработках и настройках Mobile SMARTS. Аналогично у платформы и конфигураций «1С: Предприятия» — платформа 1С предоставляет интерфейс для разработки конфигураций и закрыта от изменений разработчиками.

Весь функционал ядра разбит на глобальное ядро/ контекст (исполняется на сервере 1С) и локальное ядро/ контекст (исполняется на тонком клиенте):

- **Глобальное ядро (контекст).** Это модуль обработки, реквизиты и табличные части обработки. Здесь хранится основная часть ядра – механизмы работы с [базой Mobile SMARTS](#), механизмы работы с настройками, универсальные и вспомогательные механизмы работы с данными. При работе в онлайн-режиме здесь же (в реквизитах) хранятся текущие настройки системы.
- **Локальное ядро (контекст).** Специально созданная форма обработки, в которой хранится мини-копия ядра – экспортные методы и функции, отвечающие ТОЛЬКО за работу с компонентой (обращение к методам [COM-объекта Mobile SMARTS](#)).

Для каждого из контекстов существует кэш — структуры данных, которые при запуске ядро получает из баз 1С и Mobile SMARTS, необходимые для дальнейшей работы ядра.

Локальное ядро необходимо только в случаях, когда сервер Mobile SMARTS (COM-компонента) установлен на локальной машине (там же, откуда запускается 1С). В иных случаях (файловая база 1С, сервер Mobile SMARTS установлен на сервере 1С) — используется глобальное ядро.

Техническая реализация данного механизма проста — в реквизитах объекта и в КАЖДОЙ форме создаются обязательные реквизиты «ЯдроОбщее» и «ЯдроКомпоненты» произвольного типа. Данные реквизиты содержат «транзитные» ссылки на используемый контекст.

- Реквизит «ЯдроОбщее» всегда содержит в себе значение «ОбработкаОбъект» для доступа к глобальному контексту.
- Реквизит «ЯдроКомпоненты» содержит в себе либо значение «ОбработкаОбъект» (в случае, если компонента установлена на сервере 1С), либо ссылку на форму с мини-копией ядра (в случае использования клиент-серверного варианта работы и компоненты, установленной на локальной машине).

Для исключения «перекрестного» обращения к контекстам в связи с идентичными именами реквизитов, содержащих в себе ссылки на контексты, жестко регламентируется обращение к ним:

- Из модуля объекта обращение к ядру производится только через предикат «ЭтотОбъект.»: «ЭтотОбъект.ЯдроОбщее» или «ЭтотОбъект.ЯдроКомпоненты».

- Из всех остальных модулей форм обращение производится только через предикат «ЭтаФорма.»: «ЭтаФорма.ЯдроОбщее» или «ЭтаФорма.ЯдроКомпоненты».

Процедуры и функции мини-ядра клиентского контекста полностью идентичны аналогичным процедурам и функциям серверного контекста (можно их просто копипастить). Разница будет только в их вызове, например «ЭтотОбъект.ЯдроОбщее.ПолучитьНастройку(ИмяНастройки)» или «ЭтаФорма.ЯдроКомпоненты.СохранитьДокументНаСервереSMARTS()».

Все процессы передачи данных между глобальным контекстом и остальными субъектами (формами и даже локальным контекстом, вне зависимости от месторасположения последнего) производятся строго через механизм временного хранилища 1С.

Порядок инициализации контекстов:

1. При запуске системы из модуля объекта инициализируется глобальный контекст «ЭтотОбъект.ЯдроОбщее = ЭтотОбъект»
2. Считывается настройка «Место установки сервера SMARTS».

Если место установки компоненты – на сервере, тогда инициализируется локальный контекст «ЭтотОбъект.ЯдроКомпоненты = ЭтотОбъект».

Если место установки компоненты – на клиенте, тогда локальный контекст не инициализируется, и будет инициализирован только при открытии главной формы обработки.

3. Если локальный контекст инициализирован – в кэш глобального контекста считываются все настройки из базы Mobile SMARTS («ЯдроОбщее.Настройки = ЯдроКомпоненты.ИнициализироватьНастройки»).
4. В случае работы в онлайн-режиме инициализация на этом заканчивается, ибо онлайн-режим может работать только в случае, когда компонента установлена на сервере (либо использование файлового варианта). В противном случае, при батч-режиме или при режиме автообмена – инициализация продолжается.
5. При открытии главной формы обработки мы повторно инициализируем глобальный контекст, но уже применительно к реквизитам формы – «ЭтаФорма.ЯдроОбщее = ЭтотОбъект».
6. Считывается настройка «Место установки сервера Mobile SMARTS».

Если место установки компоненты – на сервере, тогда инициализируется локальный контекст «ЭтаФорма.ЯдроКомпоненты = ЭтотОбъект».

Если место установки компоненты – на клиенте, тогда локальный контекст инициализируется как «ЭтаФорма.ЯдроКомпоненты = ПолучитьФорму(«ФормаЛокальногоКонтекста»)».

Реквизиты форм «ЯдроОбщее» и «ЯдроКомпоненты» являются наследуемыми, и должны всегда передаваться в подчиненную форму при ее открытии (например, при открытии формы обмена документами из главной формы обработки).

Локальная обработка не имеет доступа к глобальной, соответственно, все необходимые данные настроек (из глобальной обработки) для работы методов локального контекста должны передаваться непосредственно при их вызове.

Формы

Формы не являются непосредственной частью ядра. Они являются пользовательским интерфейсом для использования 3 главных функций ядра: работа с настройками, обмен справочниками, обмен документами.

Формы делятся на:

- **ключевые формы** — главная форма, форма обмена документами, форма списка отборов справочников, локальный контекст
- **формы списков** (для различных таблиц, т.е. сериализуемых настроек) — узлы, бп, отборы документов, глобальные параметры, произвольные коды, произвольные отборы документов. Содержимое этих форм хранится в настройках базы MS, далее, при старте обработке, помещается в кэш и хранится там до завершения работы обработки. Эти формы доступны только для просмотра (за исключением формы списка отборов справочников) + опционально можно экспортировать/импортировать содержимое список во внешние файлы.
- **формы настроек** (формы карточек) — узла, бп, отбора справочников, отбора документов, правила выгрузки полей, правила загрузки полей, правила выгрузки ТЧ, правила загрузки ТЧ, связи выгрузки ТЧ, произвольный код, произвольный отбор документов). При открытии в данные формы передаётся содержимое текущей строки того списка, из которого она была открыта. Форма позволяет редактировать данные и сохранять их.
- **формы просмотра** (ReadOnly) — документа MS и шапки/строки документа MS
- **служебные формы** — выбор строки из таблицы, загрузки новых ШК, настройки подключения, выбор документа для загрузки в 1С, выбор пользователя или группы, редактирования текста запроса

Формы обработки НЕ ДОЛЖНЫ содержать никаких методов работы с объектными данными 1С или Mobile SMARTS. Вся работа с данными 1С происходит в глобальной обработке, а с данными Mobile SMARTS – в локальной обработке. Все промежуточные методы подготовки, обработки и прочего с полученными данными – находятся только в глобальной обработке.

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

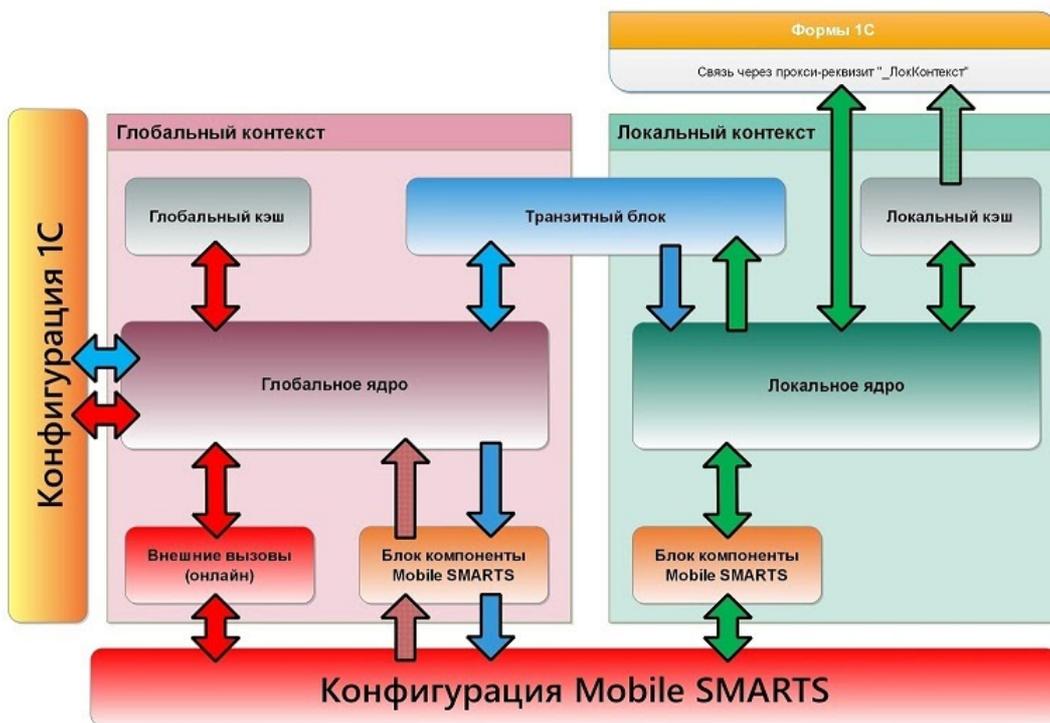
Функционал ядра обработки 1С

Последние изменения: 2024-03-26

Функционал ядра разбит на 3 крупных блока:

1. Работа с настройками обмена (**настройки бизнес-процессов, торговых объектов, глобальных параметров, отборов документов и справочников** в разрезе торговых объектов и типов обмена и т.д.).
2. **Выгрузка справочников** в режимах онлайн, офлайн и по регламенту согласно настройкам обмена.
3. **Выгрузка/загрузка документов** из 1С в Mobile SMARTS согласно настройкам обмена.

Все остальные блоки являются вспомогательными для работы с этими 3 блоками.



1. **Транзитный блок** — расположен на клиенте, нужен для передачи управления на сервер (при необходимости), параметры и результат функции передаются через временное хранилище
2. **Блок сохранения/восстановления настроек подключения и параметров текущего узла в хранилище общих настроек (далее - ХОН)** — доступно только на сервере, с клиента доступны через транзитный блок.
3. **Программный интерфейс для работы напрямую с MS (API)** — полностью дублируются на клиенте и на сервере, т.к. **база Mobile SMARTS** может быть зарегистрирована на клиенте либо на сервере 1С. Фактически, это обмен данными с Mobile SMARTS непосредственно через **COM-объекты** и вызовы их методов: терминал коннектор, база, настройки, сторадж коннектор, среда.
 - сохранение/восстановление настроек, удаление настроек, получение настроек по умолчанию;
 - выгрузка справочников (номенклатуры и доп.таблиц);
 - обмен документами - выгрузка, загрузка, удаление, получение списков документов;
 - служебные - получение списка баз, базы по ид, настроек базы, получение текста ошибки, метаданных документов, пользователей, шаблонов **весового товара**.
4. **Клиентские и серверные обёртки для работы с настройками MS:**

- вызывают соответствующие методы API MS (инициализация кэшей, чтение, установка настроек, запись/чтение из кэша, получение настроек по умолчанию);
- вспомогательные механизмы;
- **ТОЛЬКО КЛИЕНТСКИЕ МЕТОДЫ:** сохранения/восстановления из файла таблиц из форм список, либо отдельных элементов таблиц из форм настроек.

5. Клиентские и серверные методы выгрузки справочников и номенклатуры:

- методы для получения данных из 1С есть **ТОЛЬКО НА СЕРВЕРЕ**, т.к. только оттуда можно получить данные из базы, поэтому они вызываются с клиента через транзитный блок;
- также методы для формирования текстов запроса, работы с СКД, постобработка таблиц данных выполняются **ТОЛЬКО НА СЕРВЕРЕ**;
- методы для взаимодействия с API MS одинаковые на клиенте и сервере.

6. Работа с документами 1С — аналогично п.5, получение документов из 1С, запись в 1С, получение списка документов и т.д. выполняется **ТОЛЬКО НА СЕРВЕРЕ**, с клиента вызываются через транзитный блок

7. Внешние вызовы — выполняются только на сервере или внешнем соединении, нужны для работы ТСД в онлайн-режиме. В свою очередь, решают следующие задачи:

- получение актуальных данных справочников из 1С согласно настройкам отборов,
- получение списка документов 1С согласно настройкам отборов,
- выгрузка документа 1С в Mobile SMARTS для его набора на терминале,
- загрузка (завершение) документа из Mobile SMARTS в 1С.

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Хранение объектов и переменных на клиенте, на сервере и на форме

Последние изменения: 2024-03-26

Переменные ядра

Переменные ядра хранят данные, к которым требуется часто обращаться во многих методах ядра (фактически, это кэш). Необходимы для быстрого доступа к часто используемым данным, чтобы не получать их из [баз Mobile SMARTS](#)/ 1С при каждом обращении к ним.

Все данные, которые помещаются в переменные ядра при инициализации обработки (интерактивно или онлайн) откуда-то загружаются — из метаданных базы Mobile SMARTS, настроек базы Mobile SMARTS, из хранилища общих настроек 1С (ХОН), из метаданных конфигурации 1С или создаются программно и заполняются постоянными значениями-константами.

Для локального ядра это:

- клиентские переменные формы «ЛокальныйКонтекст», которые доступны в любых, но только клиентских методах формы
- реквизиты формы «ЛокальныйКонтекст», которые доступны в любых и клиентских, и серверных методах формы.

Имя переменной	Для чего нужна
Откуда загружается	
Может ли изменяться	
Клиентские переменные формы «Локальный контекст»	
РасположениеКомпонент	
Тип — Строка, значения — «НаКлиенте» или «НаСервере». Означает, что с клиента 1С или с сервера 1С будет осуществляться взаимодействие с базой Mobile SMARTS . Для обмена через COM обязательно наличие зарегистрированной внешней компоненты на соответствующей стороне (клиент/сервер). Для REST API ничего дополнительно делать не нужно.	
ХОН	
Да, в форме настройки подключения	

ДанныеТекущегоУзла

Тип — Структура. Содержит в себе

ID узла, наименование (например, «Магазин на Ленина») и ID базы Mobile SMARTS, которая привязана к этому узлу. При этом, к узлу может быть привязана либо **частная база**, либо основная база. Необходимо для случая, когда к текущей базе

1С привязан определенный узел (торговый объект – магазин, склад) и необходимо осуществлять обмен документами и справочниками

не с основной базой Mobile SMARTS, а с частной. В этом случае, в основной базе Mobile SMARTS хранятся только настройки, а справочники и документы – в частной базе.

Настройки базы Mobile SMARTS

Да, на главной форме при выборе узла (торгового объекта) Категории НастроекMS

КатегорииНастроекMS

Тип – Структура. Содержит метаданные о настройках, сгруппированные по определенным критериям:

- **сериализуемые настройки** – это настройки, которые являются таблицами. Нужны только для того чтобы знать, нужно ли сериализовывать/десериализовать при экспорте/импорте настроек в файл/из файла
- **узловые настройки** – это **настройки, которые можно сохранять в разрезе узлов (магазинов, складов)**, например, настройки обмена справочниками – т.к. для каждого узла может быть необходимость выгружать только «свою» номенклатуру, склады, остатки и т.д.
- **настройки подключения** – нужны, в первую очередь, для **подключения к базе Mobile SMARTS** – ID базы, имя базы, строка подключения, имя ИО, ID текущего узла, данные о конфигурации 1С
- **настройки справочников** – хранит флажки, означающие, какие справочники нужно выгружать – номенклатура, склады, остатки и т.д.
- **настройки номенклатуры** – хранит настройки выгрузки справочника «номенклатура» – использовать шаблоны весового товара, префикс кода весового товара, длина кода весового товара, переписывать номенклатуру при выгрузке и т.д.

Программно

Нет

НастройкиИнициализированы

Тип – Булево. Означает, что настройки успешно загружены из базы Mobile SMARTS. Если флаг = Ложь, значит, необходимо перезагрузить настройки из базы

Программно

Да, если потеряна связь с сервером MS

ПараметрыПодключенияMS
Тип – Структура. Хранит настройки подключения к базе Mobile SMARTS — ID базы, Имя базы, ID приложения, Строка подключения, QR, Логин, Токен, имя ИО, путь ИО, тип ИО, тип установленного продукта, уровень приложения
ХОН
Да, в форме настройки подключения
СтруктураМетаданных
Тип – Структура. Хранит метаданные документов 1С, метаданные документов MS, пользователи MS, устройства MS, таблицы MS, УниПолейНоменклатуры
Метаданные базы 1С, метаданные базы MS, Настройки базы MS
Нет
СОМЧастнойБазыMS
Тип – Структура. Хранит СОМ-объекты для работы с частной базой – TerminalConnector и StorageConnector. Заполняется, только если подключение к базе Mobile SMARTS осуществляется через СОМ. Для REST API не заполняется.
Программно
Да, если изменим базу, к которой будем подключаться для обмена
СОМВременнойБазыMS
Аналогично СОМЧастнойБазыMS, но используется только в форме подключения для проверки соединения, чтобы не затирать соединение с существующими базами
Программно
Да, т.к. служебная переменная
СОМБазыMS
Аналогично СОМЧастнойБазыMS, но используется для взаимодействия с основной базой MS
Программно
Да, если изменим базу, к которой будем подключаться для обмена
ТекущиеНастройкиMS
Тип – Структура. Содержит все настройки текущей базы MS в виде пар Ключ-Значение, например, ВыгружатьНоменклатуруСразу = Истина и т.д.
Настройки базы MS
Да, из всех форм, где могут редактироваться настройки
ПутьКОбработке
НЕ ИСПОЛЬЗУЕТСЯ!!! Путь хранится в переменной ядра ПараметрыПодключенияMS. ПутьИнтеграционнойОбработки
—
—

КОМсоздан
Тип – Булево. Означает, что успешно созданы COM-объекты для работы с базами MS – TerminalConnector и StorageConnector и помещены в структуры COMЧастнойБазыMS/ COMБазыMS
Программно
Да, если потеряна связь с сервером MS
ПраваПользователя
Тип – Строка. Хранит роль пользователя в 1С – «Администратор», «Пользователь», влияет на отображение некоторых кнопок на главной форме – кнопки настройки, кнопки выбора ИО
Метаданные базы 1С
Нет
ФормаИнтеграционнойОбработки
Тип – ФормаКлиентскогоПриложения. Используется для вызова клиентского обработчика «После открытия формы документа 1С»
Программно
Да, в форме настройки ИО
Реквизиты формы «Локальный контекст»
_ИнтеграционнаяОбработка
Адрес
Тип – Строка. Хранит адрес интеграционной обработки во временном хранилище. Впоследствии, интеграционная обработка извлекается из временного хранилища на стороне сервера и из нее вызываются экспортные методы – обработчики загрузки документа и т.д.
Программно
Да, в форме настройки ИО

АдресаПеременных

Тип – Структура. Хранит в себе адреса во временном хранилище других переменных ядра. При изменении значений в самих переменных ядра параллельно изменяются эти же значения и во временном хранилище. Эту структуру можно передавать в глобальное ядро и оттуда иметь доступ практически ко всем переменным локального ядра.

Состав структуры, жирным и подчеркнутым выделены переменные, хранящие адреса аналогичных переменных ядра:

- **ДанныеТекущегоУзла**
- **ИдентификаторФормы** – служебный УИД для хранения данных во временном хранилище, пока открыта обработка
- **ИнтеграционнаяОбработка** - см. **ИнтеграционнаяОбработка**
- **Адрес**
- **ИнформацияСовместимости** – версия ИО, поддерживаемые конфигурации 1С, поддерживаемые версии конфигураций 1С, используется для сообщения на главной форме о успешном/неуспешном подключении ИО и её совместимости с текущей базой 1С
- **ИспользоватьREST_API** – флаг, означает возможность использования REST API
- **НастройкиИнициализированы**
- **ПараметрыПодключенияMS**
- **РасположениеКомпоненты**
- **СтруктураМетаданных**
- **СтруктураПодключения**
- **БазыMS** – аналог **ПараметрыПодключенияMS**
- **ТекущиеНастройкиMS**
- **ТипИнтерфейса**

Программно,
т.к. в нее просто сохраняются
другие, уже инициализи-
рованные, переменные ядра

—

ЕстьЧастнаяБаза

Тип – Булево. Означает, что к текущей базе 1С привязан какой-либо узел и этот узел будет использовать частную базу для обмена документами и справочниками

Программно

Да, на главной форме при выборе узла (торгового объекта)

ИмяИнтеграционнойОбработки

Тип – Строка, СправочникСсылка
Для внешней ИО – путь к файлу внешней обработки.
Для встроенной в конфигурацию 1С ИО – имя метаданных встроенной обработки.
Для ИО из справочника «Дополнительные отчеты и обработки» - ссылку на элемент справочника

ХОН

Да, в форме настройки ИО

ИнтеграционнаяОбработка Подключена
Тип – Булево. Означает, что интеграционная обработка успешно подключена. Флаг взводится при инициализации ядра
Программно
Да, если потеряна связь с сервером MS
ПолноеИмяОбъекта
Тип – Строка. Полное имя обработки-ядра, например, «ВнешняяОбработка.КлеверенсТСД_ОсновнаяОбработка»
Программно
Нет
ТипОбработки
Тип – Строка. Определяет расположение ИО - «ФайлНаДиске», «Встроенная» или «СправочникСсылка»
Настройки базы MS
Да, в форме настройки ИО

Для глобального ядра это:

- реквизиты обработки, доступны из любого метода модуля объекта;
 - ЭтоWebСервис – флаг, означающий работу ядра из [веб-сервиса](#);
 - СОМЧастнойБазыMS, СОМВременнойБазыMS, СОМБазыMS – аналоги одноименных переменных локального ядра.
- серверные переменные модуля объекта, доступны из любого метода модуля объекта;
 - КэшАдресаПеременных – копия переменной АдресаПеременных, но используется только для онлайн-режима, когда между вызовами ядра нужно сохранять значения переменных ядра.
- переменная локального ядра АдресаПеременных, передающаяся в методы глобального ядра в качестве параметра, посредством которой есть доступ к большинству переменных локального ядра.

Что хранится в хранилище общих настроек 1С (ХОН)

Принцип сохранения/ загрузки данных в ХОН

Если у пользователя 1С есть право доступа «Администрирование данных», то сохраняется и индивидуальная настройка по имени пользователя, и общая (глобальная) настройка для всей базы 1С, с именем пользователя, равным пустой строке «».

Иначе сохраняется только индивидуальная настройка по имени пользователя.

Аналогичный принцип при загрузке данных из ХОН, но можно при загрузке указывать, загружать только

пользовательскую настройку или дополнительно еще и общую настройку

Ключ	Значение
Cleverence_ПараметрыПодключенияMS	Структура, содержащая Ид базы, имя базы, Ид приложения, строку подключения, путь и имя ИО, версию приложения и т.д.
Cleverence_РасположениеКомпоненты	Расположение внешней COM-компоненты для работы из 1С с базой MS: на клиенте 1С или на сервере 1С (файлы Cleverence.MobileSMARTS.ComConnector.dll, Cleverence.Infrastructure.dll, Cleverence.Warehouse.TerminalConnector.dll). При этом, сама база MS может быть расположена где угодно, главное, чтобы к ней был доступ
Cleverence_ДатаАктуальностиКэша	Дата и время, когда в последний раз из MS (файл customsettings.xml) в кэш были считаны настройки базы. Используется только для веб-сервиса либо внешнего соединения. Если прошло более 15 минут после последнего получения настроек, то нужно перечитать настройки из MS, т.к. за это время настройки могли поменять интерактивно из 1С или из панели MS
Cleverence_СтруктураМетаданныхДокументы1С	Метаданные документов 1С – имена документов, их реквизиты, табличные части, реквизиты табличных частей
Cleverence_СтруктураНастроек	Все настройки базы MS, такие как: настройки отборов документов и справочников, бизнес-процессы, произвольные коды и т.д.
Cleverence_СтруктураНастроек_ + ID_ТекущегоУзла	Все настройки базы MS, но уже для конкретного торгового объекта
Cleverence_СтруктураМетаданныхДокументыMS	Метаданные документов MS – имена документов, их реквизиты, табличные части, реквизиты табличных частей
Cleverence_ПользователиMS	Пользователи и группы пользователей , которые заведены в подключенной базе MS
Cleverence_ТаблицыMS	Таблицы MS (имена таблиц, имена полей таблиц) таких как: «Склады», «Остатки», «Цены» и т.д.
Cleverence_УстройстваMS	Ид и имена устройств, подключенных к базе
Cleverence_ИмяИнтеграционнойОбработки	Имя ИО, например, «ИнтеграционнаяОбработка_УТ_11_4»
Cleverence_РежимВыбораБазы	По строке подключения ИЛИ из списка
Cleverence_ТипИнтерфейса	REST_API ИЛИ COM

Cleverence_REST_API_ДатаАктуальностиСоединения

Дата и время последнего подключения по REST API к базе MS. Используется только для веб-сервиса либо внешнего соединения

Cleverence_ДатаАктуальностиНастроек_ + ПостфиксБазы

Дата, когда были сохранены/изменены настройки пользователем.

Обновляется в методах КомЯдро_ЗаписатьНастройкиSMARTS.

Проверка даты осуществляется в онлайн-вызовах со стороны MS, если настройки изменились, то происходит переинициализация настроек

Не нашли что искали?



Задать вопрос в техническую поддержку

Описание хранения настроек в базе Mobile SMARTS и их конвертация

Последние изменения: 2024-03-26

В каждом шаблоне **базы Mobile SMARTS** (например «Клеверенс Магазин 15 Полный.mstmpl») в папке «\Обработки 1С\Настройки по умолчанию» располагаются файлы с настройками по умолчанию (они же — типовые настройки) для разных конфигураций 1С:

Клеверенс Магазин 15 Полный > Обработки 1С > Настройки по-умолчанию

Имя	Дата изменения	Тип	Размер
 astorVashMagazin	15.04.2020 15:49	Файл "XML"	834 КБ
 DalionTrend	15.04.2020 15:49	Файл "XML"	724 КБ
 DalionUmPro	15.04.2020 15:49	Файл "XML"	853 КБ
 roznica22	15.04.2020 15:49	Файл "XML"	1 347 КБ
 roznica23	15.04.2020 15:49	Файл "XML"	1 370 КБ
 shtrihm7	15.04.2020 15:49	Файл "XML"	614 КБ
 shtrihm52	15.04.2020 15:49	Файл "XML"	774 КБ
 unf16	15.04.2020 15:49	Файл "XML"	1 183 КБ
 unfukr16	15.04.2020 15:49	Файл "XML"	1 161 КБ
 ut103	15.04.2020 15:49	Файл "XML"	582 КБ
 ut110	15.04.2020 15:49	Файл "XML"	1 633 КБ
 ut112	15.04.2020 15:49	Файл "XML"	1 704 КБ
 ut114	15.04.2020 15:49	Файл "XML"	1 726 КБ

Настройка "Склад 15, Полный"

— □ ×

Настройка подключения 1С

Выберите конфигурацию и базу 1С, с которой будет происходить работа в Mobile SMARTS!



Конфигурация 1С: 1С: Управление торговлей 11.4 (поддержка он-лайн)

Выбранная конфигурация, с которой будет происходить дальнейшая работа. От выбранной конфигурации будет зависеть по каким настройкам будет работать выбранная база и какую интеграционную обработку использовать.

База данных 1С: Имя базы: ut114demo
Версия 1С: 8.3.16.1063
Папка/Адрес: Srvr="VM-Server2019-PSG-1С";Ref="ut114demo";

Изменить...

Выберите базу 1С, с которой будет происходить работа. База должна соответствовать выбранной конфигурации.

Помощь

Назад

Далее

Прервать

В мастере создания новой базы из шаблона мы выбираем, с какой **конфигурацией 1С** будет обмениваться данными новая база Mobile SMARTS. Соответствие конфигураций 1С и файлов настроек хранится в файле шаблона «1СConfigs.xml»:

```

94 <Config id="UT114" name="1С: Управление торговлей 11.4" MainFilePath="КлеверенсТЦД_ОсновнаяОбработка.epf" SettingsFile = "ut114.xml" OnlineSupport="True" OneCConfigName = "УправлениеТорговлей"
95 <Support>Shop15Mini</Support>
96 <Support>Shop15Standard</Support>
97 <Support>Shop15Adv</Support>
98 <Support>Shop15Full</Support>
99 </Config>

```

После распаковки данных из шаблона в папку базы Mobile SMARTS копируется папка «Обработки 1С» с файлами настроек для всех конфигураций 1С, копируется файл «1СConfigs.xml», а также копируется файл, соответствующий выбранной конфигурации 1С, и сохраняется с именем «default_customsettings.xml». Например, мы выбрали конфигурацию «Управление торговлей 11.4», поэтому в файле «default_customsettings.xml» хранятся данные из файла «ut114.xml» (на скриншоте можно увидеть, что размеры файлов совпадают с точностью до байта).

Имя	Тип	Размер	Дата	Атрибуты
astorVashMagazin	xml	853 670	15.04.2020 15:49	—
DalionTrend	xml	740 403	15.04.2020 15:49	-a-
DalionUmPro	xml	873 109	15.04.2020 15:49	-a-
roznica22	xml	1 379 191	15.04.2020 15:49	-a-
roznica23	xml	1 402 577	15.04.2020 15:49	-a-
shtrihm52	xml	792 145	15.04.2020 15:49	-a-
shtrihm7	xml	628 145	15.04.2020 15:49	-a-
unf 16	xml	1 210 831	15.04.2020 15:49	-a-
unfukr16	xml	1 188 187	15.04.2020 15:49	-a-
ut 103	xml	595 005	15.04.2020 15:49	-a-
ut 110	xml	1 671 722	15.04.2020 15:49	-a-
ut 112	xml	1 744 076	15.04.2020 15:49	-a-
ut 114	xml	1 766 599	15.04.2020 15:49	-a-

Имя	Тип	Размер	Дата	Атрибу
Backup	<Папка>		16.04.2020 16:52	—
Client Installer	<Папка>		15.04.2020 19:15	—
Connections	<Папка>		11.04.2020 12:51	—
Control Panel	<Папка>		30.03.2020 09:00	—
Documents	<Папка>		11.04.2020 15:59	—
DOSTUP	<Папка>		30.03.2020 09:00	—
Label Templates	<Папка>		30.03.2020 09:00	—
Licenses	<Папка>		30.03.2020 09:01	—
Logs	<Папка>		15.04.2020 19:15	—
Server	<Папка>		30.03.2020 10:58	—
XlsCsv	<Папка>		30.03.2020 09:00	—
Обработки 1С	<Папка>		30.03.2020 09:00	—
Cleverence.Install.Template.Common	dll	750 224	30.03.2020 09:00	-a-
appres	resources	254 825	30.03.2020 09:00	-a-
Cleverence.Warehouse.AuthorizationTokens	smc	934	11.04.2020 15:50	-a-
1СConfigs	xml	24 736	30.03.2020 09:00	-a-
AppDescription	xml	1 369	30.03.2020 09:00	-a-
customsettings	xml	5 819	11.04.2020 15:59	-a-
default_customsettings	xml	1 766 599	30.03.2020 09:00	-a-
OnRemoteInstall	xml	259	30.03.2020 09:00	-a-
OnUpdate	xml	973	30.03.2020 09:00	-a-
settings	xml	682	11.04.2020 15:59	-a-

В файле «default_customsettings.xml» хранятся все настройки базы, которые будут использованы по умолчанию. Так как это типовые настройки, то их нельзя изменять программно в коде 1С.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <SettingsValueList xmlns:clr="http://schemas.cleverence.ru clr" capacity="64" count="24">
3   <SettingsValueNode name="ВерсияНастроек ПоУмолчанию">
8   <SettingsValueNode name="ИмяИнтеграционнойОбработкиКонфигурации1С">
13  <SettingsValueNode name="ПутьИнтеграционнойОбработкиКонфигурации1С">
18  <SettingsValueNode name="ВыгружатьСправкиАСразу">
23  <SettingsValueNode name="ВыгружатьДополнительнуюИнформациюСразу">
28  <SettingsValueNode name="ВыгружатьИнформациюДляЦенниковСразу">
33  <SettingsValueNode name="ВыгружатьКИЗСразу">
38  <SettingsValueNode name="ВыгружатьОстаткиПоячейкамСразу">
43  <SettingsValueNode name="ИспользоватьШаблоныВесовогоТовара">
48  <SettingsValueNode name="ПрефиксКодаВесовогоТовара">
53  <SettingsValueNode name="ЧислоЗнаковПослеЗапятойВесТовара">
58  <SettingsValueNode name="ДлинаКодаВесовогоТовара">
63  <SettingsValueNode name="ВыгружатьКонтрагентовСразу">
68  <SettingsValueNode name="ВыгружатьНоменклатуруСразу">
73  <SettingsValueNode name="ВыгружатьОстаткиСразу">
78  <SettingsValueNode name="ВыгружатьСерииСразу">
83  <SettingsValueNode name="ВыгружатьСкладыСразу">
88  <SettingsValueNode name="ВыгружатьХарактеристикиСразу">
93  <SettingsValueNode name="ВыгружатьЦеныСразу">
98  <SettingsValueNode name="ВыгружатьЯчейкиСразу">
103 <SettingsValueNode name="ПереписыватьНоменклатуруПриВыгрузке">
108 <SettingsValueNode name="НастройкиОтборовУзлов Интерактивный">
1224 <SettingsValueNode name="НастройкиОтборовУзлов Онлайн">
2340 <SettingsValueNode name="ФормаОбменаВыбранныеКолонки">
2400 <SettingsValueNode name="ГлобальныеПараметры">
2432 <SettingsValueNode name="ПроизвольныеКоды ПоУмолчанию">
3319 <SettingsValueNode name="НастройкиБизнесПроцессов ПоУмолчанию">
81825 </SettingsValueList>

```

После первого запуска обработки 1С в базу Mobile SMARTS записывается еще ряд настроек, которые уже считаются не типовыми, а пользовательскими, поэтому они сохраняются в отдельный файл — «customsettings.xml». Часть из них создает мастер первого запуска Mobile SMARTS, другая часть настроек создается из обработки 1С при первом запуске.

<code><SettingsValueList xmlns:clr="http://schemas.cleverence.ru/clr" capacity="32" count="26"></code>	
<code><SettingsValueNode name="НастройкиБизнесПроцессов Пользовательские"></code>	
<code><SettingsValueNode name="ВерсияОбработкиНастроек"></code>	
<code><SettingsValueNode name="ПроизвольныеКоды Пользовательские"></code>	
<code><SettingsValueNode name="ВерсияНастроек"></code>	
<code><SettingsValueNode name="ID ТекущегоУзла"></code>	
<code><SettingsValueNode name="ПодключеннаяБазаИмяКонфигурации1С"></code>	заполняются из 1С
<code><SettingsValueNode name="ПодключеннаяБазаСинонимКонфигурации1С"></code>	
<code><SettingsValueNode name="ПодключеннаяБазаСтрокаСоединения"></code>	
<code><SettingsValueNode name="ПодключеннаяБазаВерсияПлатформы1С"></code>	
<code><SettingsValueNode name="ПодключеннаяБазаВерсияКонфигурации1С"></code>	
<code><SettingsValueNode name="База1СПодключенаДата"></code>	
<code><SettingsValueNode name="Ид"></code>	
<code><SettingsValueNode name="ИдПриложения"></code>	
<code><SettingsValueNode name="УровеньПриложения"></code>	
<code><SettingsValueNode name="ИдОсновногоПриложения"></code>	
<code><SettingsValueNode name="ИмяИнтеграционнойОбработкиКонфигурации1С"></code>	
<code><SettingsValueNode name="ПутьИнтеграционнойОбработкиКонфигурации1С"></code>	
<code><SettingsValueNode name="ТипОбработки"></code>	
<code><SettingsValueNode name="OneCconfigId"></code>	
<code><SettingsValueNode name="ExchangeType"></code>	
<code><SettingsValueNode name="ПользовательОсновнойБД"></code>	
<code><SettingsValueNode name="ПарольПользователяОсновнойБД"></code>	заполняются мастером
<code><SettingsValueNode name="OneCconfigName"></code>	первого запуска
<code><SettingsValueNode name="OneCconfigVersion"></code>	
<code><SettingsValueNode name="OneCProcessingFileName"></code>	
<code><SettingsValueNode name="startWizard"></code>	

Когда мы запрашиваем у сервера Mobile SMARTS какую-то настройку по ее ключу, то сперва данная настройка ищется в пользовательских настройках, и если там не была найдена, то ищется в типовых. Если нигде не найдена, то возвращается значение «Неопределено».

В обработке 1С явно указывается доступный список настроек, которые будут считаны из базы Mobile SMARTS (они же - метаданные настроек MS, структура настроек MS). Делается это в методе `Глядро_ПолучитьПустуюСтруктуруНастроекSMARTS()`, поэтому, если в новом релизе того или иного продукта Mobile SMARTS (например, «Склад 15», «Магазин 15») обновляются метаданные настроек (добавляются новые настройки, удаляются существующие настройки, меняется тип значений существующих настроек), их также нужно явно модифицировать в этом методе для возможности корректной работы с ними в обработке:

```
// Возвращает пустую структуру настроек Mobile Smarts //...
функция Глядро_ПолучитьПустуюСтруктуруНастроекSMARTS (СериализуемыеНастройки = Неопределено) Экспорт
    СтруктураНастроекSMARTS = Новый Структура;

    СтруктураНастроекSMARTS.Вставить ("ИспользоватьШаблоныВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ПрефиксКодаВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ДлинаКодаВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ЧислоЗнаковПослеЗапятойВесТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ПрефиксВключенВКодВесовогоТовара", Ложь);
    СтруктураНастроекSMARTS.Вставить ("РазмерПорцииНоменклатуры", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ВыгружатьНоменклатуруДокументаПередВыгрузкойДокумента", Ложь);
    СтруктураНастроекSMARTS.Вставить ("УстановитьПоискПоНаименованиюИАртикулу", Ложь);

    СтруктураНастроекSMARTS.Вставить ("ВыгружатьНоменклатуруСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ВыгружатьКонтрагентовСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ВыгружатьОстаткиСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ВыгружатьСерииСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ВыгружатьСкладыСразу", Ложь);
    СтруктураНастроекSMARTS.Вставить ("ВыгружатьФормыСразу", Ложь);
```

Также, существует категория сериализуемых настроек, которые не являются значениями примитивного типа («Число», «Строка», «Дата», «Булево», «Ссылка»), и которые необходимо сериализовывать в строку определенного формата для возможности хранения их в текстовом файле и десериализовывать при загрузке из файла. Это могут быть таблицы значений, списки значений и другие составные типы данных. В обработке для целей сериализации/десериализации используются методы `ЗначениеВСтрокуВнутр()` и `ЗначениеИзСтрокиВнутр()`. Для сериализуемых настроек существует метод, в котором описывается состав

сериализуемых настроек — `Глядро_ПолучитьКатегорииНастроекMS()` и метод, который описывает, какие колонки в какой таблице могут существовать — `Глядро_СформироватьМетаданныеНастроекMS()`, например, для таблицы с настройками обмена справочниками могут быть только 2 поля — «ИмяТаблицы» и «НастройкаКомпоновщикаСтрокой». Для таблицы с настройками обмена документами это могут быть поля «ID_БизнесПроцесса», «Название», «НастройкаКомпоновщикаСтрокой» и т.д. Пример:

```

Функция Глядро_СформироватьМетаданныеНастроекMS ()

СтруктураМетаданныхНастроекMS = Новый Структура;

// Обмен справочниками
ШаблонНастроек = Новый Структура;
ШаблонНастроек.Вставить ("ИмяТаблицы", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (50)));
ШаблонНастроек.Вставить ("НастройкаКомпоновщикаСтрокой", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (0)));
СтруктураМетаданныхНастроекMS.Вставить ("НастройкиОбменаСправочников", ШаблонНастроек);

// Обмен документами
ШаблонНастроек = Новый Структура;
ШаблонНастроек.Вставить ("ID_БизнесПроцесса", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (40)));
ШаблонНастроек.Вставить ("Название", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (100)));
ШаблонНастроек.Вставить ("ТипИсходногоДокумента1С", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (200)));
ШаблонНастроек.Вставить ("ТипИсходногоДокумента1С_Синоним", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (200)));
ШаблонНастроек.Вставить ("НастройкаКомпоновщикаСтрокой", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (0)));
ШаблонНастроек.Вставить ("Используется", Новый ОписаниеТипов ("Булево"));
ШаблонНастроек.Вставить ("ВариантОтбора", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (0)));
ШаблонНастроек.Вставить ("ID_ПроизвольногоОтбора", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (0)));
СтруктураМетаданныхНастроекMS.Вставить ("НастройкиОтборовУзел_Онлайн", ШаблонНастроек);
СтруктураМетаданныхНастроекMS.Вставить ("НастройкиОтборовУзел_Интерактивный", ШаблонНастроек);
СтруктураМетаданныхНастроекMS.Вставить ("НастройкиОтборовУзел_Автообмен", ШаблонНастроек);

// Бизнес-процессы
ШаблонНастроек = Новый Структура;
ШаблонНастроек.Вставить ("ID_БизнесПроцесса", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (40)));
ШаблонНастроек.Вставить ("Используется", Новый ОписаниеТипов ("Булево"));
ШаблонНастроек.Вставить ("ЭтоТиповойБизнесПроцесс", Новый ОписаниеТипов ("Булево"));
ШаблонНастроек.Вставить ("Название", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (255)));
ШаблонНастроек.Вставить ("ТипИсходногоДокумента1С", Новый ОписаниеТипов ("Строка",,,, Новый КвалификаторыСтроки (255)));

```

Для того чтобы в **процессе работы с обработкой** каждый раз не обращаться к базе Mobile SMARTS за получением значения настройки, все настройки при запуске обработки кэшируются в переменную ядра `АдресаПеременных.ТекущиеНастройкиMS`, а для **онлайн-режима** — дополнительно в хранилище общих настроек 1С по ключу `Cleverage_СтруктураНастроек`. Впоследствии, если какая-то настройка изменилась, она сохраняется и в базу Mobile SMARTS и обновляется в кэше, чтобы поддерживать актуальность настроек в кэше. Для онлайн-режима дополнительно существует механизм обновления настроек: если кто-то параллельно работает с базой Mobile SMARTS из 1С, то этот пользователь может изменить настройки, поэтому каждые 900 секунд происходит проверка актуальности настроек в кэше. Если они стали неактуальными, они перечитываются из базы Mobile SMARTS в кэш. Для этого в хранилище общих настроек 1С сохраняются следующие данные:

- `Cleverage_ДатаАктуальностиНастроек` — дата, когда настройки были сохранены/изменены пользователем.
- `Cleverage_ДатаАктуальностиКэша` — дата, когда данные были загружены в кэш. Если дата кэша старше даты настроек, то запускается обновление кэша.

Например, вы поменяли **глобальный параметр** у какого-то узла, от которого зависит отбор документов. При работе в режиме онлайн он никак не обновится в кэше, пока соединение не будет перезапущено. Это и есть проверка на актуальность.

Иначе говоря, `Cleverage_ДатаАктуальностиНастроек` — это дата последней записи настроек в Mobile SMARTS, а `Cleverage_ДатаАктуальностиКэша` — это дата последнего считывания настроек из Mobile SMARTS в кэш. И если `Cleverage_ДатаАктуальностиКэша <= Cleverage_ДатаАктуальностиНастроек`, т.е. если после того как считали настройки, что-то еще изменили, то нужно перечитать настройки.

Во всех формах, в которых возможен просмотр или редактирование настроек, например, **бизнес-процессов**, произвольных кодов, отборов документов и т.д. существует функционал по экспорту/импорту настроек в/из файла. Это может понадобиться для создания резервной копии настроек или для переноса настроек в другую базу Mobile SMARTS.

Формат хранения настроек бизнес-процессов и произвольных кодов

Бизнес-процессы хранятся в 3х ключах:

- `НастройкиБизнесПроцессов_ПоУмолчанию`.
- `НастройкиБизнесПроцессов_Пользовательские`.
- `НастройкиБизнесПроцессов_ПользовательскиеНастройки`.

`НастройкиБизнесПроцессов_ПоУмолчанию` — это наши типовые настройки, которые недоступны для редактирования. Они разворачиваются вместе с базой Mobile SMARTS и обновляются в файле `default_customsettings.xml`. Пользователь может только поменять некоторые параметры, например, «Использование», «Контроль количества», «Режим записи», «Выбор пользователя» и т.д.

`НастройкиБизнесПроцессов_ПользовательскиеНастройки` — сюда сохраняются те самые параметры, которые разрешено менять пользователю в типовых настройках. Они сохраняются в файл `customsettings.xml`. Если по этому ключу отсутствует настройка, значит, пользователь ничего не менял в типовых бизнес-процессах.

`НастройкиБизнесПроцессов_Пользовательские` — сохраняются бизнес-процессы, которые создал пользователь. Они сохраняются в файл «`customsettings.xml`».

При инициализации настройки бизнес-процессы собираются из этих трех ключей в единую таблицу (метод `Глядро_СобратьОбщуюСтруктуруБизнесПроцессов()`) + проходит обновление настроек, если версия настроек ниже используемой в ядре.

Если пользователю надо что-то поменять в типовом бизнес-процессе, он может скопировать его, поменять что нужно и отключить типовой. Его скопированный бизнес-процесс будет иметь другой ID.

Главный вопрос - как мы обновляем наши типовые БП?

1. Разворачиваем базу Mobile SMARTS с типовыми настройками.
2. Открываем обработку, заходим в [настройки бизнес-процессов](#).
3. В чистой базе будут только типовые, недоступные для редактирования бизнес-процессы. Сохраняем их в файл.
4. Заходим в папку с базой Mobile SMARTS.
5. Удаляем, переименовываем или меняем расширение у файла `default_customsettings.xml`.
6. Перезапускаем обработку, заходим в настройки бизнес-процессов.
7. Видим пустой список, загружаем настройки из файла, в который сохраняли типовые.
8. Настройки загрузились, теперь они доступны как пользовательские.
9. Меняем всё что душе угодно.
10. Открываем файл «`customsettings.xml`», форматируем в нормальный вид. Находим ключ `НастройкиБизнесПроцессов_Пользовательские`.
11. Переименовываем этот ключ в `НастройкиБизнесПроцессов_ПоУмолчанию`.
12. Если менялись произвольные коды, аналогично находим ключ `ПроизвольныеКоды_Пользовательские` и переименовываем в `ПроизвольныеКоды_ПоУмолчанию`.
13. Копируем с заменой эти ключи в нужный файл настроек в зависимости от конфигурации —

«`\Git\2.Git_m15\Общий\Обработки 1С\Настройки по-умолчанию`».

Аналогичная ситуация с произвольными кодами:

1. `ПроизвольныеКоды_ПоУмолчанию` — по этому ключу в файле «`default_customsettings.xml`» хранятся

типовые произвольные коды. Они недоступны для изменения из обработки.

2. **ПроизвольныеКоды_Пользовательские** — по этому ключу в файле «customsettings.xml» хранятся пользовательские произвольные коды. Они доступны для изменения из обработки.

При инициализации произвольные коды собираются из этих двух ключей в единую таблицу (метод `Глядро_СобратьОбщуюСтруктуруПроизвольныхКодов()`) + проходит обновление настроек, если версия настроек ниже используемой в ядре. При экспорте бизнес-процессов в файл также экспортируются и произвольные коды

В прошлых релизах обработки пользовательские произвольные коды сохранялись в 4 ключа:

- **ПроизвольныеКоды_Выгрузка_ШапкаДокумента.**
- **ПроизвольныеКоды_Выгрузка_ТабличнаяЧастьДокумента.**
- **ПроизвольныеКоды_Загрузка_ШапкаДокумента.**
- **ПроизвольныеКоды_Загрузка_ТабличнаяЧастьДокумента**, но потом стали храниться в одном ключе **ПроизвольныеКоды_Пользовательские**, поэтому, для обратной совместимости с прошлыми релизами обработки, существует метод `Глядро_ОбъединитьПроизвольныеКоды()`, который объединяет все пользовательские произвольные коды в один ключ **ПроизвольныеКоды_Пользовательские**.

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Обновление версии структуры настроек Mobile SMARTS

Последние изменения: 2024-03-26

Подробно про механизм хранения настроек в базе Mobile SMARTS и их конвертацию можно прочитать [здесь](#).

В типовых настройках (файл «default_customsettings.xml») хранится отдельная служебная настройка по имени «ВерсияНастроек_ПоУмолчанию» (ее значение представляет собой строку, например, «1.0.1.2»). При выходе нового релиза возможна ситуация, когда изменится структура настроек (добавляются новые настройки, удаляются существующие настройки, меняется тип значений существующих настроек). В этом случае версия настроек по умолчанию будет иметь другое значение, например, «1.0.1.3». Поэтому, чтобы обеспечить работоспособность уже существующих у пользователя настроек в новом релизе, предусмотрен механизм обновления структуры настроек.

В некоторых случаях, при обновлении своей текущей [базы Mobile SMARTS](#) до нового релиза, пользователь только копирует новые файлы обработки-ядра и [интеграционной обработки](#), а файл «default_customsettings.xml» не копирует, он остается тем же, каким он был при первом разворачивании текущей базы Mobile SMARTS из шаблона. Поэтому в самой обработке-ядре существует переменная, отвечающая за версию настроек - ВерсияОбработки, например, «1.3.0.1», она получается из метода СведенияОВнешнейОбработке().ВерсияСтруктурыНастроек.

В кэше хранятся:

- ВерсияОбработки — версия из модуля обработки;
- ВерсияОбработкиКлеверенсТСД, она же ВерсияНастроек_ПоУмолчанию — версия обработки по умолчанию из «default_customsettings.xml»;
- ВерсияОбработкиНастроек — версия из «customsettings.xml»;
- ВерсияНастроек — из «customsettings.xml», нужна для хранения версии типовых бизнес-процессов.

Обновление настроек при инициализации ядра

Краткий стек вызовов при инициализации ядра:

```
Глядро_ПолучитьДанныеДляИнициализации => Глядро_ВыполнитьОбновлениеСтруктурыНастроек =>
Глядро_ВыполнитьОбработчикОбновления => Глядро_ОбновлениеНастроекДоВерсии_1_0_1_1
..... => Глядро_ОбновлениеНастроекДоВерсии_x_x_x_x =>
Глядро_ПроверитьОбновлениеТиповыхБизнесПроцессов - конвертация будет выполняться для обратной
совместимости, независимо от версии настроек => Глядро_ОбновитьСтруктуруНастроекБизнесПроцессов
=> Глядро_КонвертироватьВыгрузкуПолей - вызывается 2 раза для шапки и для ТЧ =>
Глядро_КонвертироватьЗагрузкуПолей - вызывается 2 раза для шапки и для ТЧ
```

Подробный стек вызовов при инициализации ядра:

```
=> Глядро_ПолучитьДанныеДляИнициализации() пСтруктураНастроек = Новый
Структура("ВерсияОбработкиНастроек,ВерсияОбработкиКлеверенсТСД",Неопределено,Неопределено);
пСтруктураНастроек =
Глядро_ПолучитьНастройкиНаСервереSMARTS(пСтруктураНастроек,,,АдресаПеременных); //
пСтруктураНастроек.ВерсияОбработкиНастроек = 1 003 000 001 (берется из customsettings.xml) //
пСтруктураНастроек.ВерсияОбработкиКлеверенсТСД = Ложь (откуда? из default? в customsettings такого ключа
нет) => Глядро_ВыполнитьОбновлениеСтруктурыНастроек() ВерсияНастроек_ПоУмолчанию =
пСтруктураНастроек.ВерсияОбработкиКлеверенсТСД // ВерсияНастроек_ПоУмолчанию = 0
ВерсияОбработки = СведенияОВнешнейОбработке().ВерсияСтруктурыНастроек; // ВерсияОбработки = 1 003 000
001 (берется из модуля обработки) ВерсияНастроек = пСтруктураНастроек.ВерсияОбработкиНастроек; //
```

ВерсияНастроек = 1 003 000 001

1. Если КлючНастройки = Неопределено, Тогда // т.е. обновляем все настройки ИзмененияВКэше = Новый

```
Структура("ВерсияОбработки,ВерсияОбработкиНастроек,ВерсияОбработкиКлеверенсТСД",ВерсияОбработки,
ВерсияНастроек,ВерсияНастроек_ПоУмолчанию); // версия из модуля обработки, версия из кастом, версия из
дефолта ИзменениеВНастройках = Новый Структура("ВерсияОбработкиНастроек,
ВерсияОбработкиКлеверенсТСД"); // + хранит ключи настроек, которые необходимо записать в настройки MS
Если версия из модуля обработки > версия из кастом сеттингс, тогда =>
Глядро_ВыполнитьОбработчикОбновления() // Обновление настроек, которые после реструктуризации
отправятся в базу SMARTS и сохранится версия настроек (customsettings) (в структуре изменится только
ИзменениеВНастройках.ВерсияОбработкиНастроек)
```

Если версия из модуля обработки > Версия по умолчанию (из default?) тогда =>

```
Глядро_ВыполнитьОбработчикОбновления() // Обновление настроек, которые сохранить нет возможности, а
все изменения возвращаются в кэш без сохранения (default_customsettings) (в структуре изменится
ВерсияОбработкиКлеверенсТСД) в цикле вызываем все обработчики обновления
ИзмененияВСтруктуре.Вставить(?
(ОбновлениеКэша,"ВерсияОбработкиКлеверенсТСД","ВерсияОбработкиНастроек"),МассивПереходныхВерсий[Инд
// последняя версия, на которую удалось перейти без ошибок
ИзмененияВКэше.Вставить("ВерсияОбработкиНастроек",ИзменениеВНастройках.ВерсияОбработкиНастроек);
Возврат Новый
Структура("ИзмененияВКэше,ИзменениеВНастройках",ИзмененияВКэше,ИзменениеВНастройках);
```

2. Реструктуризация по ключу настройки.

Если версия из модуля обработки > версия настроек по умолчанию, тогда

```
Глядро_ВыполнитьОбработчикОбновления() // Обновление настроек, которые сохранить нет возможности, а
все изменения возвращаются в кэш без сохранения (default_customsettings) Возврат
Структура.ВерсияОбработкиКлеверенсТСД => Глядро_ПроверитьОбновлениеТиповыхБизнесПроцессов -
конвертация будет выполняться для обратной совместимости, независимо от версии настроек
СтруктураНастроек = Новый
Структура("ВерсияНастроек,ВерсияНастроек_ПоУмолчанию",Неопределено,Неопределено);
СтруктураНастроек =
Глядро_ПолучитьНастройкиНаСервереSMARTS(СтруктураНастроек,,,АдресаПеременных); ...
преобразование типовых бизнес-процессов из таблицы значений в массив структур и сохранение, но только
в кэш ... Глядро_СохранитьНастройкуВБазеSMARTS("ВерсияНастроек",
СтруктураНастроек.ВерсияНастроек_ПоУмолчанию,,АдресаПеременных);
Глядро_ЗаписатьВКэшНастройкуSMARTS("ВерсияНастроек",СтруктураНастроек.ВерсияНастроек_ПоУмолчанию,А
... перенос пользовательских бизнес-процессов из ключа "НастройкиБизнесПроцессов" в ключ
"НастройкиБизнесПроцессов_Пользовательские" ... =>
Глядро_ОбновитьСтруктуруНастроекБизнесПроцессов - изменение, добавление, удаление колонок из
таблицы бизнес-процессов и т.д. => Глядро_КонвертироватьВыгрузкуПолей - вызывается 2 раза для шапки
и для ТЧ => Глядро_КонвертироватьЗагрузкуПолей - вызывается 2 раза для шапки и для ТЧ ...
преобразование типовых произвольных кодов из таблицы значений в массив структур и сохранение, но
только в кэш ... ... преобразование пользовательских произвольных кодов и сохранение в настройки и в
кэш ...
```

Обновление настроек при получении настроек по умолчанию

ЛокГлядро_ПолучитьНастройкиПоУмолчаниюНаСервереSMARTS, если версия обработки из модуля > версии по умолчанию, то вызываем: => Глядро_ВыполнитьОбновлениеНастроек - для каждого ключа из структуры настроек и записывает новое значение настройки по ключу, вызывает: => Глядро_ВыполнитьОбработчикОбновления - вызывает по порядку все обработчики настроек вида Глядро_ОбновлениеНастроекДоВерсии_1_0_1_1

Действия при экспорте настройки в файл

Вызывается при экспорте настройки в файл в формах списков «Узлы», «ГлобальныеПараметры», «БизнесПроцессы», «ОтборыСправочников», «ПроизвольныеКоды», «ПроизвольныеОтборыДокументов».

```
СтруктураНастроек = Новый Структура; СтруктураНастроек.Вставить("ТаблицаУзлов",
ПолучитьСтруктуруУзловДляВыгрузки(Истина));
СтруктураНастроек.Вставить("ВерсияОбработкиНастроек",_ЛокКонтекст.ЛокЯдро_ПолучитьИзКэшаНастройкуSMARTS
// это версия из кастом сеттингс, сохраняемая в файл, которая, впоследствии, может стать устаревшей,
неактуальной _ЛокКонтекст.ЛокЯдро_СохранитьНастройкиВФайл(СтруктураНастроек, "НастройкиУзлов");
```

Для бизнес-процессов и произвольных кодов делается несколько иные преобразования, т.к. эти настройки хранятся по нескольким ключам, см. код в соответствующих формах списков.

Действия при импорте настройки из файла

```
Вызывается при импорте из файла настроек в формах списков «Узлы», «ГлобальныеПараметры»,
«БизнесПроцессы», «ОтборыСправочников», «ПроизвольныеКоды», «ПроизвольныеОтборыДокументов»
СтруктураНастроек = _ЛокКонтекст.ЛокЯдро_ПолучитьНастройкиИзФайла("НастройкиУзлов"); Если
СтруктураНастроек <> Неопределено Тогда //Выполним обновление - Начало ВерсияОбработки
= _ЛокКонтекст.ЛокЯдро_ПолучитьИзКэшаНастройкуSMARTS("ВерсияОбработки"); // а это текущая версия
настроек, из модуля обработки Если СтруктураНастроек.Свойство("ВерсияОбработкиНастроек") Тогда
ВерсияОбработкиНастроек = СтруктураНастроек.ВерсияОбработкиНастроек; Иначе
ВерсияОбработкиНастроек = 0; КонецЕсли; Если ВерсияОбработки>ВерсияОбработкиНастроек
Тогда // если текущая версия выше, чем версия из файла, тогда выполняем обработчики обновления
СтруктураВызова = Новый Структура; СтруктураВызова.Вставить("ТипМетода", "Функция");
СтруктураВызова.Вставить("ИмяМетода", "ВыполнитьОбновлениеНастроек"); // алгоритм такой же как при
получении настроек по умолчанию СтруктураВызова.Вставить("ПараметрыМетода", Новый
Структура("ВерсияНастроек,СтруктураНастроек",ВерсияОбработкиНастроек,СтруктураНастроек));
СтруктураНастроек =
_ЛокКонтекст.Транзит_ВызватьГлобальныйМетодНаКлиенте(СтруктураВызова,,_ЛокКонтекст.АдресаПеременных);
КонецЕсли; //Выполним обновление - Конец
```

Для бизнес-процессов и произвольных кодов делается несколько иные преобразования, т.к. эти настройки хранятся по нескольким ключам, см. код в соответствующих формах списков.

Не нашли что искали?



Задать вопрос в техническую поддержку

Алгоритм работы с основной базой и базой для обмена на примере обработки продукта Mobile SMARTS

Последние изменения: 2024-03-26

Общее правило:

База для обмена используется в форме обмена документами в процессе загрузки/выгрузки документов и в формах выгрузки справочников и номенклатуры для выгрузки/очистки справочников, во всех остальных случаях используется основная база.

База для обмена определяется при открытии обработки или изменении текущего торгового объекта на главной форме по следующему алгоритму:

1. Если текущий торговый объект не выбран, то базой для обмена считается основная база.
2. Если текущий торговый объект выбран, то:
 - Если у него установлен переключатель, «Использовать основное подключение», то базой для обмена считается основная база.
 - Если у него установлен переключатель, «Использовать частную базу», то базой для обмена считается выбранная частная база.

Форма списка бизнес-процессов

- сохранение таблицы настроек бизнес-процессов
 - выгрузка доп.таблицы «Бизнес-процессы»
 - получение таблицы настроек БП и ТЧ с настройками
 - получение списка типов документов MS
 - восстановление настроек БП «по умолчанию»
-
- в основную базу
 - во все базы: основная + все частные
 - из основной базы
 - из основной базы
 - из основной базы

Форма настройки бизнес-процесса

- получение списка пользователей
-
- из основной базы

Форма списка настроек обмена документами

- сохранение и восстановление ТЧ с настройками
- восстановление настроек «по умолчанию»

- в основной базе
- из основной базы

Форма списка настроек обмена справочниками

- сохранение и восстановление ТЧ с настройками
- в основной базе

Форма списка торговых объектов

- сохранение и восстановление ТЧ с настройками
- выгрузка доп.таблицы «Узлы»
- в основной базе
- во все базы: основная + все частные

Форма настройки выгрузки номенклатуры

- настройки компоновки загружаются/сохраняются
- номенклатура выгружается
- очистка номенклатуры
- из основной базы
- в базу для обмена
- в базе для обмена

Форма настройки выгрузки справочников

- настройки компоновки загружаются/сохраняются
- справочник выгружается
- очистка справочника
- из основной базы
- в базу для обмена
- в базе для обмена

Форма настройки полей обмена

- сохранение и восстановление ТЧ с настройками
- в основной базе

Форма настройки произвольного кода

- сохранение и восстановление ТЧ с настройками
- в основной базе

Форма выбора табличной части

- получение списка доп.таблиц документа MS
- получение списка полей ТЧ документа MS
- из основной базы
- в основной базе

Форма настройки торгового объекта

- сохранение и восстановление ТЧ с настройками
- получение дерева пользователей
- выгрузка доп.таблицы «Узлы»
- в основной базе
- из основной базы
- во все базы: основная + все частные

Форма выбора параметра торгового объекта

- восстановление ТЧ с настройками
- в основной базе

Форма выбора торгового объекта

- восстановление ТЧ с настройками
- в основной базе

Форма выбора произвольного кода

- сохранение и восстановление ТЧ с настройками

- в основной базе

Форма просмотра документа Mobile SMARTS

- получение данных документа MS
- получение данных шапки документа MS

- из базы для обмена
- из базы для обмена

Форма обмена документами

При открытии:

- получение ТЧ с настройками
- метод «СформироватьДеревоПользователейMS»
- команда «ЗаполнитьСписокДокументовMS» получает шапки документов
- команда «УдалитьДокументMS»
- команда «ОткрытьФормуДокументаMS» получает список колонок ТЧ «СтрокиПлан» и «СтрокиФакт»

- из основной базы
- из базы для обмена

Выгрузка документов на ТСД:

- таблицу настроек БП получаем
- выгрузка документа идет

- из основной базы
- в базу для обмена

Загрузка документов с ТСД:

- таблицу настроек БП получаем
- при поиске подходящего БП получаем данные документа MS и шапку документа MS
- удаление документа после загрузки
- получение списка доп.таблиц, получение данных из доп.таблиц
- получение данных документа MS для загрузки в 1С
- получение данных документа MS для создания новых штрихкодов

- из основной базы
- из базы для обмена

Форма выбора бизнес-процесса

При открытии:

- получение списка типов документов MS
- из базы для обмена

Форма главная

- при выборе и очистке текущего узла получаем базу для обмена: «Объект.IDБазыMSДляОбмена = ПолучитьIDБазыMSДляОбмена();»
- при открытии формы и после удачного подключения базы MS в методе «СоздатьОбъектДрайвера»:

- получаем параметры подключения (ид, имя, строка подключения, имя приложения, ид приложения)
- восстанавливаем настройки в методе «ВосстановитьНастройки» (здесь восстанавливаются флажки выгрузки справочников и параметры весового товара)
- восстановление ТЧ с настройками

- из основной базы
- из основной базы
- из основной базы

получаем базу для обмена: «Объект.IDБазыMSДляОбмена = ПолучитьIDБазыMSДляОбмена();»

- при закрытии формы или при скрывании группы настройки справочников (если есть изменения и нажали «не сохранять изменения»), в методе «СохранитьНастройкиЗавершение» восстанавливаем настройки
 - «СохранитьНастройкиВыгрузкиСправочниковMS» - сохраняет флажки и параметры весового товара
-
- из основной базы
 - в основную базу

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Методичка по интеграции произвольной конфигурации «1С: Предприятия» с Mobile SMARTS

Применимо к продуктам: **Магазин 15** **Склад 15** **Магазин 15 Прайсчекер** **Кировка** **Скан 3**

Последние изменения: 2024-04-17

Минимально совместимая с продуктом версия платформы 1С — 8.3.10
Минимальная версия для работы основной и интеграционной обработок — 8.3.10
Минимальная версия для работы расширений Клеверенс — 8.3.11

Данная инструкция не подходит для интеграции старых продуктов типа «[Драйвер ТСД Wi-Fi ПРОФ](#)» и «[Драйвер инвентаризации ОС](#)» старыми методами (через промежуточную конфигурацию, старые обработки и т. д.).

Что нужно, чтобы сделать интеграцию произвольной конфигурации 1С с новыми продуктами на базе платформы Mobile SMARTS:

- 1. Написать СКД для номенклатуры, справочников и регистров, которые будут использоваться мобильной программой.**
Для большинства нужных справочников и регистров в отраслевом продукте на [платформе Mobile SMARTS](#) («[Магазин 15](#)», «[Склад 15](#)», «[Кировка](#)» и т. п.) уже есть свои метаданные, нужно только написать СКД, который выдаст информацию в заранее определенном жестком формате.
Это делается один раз и для онлайн, и для офлайн сценариев интеграции, т. е. в любом сценарии интеграции используются одни и те же СКД.
- 2. Настроить в специальной обработке «Клеверенс» правила обмена документами между 1С и мобильной программой.**
Это называется «[настройкой бизнес-процессов](#)». Бизнес-процесс в интеграции с платформой Mobile SMARTS определяет всё: от отбора нужных документов, до их выгрузки/ загрузки, правил заполнения и постобработки.
Это тоже делается один раз и для [онлайн-режима работы](#), и для [офлайн](#), т. е. для любых способов обмена используются одни и те же общие правила, даже если конкретные настройки для онлайн и офлайн разные.
- 3. Написать так называемые «обработчики» для обработки ситуаций, с которыми не справляется СКД или настройка бизнес-процесса.**
Не всё можно сделать отбором или настройкой, поэтому в обработке интеграции предусмотрены обработчики, где можно «поправить интеграцию».
Как правило, они что-то дозаполняют перед выгрузкой или после загрузки, вызывают механизмы модулей конфигурации.

Глоссарий

Клеверенс — компания-разработчик программного обеспечения для мобильных устройств, предназначенного для оптимизации бизнес-процессов линейного персонала при работе с товаром на штрихкодах.

Mobile SMARTS — платформа, разработанная компанией «Клеверенс», которая позволяет создавать, строить (программировать) алгоритмы работы (конфигурации) для мобильных устройств, подключенных к этой платформе + веб-служба сервера, которая входит в поставку [платформы Mobile SMARTS](#). (Пример аналогичного продукта: платформа «1С: Предприятие» и конфигурации 1С, работающие на этой платформе).

Интеграционная обработка — обработка, в которой преобразуются метаданные подключенной конфигурации 1С в метаданные для обработки КлеверенсТСД. Также в ней содержатся обработчики, выполняемые при выгрузке и/или загрузке документов, и СКД для выгрузки справочников и дополнительных таблиц в конфигурацию Mobile SMARTS.

Обработка КлеверенсТСД — обработка, в которой заложен используемый функционал и написаны методы обмена с базой Mobile SMARTS, чтобы их не нужно было изучать программистам 1С. Частные переменные и значения из конфигурации настраиваются в интеграционной обработке.

База Mobile SMARTS — база развернутого продукта («Склад 15» или «Магазин 15»), в которой будут храниться все выставленные настройки. [Описание структуры папки с базой](#).

Порядок действий для интеграции с учетной системой

1. Первым делом ознакомимся с [продуктом](#), с которым собираемся делать интеграцию.
2. Скачиваем и устанавливаем продукт по инструкции (см. соответствующую инструкцию для [Склада 15](#), [Магазина 15](#)). При установке продукта **ОБЯЗАТЕЛЬНО** выбираем либо вариант «под самостоятельную интеграцию», либо под ту конфигурацию 1С, которая наиболее подходит по метаданным к вашей конфигурации.
3. После установки у вас появится [база Mobile SMARTS](#), с которой будут работать мобильные устройства и будет происходить обмен данными с 1С.
4. Начнем адаптацию вашей учетной системой с изменения реквизитов интеграционной обработки. Берем за основу наиболее подходящую по метаданным конфигурации 1С интеграционную обработку из поставляемых с продуктом. Копируем её и переименовываем. Обработки интеграции находятся в папке базы: «\Обработки 1С\Интеграционные обработки».

« ProgramData > Cleverence > Базы Mobile SMARTS > Склад 15, Омни > Обработки 1С > Обработки интеграции

Имя	Дата изменения	Тип	Размер
ИнтеграционнаяОбработка_WMS_TL	21.09.2021 17:16	1С:Enterprise 8 Ext...	49 КБ
ИнтеграционнаяОбработка_Альфа_Ав...	21.09.2021 17:16	1С:Enterprise 8 Ext...	68 КБ
ИнтеграционнаяОбработка_Альфа_Ав...	21.09.2021 17:16	1С:Enterprise 8 Ext...	69 КБ
ИнтеграционнаяОбработка_БП_3_0	21.09.2021 17:16	1С:Enterprise 8 Ext...	156 КБ
ИнтеграционнаяОбработка_Медицина...	21.09.2021 17:16	1С:Enterprise 8 Ext...	59 КБ
ИнтеграционнаяОбработка_Пример	21.09.2021 17:16	1С:Enterprise 8 Ext...	73 КБ
ИнтеграционнаяОбработка_УАС_1_4	21.09.2021 17:16	1С:Enterprise 8 Ext...	79 КБ
ИнтеграционнаяОбработка_УНФ_1_6	21.09.2021 17:16	1С:Enterprise 8 Ext...	173 КБ
ИнтеграционнаяОбработка_УНФ_1_6_25	21.09.2021 17:16	1С:Enterprise 8 Ext...	178 КБ
ИнтеграционнаяОбработка_УНФ_КАЗ_...	21.09.2021 17:16	1С:Enterprise 8 Ext...	82 КБ
ИнтеграционнаяОбработка_УНФ_УКР_...	21.09.2021 17:16	1С:Enterprise 8 Ext...	86 КБ
ИнтеграционнаяОбработка_УТ_10_3	21.09.2021 17:16	1С:Enterprise 8 Ext...	175 КБ
ИнтеграционнаяОбработка_УТ_11_0	21.09.2021 17:16	1С:Enterprise 8 Ext...	90 КБ
ИнтеграционнаяОбработка_УТ_11_4	21.09.2021 17:16	1С:Enterprise 8 Ext...	277 КБ
ИнтеграционнаяОбработка_УТ_УКР_3_2	21.09.2021 17:16	1С:Enterprise 8 Ext...	112 КБ
ИнтеграционнаяОбработка_ШтрихМ_7...	21.09.2021 17:16	1С:Enterprise 8 Ext...	70 КБ

5. Открываем конфигуратором общий модуль скопированной обработки.
6. Изменяем проверяемую конфигурацию и версию в начальных функциях обработки, в области **#Область ИнформацияОСовместимости** на свои версии и наименования.
Функция ПоддерживаемаяКонфигурация()

```

функция ПоддерживаемаяКонфигурация ()
    ПоддерживаемаяКонф = Новый Структура;
    МассивИменПоддерживаемыхКонфигураций = Новый Массив;
    МассивИменПоддерживаемыхКонфигураций.Добавить ("комплекснаяавтоматизация");
    МассивИменПоддерживаемыхКонфигураций.Добавить ("управлениепредприятием");
    МассивИменПоддерживаемыхКонфигураций.Добавить ("управлениеторговлейдлябеларуси");
    МассивИменПоддерживаемыхКонфигураций.Добавить ("управлениеторговлейдляказахстана");
    МассивИменПоддерживаемыхКонфигураций.Добавить ("управлениеторговлей");

    ПоддерживаемаяКонф.Вставить ("МассивИменКонфигураций", МассивИменПоддерживаемыхКонфигураций);
    ПоддерживаемаяКонф.Вставить ("СинонимКонфигурации", "KA2, ERP2, УТБЕЛЗ, УТКЗ3, УТ11");
    Возврат ПоддерживаемаяКонф;
- Конецфункции

```

Функция ПоддерживаемыеВерсииКонфигурации (ИмяТекущейКонфигурации)

```

функция ВерсияAPI () ...
функция ПоддерживаемыеВерсииКонфигурации (ИмяТекущейКонфигурации)

    ДиапазонВерсий = Новый Структура;

    Если Найти (ИмяТекущейКонфигурации, "комплекснаяавтоматизация") Тогда
        ДиапазонВерсий.Вставить ("НачальнаяВерсия", "2.2.3.67");
        ДиапазонВерсий.Вставить ("КонечнаяВерсия", "2.4.99.92");

    ИначеЕсли Найти (ИмяТекущейКонфигурации, "управлениепредприятием") Тогда

```

Функция СведенияОВнешнейОбработке() Экспорт

```

функция СведенияОВнешнейОбработке() Экспорт

    РегистрационныеДанные = Новый Структура;

    РегистрационныеДанные.Вставить ("БезопасныйРежим", Ложь);
    РегистрационныеДанные.Вставить ("Вид", "ДополнительнаяОбработка");
    РегистрационныеДанные.Вставить ("Наименование", "ИнтеграционнаяОбработка_УТ_11_4");
    РегистрационныеДанные.Вставить ("Версия", ВерсияИнтеграции());
    РегистрационныеДанные.Вставить ("ВерсияСтруктурыНастроек", "0.0.0.0"); //Нужно для запуска процесса реструктуризации настроек
    РегистрационныеДанные.Вставить ("Информация", "Данная обработка осуществляет получение метаданных и отвечает за выполнение обработчиков
    //РегистрационныеДанные.Вставить ("Назначение", РегистрационныеДанные.Наименование);

```

Функция Конф_ПрефиксКонфигурации() Экспорт

```

функция Конф_ПрефиксКонфигурации() Экспорт

    Возврат "УТ_11_4";
- Конецфункции

```

- Для работы основного функционала интеграции потребуется привести метаданные вашей конфигурации к метаданным нашей обработки КлеверенсТСД. Это происходит в области: **#Область МетаданныеКонфигурации**.

```

функция Конф_ПолучитьМетаданныеНСИ() Экспорт
    // Номенклатура
    МетаданныеНоменклатура = Новый Структура;
    МетаданныеНоменклатура.Вставить ("ИмяСправочника", "Номенклатура");

    МетаданныеБазоваяЕИ = Новый Структура;
    МетаданныеБазоваяЕИ.Вставить ("ИмяСправочника", "УпаковкиЕдиницыИзмерения");
    МетаданныеБазоваяЕИ.Вставить ("ИмяРеквизита", "ЕдиницаИзмерения");

    МетаданныеНоменклатура.Вставить ("БазоваяЕдиницаИзмерения", МетаданныеБазоваяЕИ);

    // Характеристики
    МетаданныеХарактеристики = Новый Структура;
    МетаданныеХарактеристики.Вставить ("Использование", Истина);
    МетаданныеХарактеристики.Вставить ("ИмяСправочника", "ХарактеристикиНоменклатуры");
    МассивВладельцев = Новый Массив;
    МассивВладельцев.Добавить ("Номенклатура");
    МассивВладельцев.Добавить ("ВидыНоменклатуры");
    МетаданныеХарактеристики.Вставить ("Владелец", МассивВладельцев);

```

Указываем имена справочников, к которым будет обращаться (по каким реквизитам) в конфигурацию

из нашей интеграции.

```

- функция Конф_ИспользоватьХарактеристикиНоменклатуры () Экспорт
  Возврат Константы.ИспользоватьХарактеристикиНоменклатуры.Получить ();
- Конецфункции

```

8. Необходимо изменить все экспортные функции общего модуля для работы с вашей конфигурацией, за исключением экспортных функций обработчиков событий бизнес-процессов в области.
#Область ОбработчикиВыгрузкиДокументов.
9. Список обязательных экспортных функций в модуле интеграционной обработки, необходимых для работы:
 - СведенияОВнешнейОбработке.
 - ИнформацияСовместимости.
 - Конф_ПроверкаСоответствияИспользуемойКонфигурации.
 - Конф_ПрефиксКонфигурации.
 - Конф_ПолучитьМетаданныеНСИ.
 - Конф_ИспользоватьХарактеристикиНоменклатуры.
 - Конф_ПолучитьПустуюСсылкуСправочника.
 - Конф_ПолучитьСписокИспользуемыхСправочников.
 - ПолучитьМакетСКД.
10. Удаляем или комментируем лишние обработчики из интеграционной обработки, так как в них могут использоваться вызовы модулей типовой конфигурации, из-за чего интеграционная обработка не сможет подключиться к основной.

```

- #Область СозданиеНоменклатуры
- функция УТ_11_2_СоздатьНоменклатуруВБазе (СтруктураПараметров, СтрокаТСД = Неопределено) ...
- #КонецОбласти

- ОбработчикиВыгрузкиДокументов

- #Область ОбработчикиЗаполненияДокументов
- ОбщиеОбработчики
- УТ_11_2_ОбщиеОбработчики
- УТ_11_2_ПоступлениеТоваровУслуг

```

11. Интеграционная обработка готова. Если всё было сделано правильно, то она сможет загрузиться в основной обработке КлеверенсТСД.
12. В функции **Функция Конф_ПолучитьСписокИспользуемыхСправочников () Экспорт** нужно указать список выгружаемых справочников из 1С в соответствующие таблицы базы Mobile SMARTS.

```

Функция Конф_ПолучитьСписокИспользуемыхСправочников () Экспорт

СписокВыбора = Новый СписокЗначений;
СписокВыбора.Добавить ("Номенклатура", "Номенклатура");
Если Константы.ИспользоватьХарактеристикиНоменклатуры.Получить () Тогда
    СписокВыбора.Добавить ("Характеристики", "Характеристики номенклатуры");
КонецЕсли;
СписокВыбора.Добавить ("Контрагенты", "Контрагенты");
СписокВыбора.Добавить ("Склады", "Склады (места хранения)");
СписокВыбора.Добавить ("Остатки", "Остатки номенклатуры на складах");
СписокВыбора.Добавить ("Цены", "Цены номенклатуры");
Если Константы.ИспользоватьСерииНоменклатуры.Получить () Тогда
    СписокВыбора.Добавить ("Серии", "Серии номенклатуры");
КонецЕсли;
СписокВыбора.Добавить ("Ячейки", "Ячейки");
СписокВыбора.Добавить ("ДопРеквизиты", "Дополнительные реквизиты");
СписокВыбора.Добавить ("ОстаткиПоЯчейкам", "Остатки по ячейкам");

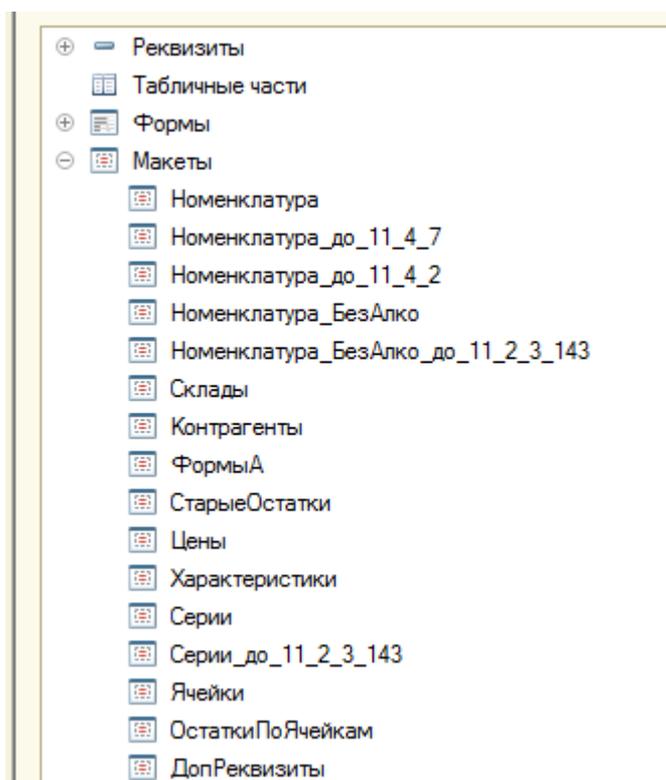
Если Метаданные.Справочники.Найти ("ШтрихкодыУпаковокТоваров") <> Неопределено Тогда
    СписокВыбора.Добавить ("ТранспортныеУпаковки", "Транспортные упаковки");
КонецЕсли;

Возврат СписокВыбора;

Конецфункции

```

13. Для каждой выгружаемой таблицы из 1С в базу Mobile SMARTS, создаем в обработке макет системы компоновки данных (СКД).



После создания макетов СКД, в функции ПолучитьМакетСКД указываем в каких случаях и какой макет СКД нужно использовать.

```

#Область МакетыСКД
функция ПолучитьМакетСКД(ИмяТаблицы, РаботаБезАлкоголя = Ложь) Экспорт
    ИмяТекущейКонфигурации = НРег(СокрЛП(Метаданные.Имя));

    Если ИмяТаблицы = "Номенклатура" Тогда
        МетаданныеНСИ = Конф_ПолучитьМетаданныеНСИ();
        ЕстьАлкогольныеМетаданные = МетаданныеНСИ.АлкоНоменклатура.Использование;
        ВерсияКонфигурации = ПолучитьВерсиюКонфигурации(Метаданные.Версия);

        Если //Найти(ИмяТекущейКонфигурации, "управлениеторговлей") И ВерсияКонфигурации <= 11002003143
            НовыйМеханизмСерий() Тогда

            Если НЕ ЕстьАлкогольныеМетаданные
                ИЛИ РаботаБезАлкоголя
                ИЛИ Найти(ИмяТекущейКонфигурации, "управлениеторговлейдлябеларуси")
                ИЛИ Найти(ИмяТекущейКонфигурации, "управлениеторговлейдляказахстана") Тогда

                ИмяМакета = "Номенклатура_БезАлко";

            Иначе
                Попытка
                    ИспользоватьАлкоголь = ПолучитьфункциональнуюОпцию("ВестиСведенияДляДекларацийПоАлкогольнойПродукции");
                Исключение
                    ИспользоватьАлкоголь = Ложь;
                КонечПопытки;
    
```

14. В режиме 1С предприятия открываем основную обработку КлеверенсТСД, и в открывшемся окне подключаем созданную интеграционную обработку. Если всё сделано правильно, то обработка подцепится и покажет поддерживаемые версии и конфигурации.

Подключенная база Mobile SMARTS:

Склад 15, Базовый

Версия основной обработки:

1.0.2.59 Core, 25.10.2019, УФ

Текущая конфигурация:

УправлениеТорговлей, 11.4.8.84

Версия обработки интеграции:

API: v1, 1.0.2.60 Int, 25.10.2019

Путь к обработке интеграции:

C:\ProgramData\Cleverence\Базы Mobile SMARTS\6c1873fb-bc31-4baf-8436-39dfbb96cbf9\Обработки 1С\Обработки | ...

Конфигурации 1С, поддерживаемые обработкой интеграции:

КА2, ERP2, УТБЕЛЗ, УТК33, УТ11

Версии конфигураций 1С, поддерживаемые обработкой интеграции:

11.2.7.13-11.4.99.92

Состояние:

Можно работать! Всё загружено без ошибок!

15. Интеграционная обработка может быть использована в виде файла внешней обработки, добавлена в справочник с реквизитом типа ХранилищеЗначения для хранения обработки (например справочник «Дополнительные отчеты и обработки» в УТ 11.4) или как обработка, встроенная в конфигурацию.



Форма выбора интеграционной обработки *

Подтвердить

✕ Закрыть

Версия обработки интеграции:

API: v1, 1.0.2.58 Int, 18.09.2019

Конфигурации 1С, поддерживаемые обработкой интеграции:

КА2, ERP2, УТБЕЛ3, УТК33, УТ11

Версии конфигураций 1С, поддерживаемые обработкой интеграции:

11.2.7.13-11.4.9.92

Тип обработки:

Файл на диске

Элемент справочника

Встроенная

Ссылка на обработку интеграции:

ИнтеграционнаяОбработка УТ114

16. В обработке КлеверенсТСД в окне настройки обмена справочников можно настроить дополнительные отборы справочников и посмотреть предварительный список выгружаемых данных по каждой таблице.



Клеверенс: Настройка обмена справочников

Сохранить и закрыть

Сохранить

Сохранить в файл...

Восстановить из файла...

Переписывать номенклатуру на ТСД при выгрузке Выгружать номенклатуру документа при выгрузке документа

Разбить номенклатуру для выгрузки по: позиций (0 - не разбивать) ?

Установить поиск по наименованию и артикулу ?

Использовать шаблоны весового товара

Выбрать все

Снять все

Выгрузить выбранные

Очистить выбранные

Выгружать	Наименование	Выгрузка	Предпросмотр
<input checked="" type="checkbox"/>	Номенклатура	Выгрузка...	Предпросмотр...
<input checked="" type="checkbox"/>	Характеристики номенклатуры	Выгрузка...	Предпросмотр...
<input checked="" type="checkbox"/>	Контрагенты	Выгрузка...	Предпросмотр...

17. Для выгрузки и загрузки документов, в **настройках бизнес-процессов** добавим новый бизнес-процесс (БП) или копируем из имеющихся если есть (они появляются если выбрали работу с конкретной учетной системой в мастере настроек).

Клеверенс: Настройка бизнес-процесса

Статусы

Использование бизнес-процесса Отключено Пользовательский

Название:

Движение документов

Тип исходного документа 1С: ▾ ×

Тип документа MS: ▾

Загружать в тот же документ 1С:

Тип результирующего документа 1С: ▾ ×

Настройки выгрузки документа из 1С в Mobile SMARTS

Выполнить только произвольный код при выгрузке:

Реквизиты шапки: [Настроить правила выгрузки шапки документа...](#)

Реквизиты табличной части: [Настроить правила выгрузки табличной частей...](#)

Настройки загрузки документа из Mobile SMARTS в 1С

Выполнить только произвольный код при загрузке:

Реквизиты шапки: [Настроить правила загрузки шапки документа...](#)

Реквизиты табличной части: [Настроить правила загрузки табличной частей...](#)

Прочие настройки

Записывать документ после загрузки: ▾ Создавать новый документ при загрузке:

Проводить документ после загрузки: Коллективная работа:

Не разбивать объединенный документ при загрузке: Контроль количества:

Кому назначать: ...

Для каждого пользователя свой экземпляр документа:

Представление шапки: [Настроить реквизиты представления шапки...](#)

Комментарий:

ID бизнес процесса: ВРеbecf5dcb45047af99f6a6340d70fa08

В настройках БП обязательно нужно указать название, тип документа MS, тип исходящего документа 1С (для выгрузки) или тип результирующего документа 1С (для загрузки).

Указываем правила загрузки/ выгрузки реквизитов шапки и табличных частей документов в виде соответствий реквизитов.

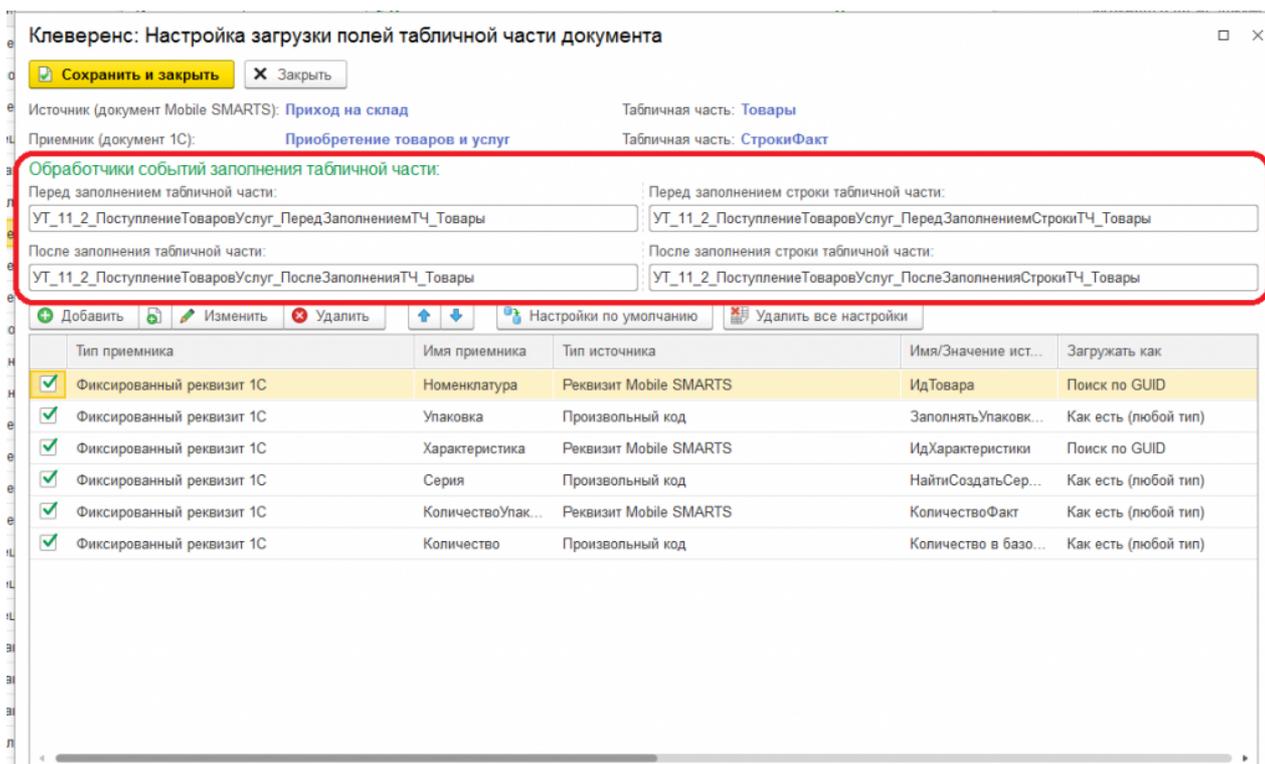
Клеверенс: Настройка выгрузки полей шапки документа

Источник (документ 1С): **ПриобретениеТоваровУслуг**

Приёмник (документ Mobile SMARTS): **Поступление**

	Тип приемника	Имя приёмника	Тип источника	Имя/Значение ист...	Выгружать как	Комментарий
<input checked="" type="checkbox"/>	Фиксированный реквизит Mobile SMARTS	Ид	Реквизит 1С	Ссылка	Ссылка (строка GUID)	
<input checked="" type="checkbox"/>	Фиксированный реквизит Mobile SMARTS	ИдСклада	Произвольное зна...	1	Как есть (простой тип)	
<input checked="" type="checkbox"/>	Фиксированный реквизит Mobile SMARTS	ВыдаватьПоШт...	Произвольное зна...	Да	Как есть (простой тип)	
<input checked="" type="checkbox"/>	Фиксированный реквизит Mobile SMARTS	ИдКонтрагента	Реквизит 1С	Контрагент	Ссылка (строка GUID)	
<input checked="" type="checkbox"/>	Фиксированный реквизит Mobile SMARTS	ИдСклада1С	Реквизит 1С	Склад	Ссылка (строка GUID)	

18. Для более тонкой настройки операции загрузки и заполнения документа в базу 1С, можно использовать обработчики из интеграционной обработки.



Названия обработчиков событий являются названием экспортных процедур в модуле интеграционной обработки. При возникновении события будет вызвано выполнение данных процедур:

```

#Область УТ_11_2_ПоступлениеТоваровУслуг
//Шапка и форма документа
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПередЗаполнениемДокумента (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПослеЗаполненияДокумента (СтруктураПараметров) Экспорт ...

//ТЧ_Товары
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПередЗаполнениемТЧ_Товары (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПередЗаполнениемСтрокиТЧ_Товары (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПослеЗаполненияСтрокиТЧ_Товары (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПослеЗаполненияТЧ_Товары (СтруктураПараметров) Экспорт ...

//ТЧ_Серии
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПередЗаполнениемТЧ_Серии (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПередЗаполнениемСтрокиТЧ_Серии (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПослеЗаполненияСтрокиТЧ_Серии (СтруктураПараметров) Экспорт ...
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПослеЗаполненияТЧ_Серии (СтруктураПараметров) Экспорт ...

#КонецОбласти

#Область УТ_11_2_ЗаказПоставщику
//Шапка и форма документа
Процедура УТ_11_2_ЗаказПоставщику_ПередЗаполнениемДокумента (СтруктураПараметров) Экспорт ...

```

При настройке правил загрузки полей шапки есть возможность указать обработчик события «После открытия формы». Обработчик события «После открытия формы» выполняется в момент окончания заполнения документа и после открытия формы загружаемого документа и вызывает экспортную процедуру из модуля формы интеграционной обработки.

Клеверенс: Настройка загрузки полей шапки документа

Сохранить и закрыть Закрыть

Источник (документ Mobile SMARTS): Приход на склад

Приемник (документ 1С): Приобретение товаров и услуг

Обработчик после открытия формы документа:
 УТ_11_2_ПоступлениеТоваровУслуг_ПослеОткрытияФормыДокумента

Обработчики событий заполнения документа:

Перед заполнением документа: УТ_11_2_ПоступлениеТоваровУслуг_ПередЗаполнениемДокумента

После заполнения документа: УТ_11_2_ПоступлениеТоваровУслуг_ПослеЗаполненияДокумента

Тип приемника	Имя приемника	Тип источника	Имя/Значение ист...	Загружать как

```

#Область Обработчики_ПослеОткрытияФормыДокумента
&НаКлиенте
Процедура УТ_11_2_Общий_ПослеОткрытияФормыДокумента (СтруктураПараметров) Экспорт ...
&НаКлиенте
Процедура УТ_11_2_ПоступлениеТоваровУслуг_ПослеОткрытияФормыДокумента (СтруктураПараметров) Экспорт ...
&НаКлиенте
Процедура УТ_11_2_ЗаказПоставщику_ПослеОткрытияФормыДокумента (СтруктураПараметров) Экспорт ...
  
```

19. Вместо настраиваемой в бизнес-процессе выгрузки документа 1С в базу Mobile SMARTS полями, есть возможность использовать полностью произвольный алгоритм выгрузки и загрузки.

Клеверенс: Настройка бизнес-процесса

Сохранить и закрыть | Сохранить | Скопировать | Закрыть

Статусы
Использование бизнес-процесса **Включено** **Пользовательский**

Название:

Движение документов

Тип исходного документа 1С:

Тип документа MS:

Загружать в тот же документ 1С:

Тип результирующего документа 1С:

Настройки выгрузки документа из 1С в Mobile SMARTS

Выполнить только произвольный код при выгрузке:

Обработчик выгрузки документа

Настройки загрузки документа из Mobile SMARTS в 1С

Выполнить только произвольный код при загрузке:

Реквизиты шапки: [Настроить правила загрузки шапки документа...](#)

Реквизиты табличной части: [Настроить правила загрузки табличной частей...](#)

Прочие настройки

Записывать документ после загрузки: Создавать новый документ при загрузке:

Проводить документ после загрузки: Коллективная работа:

Не разбивать объединенный документ при загрузке: Контроль количества:

Кому назначать:

Для каждого пользователя свой экземпляр документа:

Представление шапки: [Настроить реквизиты представления шапки...](#)

Комментарий:

ID бизнес процесса: ВРс5е1d3f272784f259ae2f5c9c5f7d5c8

Название обработчика выгрузки документа должно соответствовать названию экспортной функции в модуле интеграционной обработки отвечающей за выгрузку документа.

функция УТ_11_2_ПроизвольнаяВыгрузка(Документ1С, ДанныеНастроекИзКэша, НастройкаБП) Экспорт

```

ДанныеДокумента = Новый Структура();

ДанныеШапки = Новый Структура();
ДанныеШапки.Вставить("Name", Строка(Документ1С));
ДанныеШапки.Вставить("ИмяТипаДокумента", НастройкаБП.ТипДокументаMS);
ДанныеШапки.Вставить("НастройкаБизнесПроцесса", НастройкаБП.ID_БизнесПроцесса);
ДанныеШапки.Вставить("ИмяБП", НастройкаБП.Название);
ДанныеШапки.Вставить("Ид", XMLСтрока(Документ1С));
ДанныеШапки.Вставить("ИдСклада", 1);
ДанныеШапки.Вставить("ИдКонтрагента", XMLСтрока(Документ1С.Контрагент));
ДанныеШапки.Вставить("ИдСклада1С", XMLСтрока(Документ1С.Склад));
ДанныеШапки.Вставить("ИмяСклада", Документ1С.Склад.Наименование);
ДанныеШапки.Вставить("ИмяКонтрагента", Документ1С.Контрагент.Наименование);
ДанныеШапки.Вставить("РежимУпаковок", "КоробкиПалеты");
ДанныеШапки.Вставить("ВыдаватьПоШтрихкоду", Истина); //Нужно подставить глобальный параметр

Если НастройкаБП.КонтрольКоличества Тогда
    ДанныеШапки.Вставить("КонтрольКолва", НастройкаБП.КонтрольКоличества);
КонецЕсли;
Если НастройкаБП.КоллективнаяРабота Тогда
    ДанныеШапки.Вставить("ServerHosted", НастройкаБП.КоллективнаяРабота);
КонецЕсли;

пТабличныеЧасти = Новый Структура;
МассивДанныхСтрок = Новый Массив;
ТаблицаДанныхТЧ = ПолучитьТабличнуюЧастьДляВыгрузки();
Глядро_СвернутьТаблицуЗначений(ТаблицаДанныхТЧ, "КоличествоУпаковок");

Для Каждого СтрокаТЧ Из ТаблицаДанныхТЧ Цикл

    // Строка документа
    ДанныеДляВыгрузки_Строка = Новый Структура;
    ДанныеДляВыгрузки_Строка.Вставить("ИдТовара", XMLСтрока(СтрокаТЧ.Номенклатура));
    ДанныеДляВыгрузки_Строка.Вставить("ИдУпаковки", XMLСтрока(СтрокаТЧ.Номенклатура.ЕдиницаИзмерения));
    ДанныеДляВыгрузки_Строка.Вставить("ИдЕдиницыИзмерения", XMLСтрока(СтрокаТЧ.Номенклатура.ЕдиницаИзмерения));
    ДанныеДляВыгрузки_Строка.Вставить("КоличествоПлан", СтрокаТЧ.КоличествоУпаковок);
    ДанныеДляВыгрузки_Строка.Вставить("ИдБлока", СтрокаТЧ.ШтрихкодБлока);
    ДанныеДляВыгрузки_Строка.Вставить("SSCC", СтрокаТЧ.ШтрихкодПалеты);
    ДанныеДляВыгрузки_Строка.Вставить("ИдКоробки", СтрокаТЧ.ШтрихкодКоробки);
    ДанныеДляВыгрузки_Строка.Вставить("Марка", СтрокаТЧ.ШтрихкодБлока);
    ДанныеДляВыгрузки_Строка.Вставить("Маркировка", "Табак");

    МассивДанныхСтрок.Добавить(ДанныеДляВыгрузки_Строка);
КонецЦикла;

пТабличныеЧасти.Вставить("СтрокиПлан", МассивДанныхСтрок);

ДанныеДокумента.Вставить("ШапкаДокумента", ДанныеШапки);
ДанныеДокумента.Вставить("ТабличныеЧасти", пТабличныеЧасти);

Возврат ДанныеДокумента;

```

Конецфункции

20. После создания и настройки бизнес-процесса, добавим настройку отбора документов для интерактивного и онлайн обмена:

Клеверенс: Настройка отбора документов □ ×

Сохранить настройку Показать результаты отбора ▾

Статусы

Использование отбора: Включено

ID Торгового объекта: Торговый объект:

ID Бизнес-процесса: Тип документа 1С:

Тип обмена: Наименование БП:

Вариант отбора: Свой запрос:

Обязательные параметры		Параметры отбора	
Параметр	Значение	Представление	
Глубина отбора	500	⊖ Отбор	
Комментарий отсечения		<input type="checkbox"/> = Проведен	Равно Да
		<input checked="" type="checkbox"/> = Пометка удаления	Не равно Да
		<input type="checkbox"/> = Согласован	Равно Да

Значения общих параметров (справочно)

Наименование	Значение
КонтрольТовара: Разрешить сканировать только заявленный товар в документе "Пересчет товаров".	Нет
Назначение: Документ хранится на сервере, до его выбора пользователем с ТСД.	Да
Режим работы с палетами и коробами при приемке (для приобретения)	БезУпаковок
Режим работы с палетами и коробами при отгрузке (для заказа и реализации для обоих ГТТ)	БезУпаковок

Настройка отборов документов нужна для указания диапазона документов, с которыми будут работать пользователи в интерактивном и онлайн режимах. После настройки отбора можно нажать на кнопку «Показать результат отбора», и, если все настроено верно, появится список документов, соответствующий отбору.

21. Интерактивный обмен документами происходит на форме «Обмен документами». Онлайн-отбор происходит при вызове функции «ПолучитьСписокДокументов» с мобильного устройства, для получения результата отбора при работе в онлайне.

The screenshot displays the 'Обмен документами' (Document Exchange) interface. It is divided into two main sections: 'Документы 1С' (1С Documents) on the left and 'Документы Mobile SMARTS' (Mobile SMARTS Documents) on the right.

Документы 1С: This table lists documents with columns for 'Ссылка' (Link), 'Ск...' (Code), 'Пом...' (Priority), 'Конт...' (Contract), 'ВидО...' (Type), 'Колпечес...' (Quantity), and 'Сумма...' (Sum). The first row shows a document with ID 'ТД00-00000...' and a quantity of 0. Subsequent rows show similar documents with quantities of 3 and 10,800.

Документы Mobile SMARTS: This table lists documents with columns for 'Загружать' (Load), 'Дата создания' (Creation Date), 'Наименование' (Name), 'План' (Plan), 'Факт' (Fact), 'Состояние' (Status), and 'Им...' (Import). The first row shows a document created on 26.09.2019 with a plan of 10 and a status of 'Выгружен' (Exported). Other rows show documents with various dates, names, and statuses like 'Завершен, без...' (Completed, without...) and 'Выгружен' (Exported).

По всем вопросам и проблемам, возникшим в процессе самостоятельной интеграции, пишите на адрес электронной почты: 1cprog@cleverence.ru.

 интеграция, 1С

Не нашли что искали?



Задать вопрос в техническую поддержку

Рекомендации к проведению интеграции платформы Mobile SMARTS и конфигураций на базе «1С: Предприятие»

Последние изменения: 2024-03-26

В этой статье мы собрали основные советы, которые могут помочь как неопытным, так и давно работающим внедренцам Mobile SMARTS. Они смогут сделать процесс разработки интеграций с конфигурациями на базе «1С: Предприятия» чуть проще и позволят избежать подводных камней.

1. Избавьтесь от избыточных форм обработки – форма выбора бизнес-процесса (2 штуки), форма выбора пользователя и др. Их назначение только в том, чтобы отображать некоторую таблицу (которую сами же и формируют по своим алгоритмам) и возвращать в родительскую форму выбранную строку таблицы. В УФ это приводит к необходимости асинхронных перекрестных вызовов, что увеличивает количество кода и сложность его сопровождения. Как вариант, можно использовать либо типовые методы типа `ВыбратьСтроку()`, либо оставить только одну форму, в которую передавать заранее сформированную таблицу для выбора строки.
2. Не используйте методы компоненты `GetField()`, `SetField()` и прочих `GetEnvironment()`.
3. Почаще получайте обработки из хранилища и переносите изменения из них в вашу разрабатываемую версию. Если есть изменения в формах, лучше их не объединять!!! Ибо сливаются не только модули, но и все параметры формы — галочки, подвязки и т.д. Либо после слияния заново на форму свои изменения добавляйте вручную, либо просто заменяйте обработку в хранилище своей обработкой, если последний коммит обработки был вашим.
4. Если есть возможность, используйте тестовую среду, максимально приближенную к реальным условиям. Разнесите клиент и сервер «1С: Предприятия» по разным ПК, [установите Mobile SMARTS](#) на разных ПК, попробуйте [подключиться к базе удаленно](#), через интернет по протоколу `https`.
5. Пишите максимально читабельный код, добавляйте отступы везде, где посчитаете нужным, например:

Плохо
Хорошо
<code>переменная1 = один; переменная2 = три;</code>
<code>переменная1 = один; переменная2 = три;</code>

При добавлении отступов используйте клавишу «Tab», а не на пробел, чтобы форматирование текста не «съезжало» при копипасте.

6. Избегайте двусмысленности в написании кода, имен переменных! Старайтесь унифицировать имена переменных, делать их говорящими. Код должен быть максимально понятным.
7. Пишите комментарии в каких-то неочевидных и непонятных местах кода.
8. Используйте обработку исключений:

```
ОписаниеОшибки = ОписаниеОшибки();
Сообщить("Не удалось сделать что-то там. Причина: " + ОписаниеОшибки);
```

В таком случае метод `ОписаниеОшибки()` сможет вернуть описание ошибки.

9. Не используйте букву «ё». Ни в именах переменных, ни в реквизитах, ни в элементах формы, ни в синонимах, ни в комментариях.
10. Тщательно тестируйте свой код: в разных конфигурациях, в режиме клиента и сервера, в разных бизнес-процессах. Проводите пошаговую отладку алгоритма.
11. **ОБЯЗАТЕЛЬНО!!** После сравнения/ объединения обработки перепроверьте её работоспособность в реальных условиях!!!

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Интеграционная обработка «1С: Предприятия»

Последние изменения: 2024-03-26

Что такое интеграционная обработка

Интеграционная обработка — дополнительная обработка отвечающая за работу конфигурации 1С. Она нужна для получения метаданных подключенной конфигурации, получения и проверки констант подключенной конфигурации, содержит в себе необходимые запросы, СКД, обработчики которые выполняются для данной конфигурации. Работает совместно с основной обработкой, которая является ядром и отвечает за функционал. Интеграционная обработка подключается из обработки ядра.

Подключение интеграционных обработок

Подключить и использовать интеграционную обработку в основной обработке ядра можно несколькими способами, так как могут быть разные схемы её использования и обновления.

Вариант по умолчанию, который используется при первом развертывании продуктов Mobile SMARTS — относительный путь в каталоге базы. Задается он в начальных настройках, который распаковывается вместе с настройками для выбранной при настройке мастера конфигурацией 1С.

```
<SettingsValueNode name="ИмяИнтеграционнойОбработкиКонфигурации1С">
  <List count="0" />
  <SortedDictionary>null</SortedDictionary>
  <Value clr:Type="String">ИнтеграционнаяОбработка_УТ_11_4.epf</Value>
</SettingsValueNode>
<SettingsValueNode name="ПутьИнтеграционнойОбработкиКонфигурации1С">
  <List count="0" />
  <SortedDictionary>null</SortedDictionary>
  <Value clr:Type="String">\Обработки 1С\Обработки интеграции\</Value>
</SettingsValueNode>
```

Путь к интеграционной обработке может быть как относительный, так и полный, обработка ядра понимает любой из них.

```

<SettingsValueNode name="ИмяИнтеграционнойОбработкиКонфигурации1С">
  <List count="0" />
  <SortedDictionary>null</SortedDictionary>
  <Value clr:Type="String">ИнтеграционнаяОбработка_УТ_11_4.epf</Value>
</SettingsValueNode>
<SettingsValueNode name="ПутьИнтеграционнойОбработкиКонфигурации1С">
  <List count="0" />
  <SortedDictionary>null</SortedDictionary>
  <Value clr:Type="String">C:\ProgramData\Cleverence\Базы Mobile SMARTS\
Склад 15, Базовый 2\Обработки 1С\Обработки интеграции</Value>
</SettingsValueNode>

```

Настройка подключения интеграционной обработки

Если интеграционная обработка не подключена или не имеет поддержку текущей конфигурации или не поддерживает конкретную версию конфигурации, то на главной форме основной обработки ядра «Клеверенс» есть кнопка для возможности выбора и подключения другой интеграционной обработки («Показать настройки» --> «Настроить интеграционную обработку»).

Чтобы включить безусловное отображение кнопки выбора интеграционной обработки включите **полную отладочную информацию «Вся информация»** в расширенных настройках.

Открывается форма выбора интеграционной обработки.

Есть три возможных варианта расположения обработки:

1. Обработка в файле на диске
2. Подключенная обработка в справочнике дополнительных обработок
3. Встроенная в конфигурацию или расширение интеграционная обработка

Как файл на диске (по умолчанию)

← → Форма выбора интеграционной обработки

Версия обработки интеграции:

Конфигурации 1С, поддерживаемые обработкой интеграции:

Версии конфигураций 1С, поддерживаемые обработкой интеграции:

Тип обработки:

 Файл на диске

 Элемент справочника

 Встроенная

Путь к файлу обработки интеграции:

Мы можем выбрать интеграционную обработку из файла на ПК, или в сетевом окружении. Если выполнение обработки будет происходить на сервере 1С, то у владельца службы сервера 1С должны быть права доступа на чтение к папке с интеграционной обработкой.

Как элемент справочника в конфигурации 1С

В конфигурациях 1С есть отдельный справочник с дополнительными отчетами и обработками, добавляем в этот справочник интеграционную обработку и можно использовать её.

← → Форма выбора интеграционной обработки *

Версия обработки интеграции:

Конфигурации 1С, поддерживаемые обработкой интеграции:

Версии конфигураций 1С, поддерживаемые обработкой интеграции:

Тип обработки:

 Файл на диске

 Элемент справочника

 Встроенная

Ссылка на обработку интеграции:

Выбор типа данных ✕

- Группа финансового учета доходов/расходов
- Группа финансового учета номенклатуры
- Группа финансового учета расчетов
- Группировка поддерживаемых запасов
- Данные информационного центра
- Договор кредита (депозита)
- Договор с контрагентом
- Дополнительное значение
- Дополнительное значение (иерархия)
- Дополнительные обработчики очереди заданий
- Дополнительный отчет или обработка
- Драйвер оборудования

Выбираем добавленную обработку:

Дополнительные отчеты и обработки (1С:Предприятие)

Дополнительные отчеты и обработки

Наименование
[-] За ассортимент
[-] ИнтеграционнаяОбработка_УТ114
[-] Клеверенс Софт: Загрузка данных с ТСД
[-] Процент за способ доставки
[-] Шаблон сообщения по изменившемуся статусу документа "Заказ Клиента"
[-] Шаблон сообщения, предназначенный для оповещения о выписке счета (SMS).

После выбора отобразится вся информация о выбранной обработке, и её можно будет подключить.

← → Форма выбора интеграционной обработки *

Версия обработки интеграции:
API: v1, 1.0.1.38 Int, 24.04.2019

Конфигурации 1С, поддерживаемые обработкой интеграции:
КА2, ERP2, УТБЕЛЗ, УТКЗЗ, УТ11

Версии конфигураций 1С, поддерживаемые обработкой интеграции:
11.0.7.13-11.4.7.150

Тип обработки:

Ссылка на обработку интеграции:
ИнтеграционнаяОбработка_УТ114

Если информация не отобразилась, значит метаданные подключаемой конфигурации конфликтуют с выбранной интеграционной обработкой.

ВНИМАНИЕ!!! Информация о ссылке на справочник хранится у конкретного пользователя в базе 1С, поэтому для работы в онлайне настраивайте под пользователем, который будет использоваться для подключения к базе 1С.

Как встроенная обработка в конфигурации 1С или расширения

Интеграционную обработку можно встроить в основную конфигурацию или в расширение, и выбрать её в третьем варианте подключения интеграционной обработки.



Форма выбора интеграционной обработки *

Подтвердить

✕ Закрыть

Версия обработки интеграции:

Не подключена

Конфигурации 1С, поддерживаемые обработкой интеграции:

Не подключена

Версии конфигураций 1С, поддерживаемые обработкой интеграции:

Тип обработки:

Файл на диске

Элемент справочника

Встроенная

Обработка:

- Установка обновлений
- Формирование заказов на сборку (разборку) по плану
- Формирование заказов поставщикам
- Формирование заказов поставщику по плану
- Помощник формирования передач и выкупов товаров
- КлеверенсТСД_УправляемыеФормы
- УТ_11_4**

ВНИМАНИЕ!!! Информация о ссылке на справочник хранится у конкретного пользователя в базе 1С, поэтому для работы в онлайне настраивайте под пользователем, который будет использоваться для подключения к базе 1С.

Заключение

В данной статье расписаны все варианты подключения интеграционных обработок, для поддержания всевозможных случаев от самых простых до самых сложных условий работы с учетными базами на платформе «1С: Предприятие». Вы сможете подобрать из них наиболее подходящий вариант подключения интеграционной обработки. Любой из них по-своему хорош.

Не нашли что искали?



Задать вопрос в техническую поддержку

Первый запуск обработки 1С из мастера настройки Mobile SMARTS

Последние изменения: 2024-03-26

При первом запуске в 1С передается параметр-строка вида: «с65ccd10-6daa-496e-92aa-80043476f2e8|C:\ProgramData\Cleverence\Базы Mobile SMARTS\ Магазин 15, Полный\Backup|http://127.0.0.1:10501/c65ccd10-6daa-496e-92aa-80043476f2e8?altip=SERVER_NAME»

Данная строка состоит из 3 подстрок, разделенных символом «/»:

- ID базы — с65ccd10-6daa-496e-92aa-80043476f2e8.
- Путь до папки с бэкапами — C:\ProgramData\Cleverence\Базы Mobile SMARTS\ Магазин 15, Полный\Backup.
- URL базы = протокол + IP сервера Mobile SMARTS + порт + ID базы — http://127.0.0.1:10501/c65ccd10-6daa-496e-92aa-80043476f2e8?altip=SERVER_NAME.

Далее запускается алгоритм проверки существования базы, которая была передана в параметрах запуска и её доступность для подключения. Для этого:

1. Проверяем подключение через **REST API** по URL базы (сначала – с клиента, в случае неудачи – с сервера), затем проверяем совместимость **платформы Mobile SMARTS** с REST API. Если все получилось, значит, текущая база существует и доступна для подключения. Если нет – пытаемся подключиться через **COM** сначала с клиента, в случае неудачи – с сервера.
2. Если URL базы отсутствует в параметре запуска, берем 1-ую подстроку параметра в качестве ID базы и пытаемся подключиться через COM сначала с клиента, в случае неудачи – с сервера.
3. Сохраняем в хранилище общих настроек 1С (XOH) по ключу «Cleverence_РасположениеКомпоненты» переменную ядра «РасположениеКомпоненты» (НаКлиенте или НаСервере).

Если база существует и доступна для подключения:

1. Сохраняем в XOH по ключу «Cleverence_ПараметрыПодключенияMS» переменную ядра «ПараметрыПодключенияMS» – всю структуру параметров подключения базы (ID базы, имя базы, ID приложения, строка подключения, QR, логин, токен, имя ИО, тип установленного продукта, уровень приложения).
2. Сохраняем в XOH по ключу «Cleverence_РежимВыбораБазы» режим выбора базы («СтрокаПодключения» или «ИзСписка»).
3. Сохраняем в XOH по ключу «Cleverence_ТипИнтерфейса» тип интерфейса (REST_API или COM).

Далее производим инициализацию ядра (локального или глобального, в зависимости от того, откуда доступна база **Mobile SMARTS**– с клиента или с сервера):

1. В переменную ядра «КатегорииНастроекMS» получаем категории настроек: сериализуемые, узловые, настройки подключения, настройки справочников, настройки номенклатуры.
2. Получаем пустую структуру настроек в переменную ядра «ТекущиеНастройкиMS» – все возможные ключи настроек (например, «ВыгружатьНоменклатуруСразу», «НастройкиОбменаСправочников», «НастройкиБизнесПроцессов»).

3. Получаем значения настроек из базы Mobile SMARTS и заполняем ими пустую структуру настроек. Если выбран какой-либо текущий узел для текущей базы 1С, то получаем настройки из базы Mobile SMARTS, привязанной к этому узлу (это может быть как **частная база**, так и основная база Mobile SMARTS).
4. В переменную ядра «СтруктураМетаданных» сохраняем метаданные документов 1С, метаданные документов MS, пользователи MS, устройства MS, таблицы MS, UniПолейНоменклатуры.
5. В переменную ядра «ДанныеТекущегоУзла» получаем Ид, Имя узла и Ид базы MS, привязанной к этому узлу.
6. В переменную ядра «ПраваПользователя» получаем права пользователя, например, «Администратор», «Пользователь». Механизм пока в разработке, является заготовкой для своей системы разграничения прав.
7. Выполняем обновление структуры настроек, если структура настроек изменилась.
8. Временная заплатка — объединяем произвольные коды, если произвольные коды лежат в 4-х ключах:
9. ПроизвольныеКоды_Выгрузка_ШапкаДокумента. ПроизвольныеКоды_Выгрузка_ТабличнаяЧастьДокумента. ПроизвольныеКоды_Загрузка_ШапкаДокумента. ПроизвольныеКоды_Загрузка_ТабличнаяЧастьДокумента.
10. Проверяем обновление типовых **бизнес-процессов**.
11. Собираем общую структуру настроек бизнес-процессов из ключей «НастройкиБизнесПроцессов_ПоУмолчанию», «НастройкиБизнесПроцессов_Пользовательские», «НастройкиБизнесПроцессов_ПользовательскиеНастройки» в один ключ «НастройкиБизнесПроцессов» и сохраняем в кэш.
12. Собираем общую структуру произвольных кодов из ключей «ПроизвольныеКоды_ПоУмолчанию», «ПроизвольныеКоды_Пользовательские» в один ключ «ПроизвольныеКоды» и сохраняем в кэш.

Завершающие действия

- Выгрузить бизнес-процессы в таблицу Mobile SMARTS «БизнесПроцессы».
- Выгрузить ячейки в таблицу Mobile SMARTS «Ячейки».
- Завершить работу 1С и вернуть управление в мастер настройки Mobile SMARTS.

Не нашли что искали?



Задать вопрос в техническую поддержку

Запуск обработки 1С в режиме офлайн-обмена

Последние изменения: 2024-03-26

Инициализация локального или глобального ядра, в зависимости от того, откуда доступна база **Mobile SMARTS** (с клиента или с сервера). Для глобального ядра будут вызываться методы, аналогичные методам локального ядра.

Открывается главная форма, срабатывает обработчик `ПриОткрытии()`, в нем:

1. создаем форму «`ЛокальныйКонтекст`» и сохраняем ее в переменную ядра «`ЛокКонтекст`»;
2. создаем пустую структуру «`ЛокКонтекст.АдресаПеременных`», куда будут заэкшированы данные, необходимых для работы ядра и **интеграционной обработки**;
3. получаем из хранилища общих настроек 1С (XOH) сохраненные при первом запуске «`ПараметрыПодключенияMS`» - и общую, и пользовательскую настройку;
4. если удалось получить пользовательскую настройку, используем ее, иначе - общую настройку;
5. сохраняем их в структуру «`_ЛокКонтекст.ПараметрыПодключенияMS`»;
6. аналогичным алгоритмом получаем из XOH значение переменной ядра «`РасположениеКомпоненты`», если в XOH отсутствует значение, тогда «`РасположениеКомпоненты`» = «`НаСервере`» (по умолчанию), сохраняем переменную в «`_ЛокКонтекст.РасположениеКомпоненты`»;
7. аналогичным алгоритмом получаем из XOH значение переменной ядра «`ТипИнтерфейса`», сохраняем значение в «`_ЛокКонтекст.ТипИнтерфейса`»;

Здесь не устанавливается значение по умолчанию, поэтому, если в XOH не найден тип интерфейса, то будет «`Неопределено`»).

СтатусОбработки = «`НачалоВыполненияПроцедурыЗапускаОбработки`».

8. Проверяем соединение с сервером MS, стек вызовов:

`_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS()`

если подключение через COM, то:

`_ЛокКонтекст.КомЯдро_ПроверитьСоединениеССерверомSMARTS()` →

`_ЛокКонтекст.КомЯдро_ПодключитьсяКСерверуSMARTS()` - помещение COM-объектов «`TerminalConnector`» и «`StorageConnector`» в соответствующую переменную ядра → `COMЧастнойБазыMS / COMВременнойБазыMS / COMБазыMS`, подключение базы MS к COM-объектам и авторизуемся по токenu, если авторизация сохранена →

`_ЛокКонтекст.КомЯдро_ПолучитьСтруктуруПодключенияБазыSMARTS()` - создание COM-объектов «`TerminalConnector`» и «`StorageConnector`» →

`_ЛокКонтекст.КомЯдро_ПодключитьсяКБазеSMARTS()` - подключение базы Mobile SMARTS к COM-объектам и авторизуемся по токenu, если авторизация сохранена →

`_ЛокКонтекст.КомЯдро_ПодключитьсяКБазеSMARTSНаСервереЗавершение()` - получаем структуру описания базы и проверяем соединение с базой.

если подключение через REST_API, то:

`_ЛокКонтекст.REST_API_ПодключитьсяКБазеSMARTS()` - устанавливаем HTTP-соединение с сервером Mobile SMARTS по строке подключения и выполняем авторизацию HTTP-методом GET

Если требуется [авторизация для подключения к серверу Mobile SMARTS](#) и она не сохранена, тогда открываем форму для ввода логина и пароля и после ее закрытия снова проверяем соединение с сервером Mobile SMARTS - `_ЛокКонтекст.ЛокЯдро_ПроверитьСоединениеССерверомSMARTS()`.

Если не удалось подключиться с пользовательской настройкой, пробуем подключиться с общей настройкой.

Если проверка соединения с сервером Mobile SMARTS прошла успешно, тогда меняем статус обработки:

СтатусОбработки = «БезОшибок»

Иначе - меняем статус обработки:

СтатусОбработки = «РаботаССерверомБазаНедоступна»

Для подключения через COM проверяем, что были созданы COM-объекты `«TerminalConnector»` и `«StorageConnector»`. Далее проверяем, что заполнена строка подключения базы в методе `_ЛокКонтекст.ЛокЯдро_ПроверитьСозданиеКОМОбъекта()`, меняем статус обработки:

СтатусОбработки = «НачалоПодключенияИнтеграционнойОбработкиСообщениеБезопасности»

9. Подключаем интеграционную обработку в методе `_ЛокКонтекст.ЛокЯдро_ПодключитьИнтеграционнуюОбработку()`, результаты помещаем в соответствующие переменные ядра `ИмяИнтеграционнойОбработки`, `_ИнтеграционнаяОбработкаАдрес`, `ИнтеграционнаяОбработкаПодключена`, `ФормаИнтеграционнойОбработки`, `ТипОбработки`, `АдресаПеременных.ИнтеграционнаяОбработка`
10. Для подключения через COM проверяем, что созданы COM-объекты `«TerminalConnector»` и `«StorageConnector»`, в методе `_ЛокКонтекст.ЛокЯдро_ПроверитьСозданиеКОМОбъекта()`.

Если COM-объекты не созданы, тогда выдается сообщение пользователю о том, что не удалось подключить внешнюю компоненту
11. Если COM-объекты созданы, но сервер недоступен, открывается форма настройки подключения и меняем статус обработки:

СтатусОбработки = «КоннекторMSSозданПроверяемДоступностьБазы»

Если COM-объекты созданы, сервер доступен, но недоступна текущая база, открываем форму настройки подключения

Если COM-объекты созданы, сервер доступен и доступна текущая база, тогда запускаем инициализацию ядра, метод `_ЛокКонтекст.ЛокЯдро_Инициализация()`.

Начало инициализации ядра

1. В переменную ядра `«КатегорииНастроекMS»` получаем категории настроек: сериализуемые, узловые, настройки подключения, настройки справочников, настройки номенклатуры.
2. Получаем пустую структуру настроек в переменную ядра `«ТекущиеНастройкиMS»` – все возможные ключи настроек (например, `«ВыгружатьНоменклатуруСразу»`, `«НастройкиОбменаСправочников»`, `«НастройкиБизнесПроцессов»`).
3. Получаем значения настроек из базы Mobile SMARTS и заполняем ими пустую структуру настроек. Если

выбран какой-либо текущий узел для текущей базы 1С, то получаем настройки из базы Mobile SMARTS, привязанной к этому узлу (это может быть как [частная база](#), так и основная).

4. В переменную ядра «СтруктураМетаданных» сохраняем метаданные документов 1С, метаданные документов Mobile SMARTS, пользователи, устройства, таблицы Mobile SMARTS, UniПолейНоменклатуры.
5. В переменную ядра «ДанныеТекущегоУзла» получаем ID, Имя узла и ID базы Mobile SMARTS, привязанной к этому узлу.
6. В переменную ядра «ПраваПользователя» получаем права пользователя, например, «Администратор», «Пользователь». Механизм пока в разработке, является заготовкой для своей системы разграничения прав.
7. Если инициализация проходит в локальном контексте, то вызываем метод «Глядро_ОтправитьДанныеДляИнициализацииВГлКонтекст».
8. Выполняем обновление структуры настроек, если структура настроек изменилась (описание механизма обновления структуры настроек приведено в [статье](#)).
9. Временная заплатка: объединяем произвольные коды, если произвольные коды лежат в 4-х ключах:

ПроизвольныеКоды_Выгрузка_ШапкаДокумента.
 ПроизвольныеКоды_Выгрузка_ТабличнаяЧастьДокумента.
 ПроизвольныеКоды_Загрузка_ШапкаДокумента.
 ПроизвольныеКоды_Загрузка_ТабличнаяЧастьДокумента.

10. Проверяем обновление типовых бизнес-процессов.
11. Собираем общую структуру настроек бизнес-процессов из ключей «НастройкиБизнесПроцессов_ПоУмолчанию», «НастройкиБизнесПроцессов_Пользовательские», «НастройкиБизнесПроцессов_ПользовательскиеНастройки» в один ключ «НастройкиБизнесПроцессов» и сохраняем в кэш.
12. Собираем общую структуру произвольных кодов из ключей «ПроизвольныеКоды_ПоУмолчанию», «ПроизвольныеКоды_Пользовательские» в один ключ «ПроизвольныеКоды» и сохраняем в кэш.
13. Заполняем список выбора узлов (торговых объектов) на главной форме для возможности смены узла для текущей базы 1С.
14. Устанавливаем заголовок формы в зависимости от продукта — «Магазин 15», «Склад 15» и т.д.
15. Устанавливаем ссылку на [базу знаний](#) в зависимости от продукта — [«Магазин 15»](#), [«Склад 15»](#) и т.д.
16. Устанавливаем QR-код на главной форме для быстрого подключения к нужной базе путем сканирования кода терминалом сбора данных.
17. Устанавливаем статус обработки: Конец инициализации ядра.

СтатусОбработки = «БезОшибок»

Обновление информации о интеграционной обработке (ИО)

Если [интеграционная обработка](#) не соответствует конфигурации (версии конфигурации) или её версия не подходит для текущей версии обработки-ядра, то меняем статус обработки и меняем текст и цвет статуса на соответствующий.

Например, СтатусОбработки = «ПодключеннаяИнтеграционнаяОбработкаДляДругойКонфигурации», цвет текста - томатный и т.д. для того чтобы пользователь понял, в чем ошибка, насколько она критична и как ее можно устранить.

Если интеграционную обработку не удалось подключить, то аналогично, меняем статус (“ПодключеннаяИнтеграционнаяОбработкаНеТотФайл”, “ОшибкаПриЗапускеИнтеграционнойОбработки”) и выводим соответствующее сообщение об ошибке.

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Запуск обработки 1С в режиме онлайн-обмена

Последние изменения: 2024-03-26

Только для глобального ядра, т.к. в [онлайн-режиме 1С](#) запускается в режиме внешнего соединения через COM-коннектор.

При запуске сервером Mobile SMARTS внешнего соединения к 1С через COM-коннектор срабатывает обработчик глобального ядра «ИнициализацияПодключенияСМАРТС()», на вход приходит параметр «mXmlDoc» - строка в формате JSON, содержащая информацию о текущей сессии Mobile SMARTS (если в настройках Mobile SMARTS установлен режим JSON). Результатом функции будет либо строка «xml» либо строка «json» либо «Истина» в зависимости от значения входного параметра. В обработчик вызывается метод глобального ядра «Глядро_Инициализация()», алгоритм его работы практически идентичен аналогичному методу локального ядра «_ЛокКонтекст.ЛокЯдро_Инициализация()».

1. Создать переменную ядра «АдресаПеременных».
2. Загрузить из хранилища общих настроек 1С (XOH) значения переменных ядра «ПараметрыПодключенияMS» и «ТипИнтерфейса», дополнительно сохранить их значения в «АдресаПеременных» по соответствующим ключам.
3. Сохранить в «АдресаПеременных» флаг «ИспJSON_mXmlDoc», означающий, что для обмена будет использоваться REST_API и формат данных JSON.
4. Очистить переменные ядра «COMЧастнойБазыMS / COMВременнойБазыMS / COMБазыMS».
5. Пытаемся подключиться к базе Mobile SMARTS методом «Глядро_ПроверитьСоединениеССерверомSMARTS()», если неуспешно, то заканчиваем инициализацию, иначе:
6. Получаем настройки Mobile SMARTS для инициализации ядра: либо из кэша, если прошло не более 900 секунд с момента последнего получения настроек или настройки не изменились, в противном случае - получаем настройки из [базы Mobile SMARTS](#).

Начало инициализации ядра

1. В переменную ядра «КатегорииНастроекMS» получаем категории настроек: сериализуемые, узловые, настройки подключения, настройки справочников, настройки номенклатуры.
2. Получаем пустую структуру настроек в переменную ядра «ТекущиеНастройкиMS» – все возможные ключи настроек (например, «ВыгружатьНоменклатуруСразу», «НастройкиОбменаСправочников», «НастройкиБизнесПроцессов»).
3. Получаем значения настроек из базы Mobile SMARTS и заполняем ими пустую структуру настроек. Если выбран какой-либо текущий узел для текущей базы 1С, то получаем настройки из базы Mobile SMARTS, привязанной к этому узлу (это может быть как [частная база Mobile SMARTS](#), так и основная).

Для онлайн дополнительно сохраняем эти данные в XOH для быстрого доступа к ним впоследствии.

4. В переменную ядра «СтруктураМетаданных» сохраняем метаданные документов 1С, метаданные документов MS, пользователи MS, устройства MS, таблицы MS, UniПолейНоменклатуры.

Для онлайн дополнительно сохраняем эти данные в ХОН для быстрого доступа к ним впоследствии.

5. В переменную ядра «ДанныеТекущегоУзла» получаем Ид, Имя узла и Ид базы MS, привязанной к этому узлу.
6. В переменную ядра «ПраваПользователя» получаем права пользователя, например, «Администратор», «Пользователь». Механизм пока в разработке, является заготовкой для своей системы разграничения прав.
7. В переменную ядра «АдресаПеременных» копируем значения переменных «КатегорииНастроекMS», «ТекущиеНастройкиMS», «СтруктураМетаданных», «ДанныеТекущегоУзла».
8. Выполняем обновление структуры настроек, если структура настроек изменилась (описание механизма обновления структуры настроек приведено в [статье](#)).
9. Временная заплатка: объединяем произвольные коды, если произвольные коды лежат в 4-х ключах:
 ПроизвольныеКоды_Выгрузка_ШапкаДокумента, ПроизвольныеКоды_Выгрузка_ТабличнаяЧастьДокумента, ПроизвольныеКоды_Загрузка_ШапкаДокумента, ПроизвольныеКоды_Загрузка_ТабличнаяЧастьДокумента
10. Проверяем обновление типовых **бизнес-процессов**.
11. Собираем общую структуру настроек бизнес-процессов из ключей «НастройкиБизнесПроцессов_ПоУмолчанию», «НастройкиБизнесПроцессов_Пользовательские», «НастройкиБизнесПроцессов_ПользовательскиеНастройки» в один ключ «НастройкиБизнесПроцессов» и сохраняем в кэш.
12. Собираем общую структуру произвольных кодов из ключей «ПроизвольныеКоды_ПоУмолчанию», «ПроизвольныеКоды_Пользовательские» в один ключ «ПроизвольныеКоды» и сохраняем в кэш.

Конец инициализации ядра

1. Сохраняем в ХОН дату и время последнего получения настроек из базы Mobile SMARTS по ключу «Cleverence_ДатаАктуальностиКэша».
2. Подключаем интеграционную обработку в методе «Глядро_ПодключитьИнтеграционнуюОбработку()», результаты помещаем в переменную ядра «АдресаПеременных», по ключу «ИнтеграционнаяОбработка» сохраняем адрес [интеграционной обработки](#) во временном хранилище, путь к файлу, имя ИО. По ключу «ИнтеграционнаяОбработкаПодключена» сохраняем значение «Истина».

Для онлайн-обмена существует отдельная переменная глобального ядра «КэшАдресаПеременных», которая является копией переменной локального ядра «АдресаПеременных».

Не нашли что искали?



Задать вопрос в техническую поддержку

Технология вызова произвольных (своих) функций в интеграционной обработке вызовом из Mobile SMARTS

Последние изменения: 2024-03-26

Новый функционал дает возможность запрашивать произвольные наборы данных в **онлайн-режиме** без внесения изменений в **основную обработку**.

Реализован он за счет добавления общей функции **ВызовПроизвольнойФункции** в основную обработку, которая является проксирующей и вызывает написанную функцию из **интеграционной обработки**.

Благодаря этому не нужно изучать, как работает основная обработка и искать, где выполняются функции, в которые нужно вставить свой код. Теперь можно ограничиться внесением изменений только в интеграционную обработку.

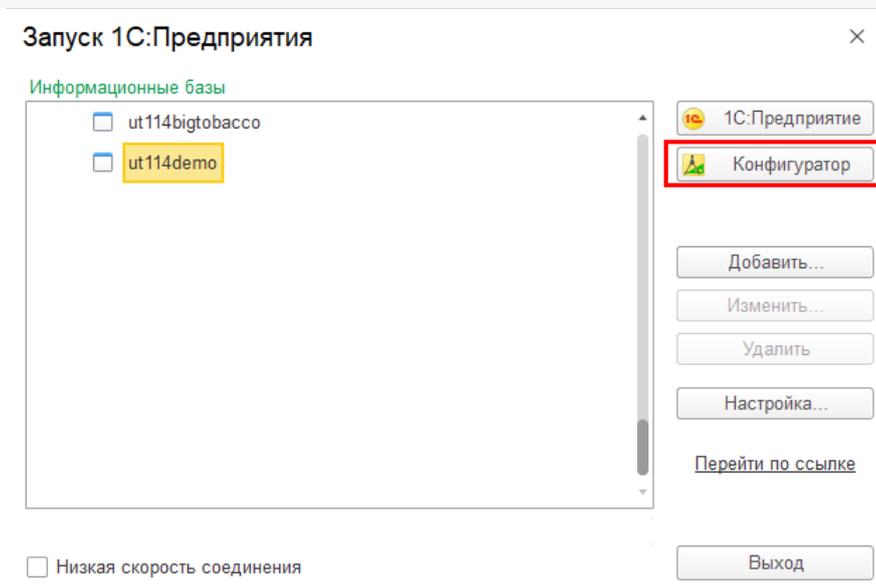
Данный функционал будет полезен партнерам «Клеверенса», внедренцам, делающим доработки, а также программистам, которые хотят добавить функционал, использующий онлайн-вызовы к 1С.

Предварительная подготовка

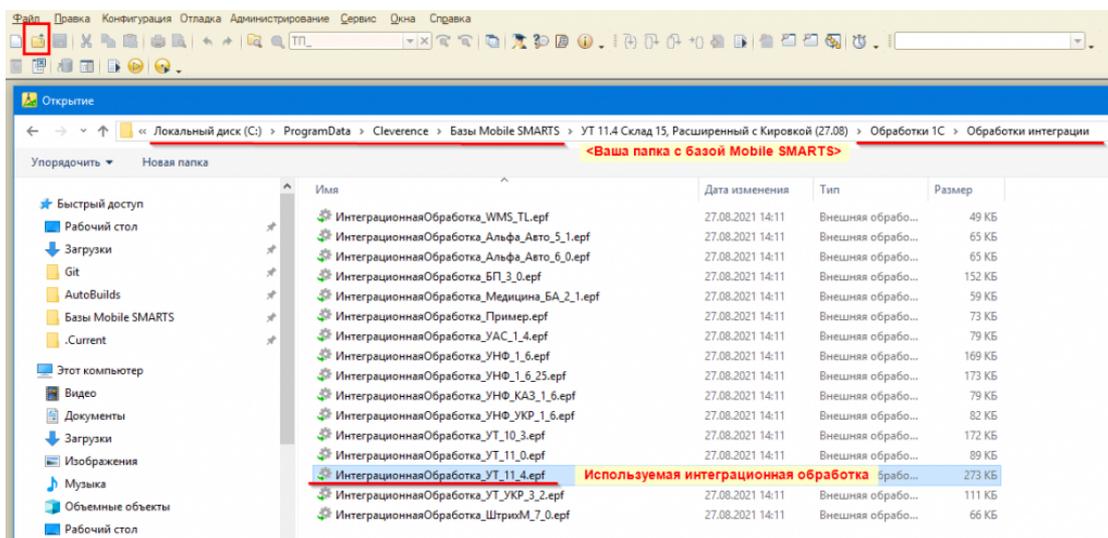
1. Включить работу в режиме онлайн через web-сервисы или COM-соединение.
2. Через конфигуратор 1С открыть интеграционную обработку и добавить в её модуль свою функцию с необходимыми входящими параметрами и возвращаемым результатом. Обязательно сделать её экспортной.
3. Добавить вызов метода внешней системы «ВызовПроизвольнойФункции» в конфигурацию Mobile SMARTS там, где необходимо обратиться за данными в 1С, указав в параметрах имя своей добавленной функции и тип возвращаемых данных.

Настройка

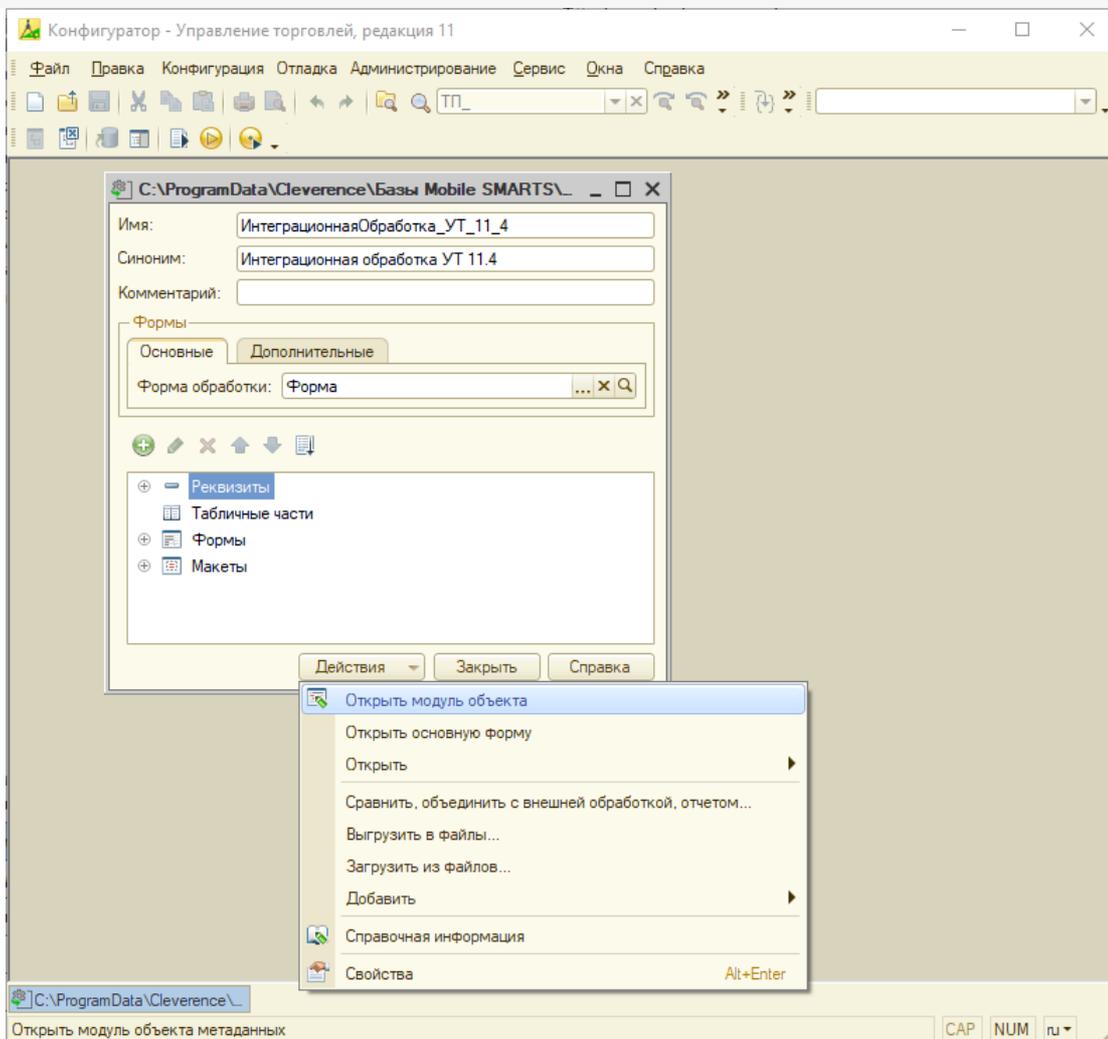
1. Для работы внешних вызовов обязательно должен быть включен **онлайн-обмен с 1С**.
2. Для добавления функции, вызываемой с мобильного устройства, в интеграционную обработку, откройте конфигуратор «1С: Предприятия».



3. Откройте интеграционную обработку из папки с базой Mobile SMARTS.



4. Откройте общий модуль объекта интеграционной обработки и добавьте свою функцию, которая обязательно должна быть экспортной.



```

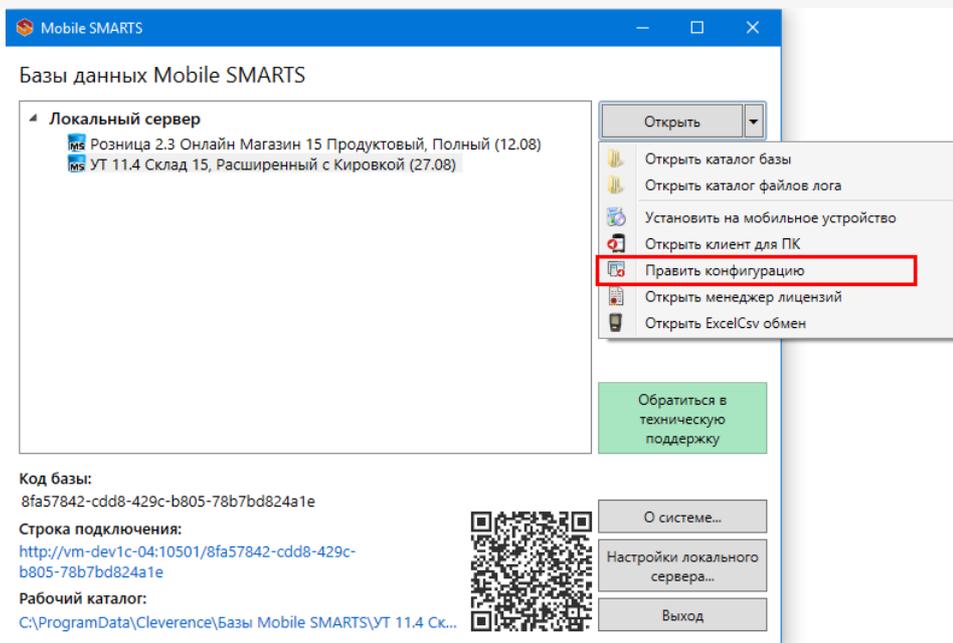
функция ПолучитьОписаниеНоменклатурыПоАртикулу (АртикулСтрокой) Экспорт
    Результат = "Товар не найден";

    Запрос = Новый Запрос;
    Запрос.Текст = "
    |ВЫБРАТЬ
    | Номенклатура.Наименование КАК Наименование,
    | Номенклатура.Описание КАК Описание
    |ИЗ
    | Справочник.Номенклатура КАК Номенклатура
    |ГДЕ
    | Номенклатура.Артикул = &Артикул";
    Запрос.УстановитьПараметр ("Артикул", АртикулСтрокой);

    Выборка = Запрос.Выполнить().Выбрать();
    Если Выборка.Следующий() Тогда
        Результат = Выборка.Наименование+"": "+Выборка.Описание;
    КонецЕсли;
    Возврат Результат
Конецфункции

```

5. Сохраните изменения и обновите в справочнике «Дополнительные отчеты и обработки» интеграционную обработку. Для чего это делается, читайте в [статье](#).
6. Запустите [панель управления Mobile SMARTS](#).



7. Добавьте в нужное место конфигурации вызов внешнего метода. **Настройте его, указывая имя функции, тип значения и передаваемые параметры** (до 10 параметров).

Параметры функции «ВызовПроизвольнойФункции»

- **ИмяФункции** (строка) — наименование функции в интеграционной обработке.
- **ТипВозвращаемогоЗначения** (строка) — **тип объекта Mobile SMARTS**.
- **Параметр1.Параметр10** (произвольный) — параметры, передаваемые из Mobile SMARTS в 1С.

Рассмотрим использование данной функции на примере конфигурации 1С «**Управление торговлей 11.4**» и программного продукта «**Склад 15**». Будем считать, что подключение к «1С: Предприятию» уже выполнено и настроен онлайн-режим через **web-сервисы**.

Пример 1

Задача: получить описание номенклатуры из 1С по артикулу на ТСД.

Решение: создаем в интеграционной обработке экспортную функцию `ПолучитьОписаниеНоменклатурыПоАртикулу`, которая по входящему параметру `АртикулСтрокой` будет возвращать нам наименование и описание найденного товара. В случае если товар не найден, будет возвращаться строка «Товар не найден».

```

функция ПолучитьОписаниеНоменклатурыПоАртикулу (АртикулСтрокой) Экспорт
    Результат = "Товар не найден";

    Запрос = Новый Запрос;
    Запрос.Текст = "
    |ВЫБРАТЬ
    | Номенклатура.Наименование КАК Наименование,
    | Номенклатура.Описание КАК Описание
    |ИЗ
    | Справочник.Номенклатура КАК Номенклатура
    |ГДЕ
    | Номенклатура.Артикул = &Артикул";
    Запрос.УстановитьПараметр ("Артикул", АртикулСтрокой);

    Выборка = Запрос.Выполнить ().Выбрать ();
    Если Выборка.Следующий () Тогда
        Результат = Выборка.Наименование+": "+Выборка.Описание;
    КонецЕсли;
    Возврат Результат
Конецфункции

```

В панели управления **Mobile SMARTS** добавим поле для ввода артикула в переменную **АртикулДляПоиска**. Полученный результат должен выводиться через переменную **ОписаниеТовара**.

The screenshot displays the configuration for a menu item in the Mobile SMARTS application. The left pane shows the 'Алгоритм' (Algorithm) section with the following configuration:

- Меню:** [Товары<г size="-1" color="Red"> {Curr...} [Склады]
- Информация о номенклатуре:** (BarcodeData, ScannedBarcode, SelectedProduct, ВозможенСкан = true)
- Склады:** f() (Склад, ВыходПоКнопке) = Выбор объекта из списка (ItemsCollection = {Склады}, SelectedItem, ВыборПоШК = false, ПоискПоИмени = true, ПросмотрИнфоПоВыбору = true, ...)
- Контрагенты:** f() (Склад, ВыходПоКнопке) = Выбор объекта из списка (ItemsCollection = {Контрагенты}, SelectedItem, ВыборПоШК = false, ПоискПоИмени = true, ПросмотрИнфоПоВыбору = true, ...)
- Получить описание:** f(АртикулДляПоиска, SelectedItem) = Редактирование поля или выбор из списка (СписокДокументов)
- Ескаре:** 1
- ОписаниеТовара:** = OneC_Connector:ВызовПроизвольнойФункции (ПолучитьОписаниеНоменклатурыПоАртикулу, ТипЗначения, АртикулДляПоиска)

The right pane shows the 'Свойства' (Properties) section for the menu item:

- Главное:** Имя: Получить описание номенклатуры
- Ввод данных:** EanAI: None; Запоминать прошлое значение: Нет; Изменять регистр: Нормальный; Имя редактируемого поля: АртикулДляПоиска; Источник данных: Поле в сессии; Тип поля: String; Установить для всех строк в буфере: Нет; Формат редактируемого поля:
- Данные:** Запрос: select (*) from СписокДокументов; Строки для вывода: {СписокДокументов}
- Интерфейс:** Выводить список значений для выбора: Да; Заголовок окна: Последние документы поступления; Заголовок списка: {Коллекция}; Колонки: Завянит от старшего; Полноэкранный режим: Завянит от старшего; Полноэкранный режим (значение): Нет; Ручное редактирование: Да; Текст в верхней части окна: {ОписаниеТовара}; Текст в нижней части окна: {ОписаниеТовара}; Текст в самом окне: Введите артикул; Формат заполнения редактируемого поля:
- Прочее:** Время бездействия: 0; Комментарий; Переход по бездействию; Фоновое изображение
- Управление:** Следующее действие
- Управление отменой по Ескаре и стеком отмен:** Заполнить на стеке: Да; Отмена действия по Ескаре: 1

После ввода артикула добавим вызов метода внешней системы «ВызовПроизвольнойФункции» со следующими параметрами:

- ПолучитьОписаниеНоменклатурыПоАртикулу — наименование нашей функции из интеграционной обработки.
- Пустая строка, т.к. нам не нужно преобразование в объект Mobile SMARTS, ожидаемый результат — строковое значение.
- {АртикулДляПоиска} — введенный артикул.

The screenshot shows a code editor on the left with several function definitions in Russian. The main focus is a 'Параметры вызова' (Call Parameters) dialog box. It contains a table with the following data:

Имя	Тип	Значение
ИмяФункции	String	ПолучитьОписаниеНоменклатурыПоАртикулу
ТипВозвращаемогоЗначения	String	
Артикул	String	{АртикулДляПоиска}

Buttons 'OK' and 'Отмена' are visible at the bottom of the dialog.

On the right, a properties window is visible with sections like 'Главное', 'Вызов', 'Данные', 'Интерфейс', 'Прочее', and 'Управление'.

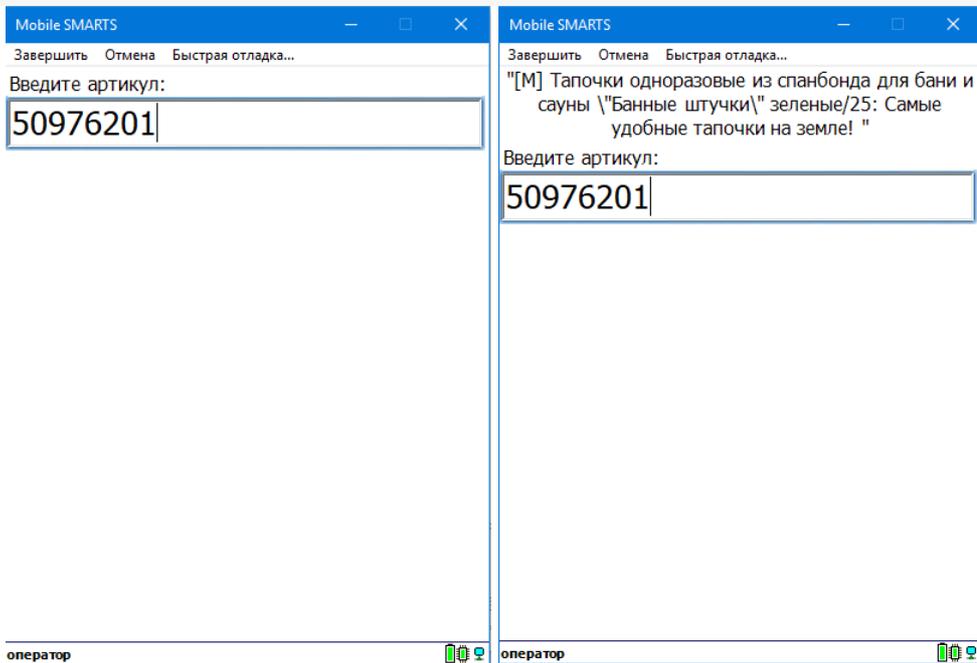
Перейдем к выполнению:

1. В «1С: Предприятии» заведена номенклатура с артикулом 50976201.

The screenshot shows the '1С: Предприятие' interface. The main title is '[M] Тапочки одноразовые из спанбонда для бани и сауны "Баннные штучки" зеленые/25 (Номенклатура)'. Below the title are buttons for 'Записать и закрыть', 'Печать', 'Отчеты', 'Перейти', and 'Выгрузить в Mobile SMARTS'. The 'Карточка' (Card) tab is active, showing details for the commodity:

- Рабочее наименование: [M] Тапочки одноразовые из спанбонда для бани и сауны "Баннные штучки"
- Наименование для печати: [M] Тапочки одноразовые из спанбонда для бани и сауны "Баннные штучки"
- Артикул: 50976201 (highlighted with a red box)
- Код: 50976201
- Штрихкоды (0)
- Описание: Самые удобные тапочки на земле! (highlighted with a red box)
- Additional parameters: Основные параметры учета, Единицы измерения и условия хранения, Регламентированный и финансовый учет, Общероссийские классификаторы.

2. Вводим этот артикул в строку поиска на ТСД, после чего на экране отображается информация о товаре по данному артикулу.



Пример 2

Задача: найти товар по артикулу и отобразить на ТСД 10 последних приходных документов с этим товаром. Необходимо вывести дату, номер, контрагента и количество этого товара из документов.

Решение: создаем в интеграционной обработке экспортную функцию `ПолучитьПоследниеПоступленияПоТовару`, которая по входящему параметру «Артикул» будет формировать таблицу значений с колонками «Дата», «Номер», «Контрагент», «Количество».

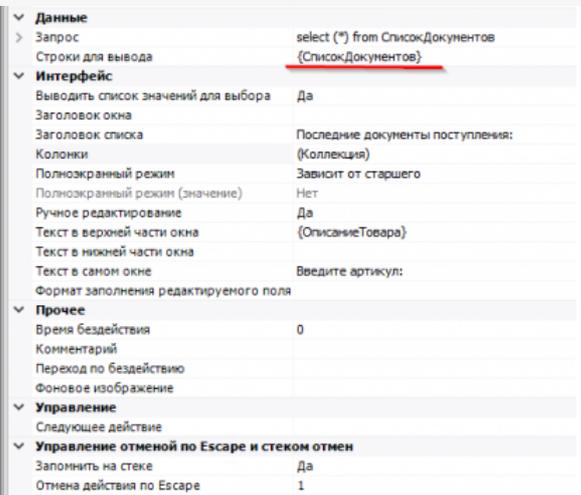
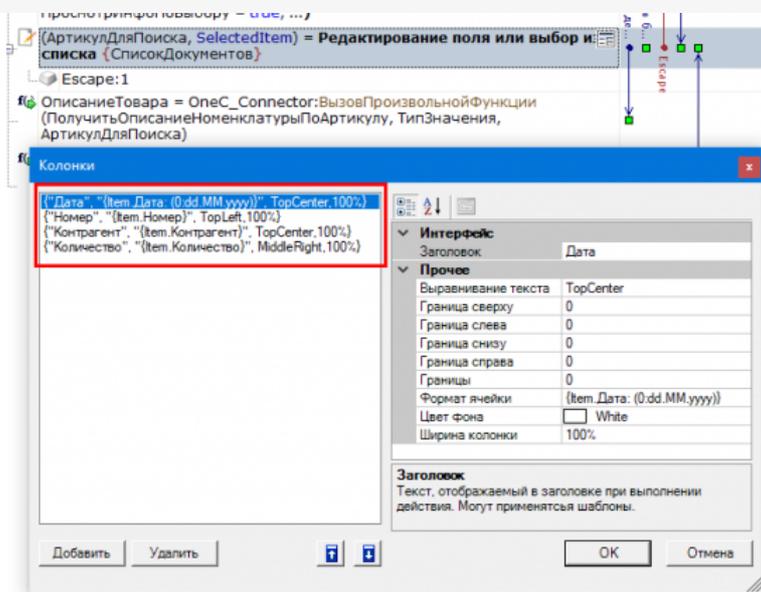
```

Функция ПолучитьПоследниеПоступленияПоТовару (Артикул) Экспорт

    Запрос = Новый Запрос;
    Запрос.Текст = "
    |ВЫБРАТЬ ПЕРВЫЕ 10
    | ПоступлениеТоваровНаСкладТовары.Ссылка.Дата КАК Дата,
    | ПоступлениеТоваровНаСкладТовары.Ссылка.Номер КАК Номер,
    | ПоступлениеТоваровНаСкладТовары.Ссылка.Контрагент.Наименование КАК Контрагент,
    | ПоступлениеТоваровНаСкладТовары.КоличествоУпаковок КАК Количество
    |ИЗ
    | Документ.ПриобретениеТоваровУслуг.Товары КАК ПоступлениеТоваровНаСкладТовары
    |ГДЕ
    | ПоступлениеТоваровНаСкладТовары.Номенклатура.Артикул = &Артикул
    |УПОРЯДОЧИТЬ ПО
    | Дата УБЫВ";
    Запрос.УстановитьПараметр ("Артикул", Артикул);
    ТаблицаРезультат = Запрос.Выполнить().Выгрузить();

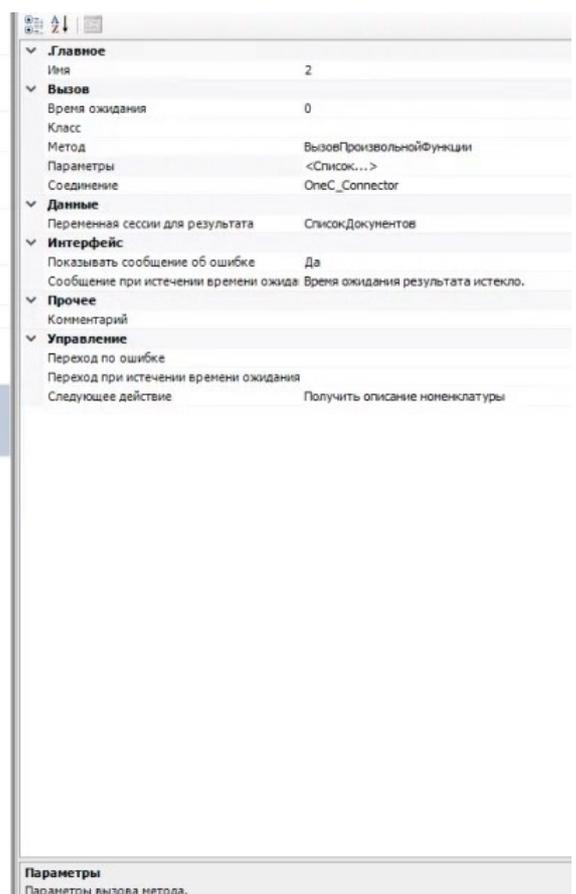
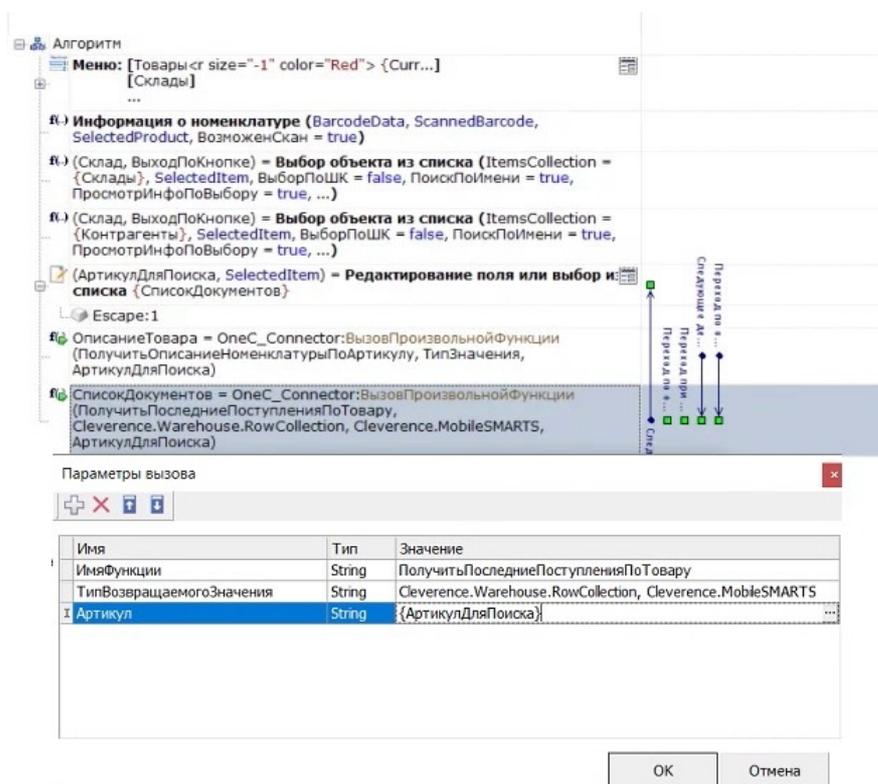
    Возврат ТаблицаРезультат;
Конецфункции
  
```

На форме из примера 1 добавим отображение списка из коллекции «СписокДокументов».



Добавим вызов метода внешней системы «ВызовПроизвольнойФункции» со следующими параметрами:

- ПолучитьПоследниеПоступленияПоТовару — наименование нашей функции из интеграционной обработки.
- Cleverence.Warehouse.RowCollection, Cleverence.MobileSMARTS — т.к. нам в результате нужен объект RowCollection.
- {АртикулДляПоиска} — введенный артикул.



Теперь при запросе артикула на ТСД будет отображаться не только описание товара, но и список последних документов прихода, в котором есть данный товар.

Mobile SMARTS

Завершить Отмена Быстрая отладка...

"Галоши мужские утепленные размер 45: Очень теплые и удобные"

Введите артикул:

10325123

Последние документы поступления:

Дата	Номер	Контрагент	Количество
17.11.2020	0000-000040	АРДАН ООО	8
17.11.2020	0000-000042	СИМА-ЛЕНД ООО ТД	8
13.11.2020	0000-000039	АНТАЛЕКС ООО	2
08.10.2020	0000-000026	АРДАН ООО	4
11.09.2020	0000-000022	АРДАН ООО	8

оператор

Mobile SMARTS

Завершить Отмена Быстрая отладка...

"[М] Тапочки одноразовые из спанбонда для бани и сауны \"Банные штучки\" зеленые/25: Самые удобные тапочки на земле! "

Введите артикул:

50976201

Последние документы поступления:

Дата	Номер	Контрагент	Количество
21.07.2021	0000-000003	ЖАНЕТТ ООО	1

оператор

Не нашли что искали?



Задать вопрос в техническую поддержку

Реализация REST API в обработке 1С

Последние изменения: 2024-03-26

Ключевые тезисы:

- REST API** работает только если при работе с ТСД используется сервер Mobile SMARTS. При прямом **подключении ТСД к компьютеру через кабель/кредл** или при обмене с учетной системой через папку использовать REST API не получится.
- Функционал REST API в **локальном и глобальном ядре** полностью идентичен, т. е. код процедур и функций практически полностью совпадает. Поэтому, если требуются доработки, их нужно делать синхронно в 3 местах — глобальное ядро, локальное ядро УФ, локальное ядро ОФ.
- Обмен данными между 1С и Mobile SMARTS реализован через **HTTP-запросы** из 1С в MS, все запрашиваемые данные передаются в качестве **параметров запроса в формате JSON** (либо в заголовках, либо в теле запроса), а сервер Mobile SMARTS возвращает ответ в виде: кода состояния и тела ответа в формате JSON.
- Коды состояния бывают в диапазоне от 100 до 500 и выше, их можно разделить на следующие группы:
 - 200+: запрос успешен, в обработке используются:
 - 200 — запрос успешно выполнен;
 - 204 — запрос успешно выполнен, но ответ не содержит тела. Используется при выгрузке/очистке справочников Mobile SMARTS, удалении документа Mobile SMARTS, изменении статуса документа Mobile SMARTS.
 - 300+: запрос перенаправлен на другой маршрут;
 - 400+: ошибка на стороне клиента, в обработке используются:
 - 401 — не удалось авторизоваться на сервере Mobile SMARTS для выполнения запроса, например, из-за неверного логина/пароля или по причине того что истек срок действия токена, поэтому потребуется переавторизация.
 - 500+: ошибка на стороне сервера.
- URI — уникальный идентификатор ресурса, используется при каждом запросе. Фактически, это путь до базы вида «`http (s)://[имя сервера]:[порт базы Mobile SMARTS]/api/v1`».
- В обработке для реализации REST API используются прикладные объекты 1С такие как:
 - HTTPСоединение** — для установления соединения с сервером Mobile SMARTS перед отправкой запросов.
 - HTTPЗапрос** — для отправки запросов серверу Mobile SMARTS.
 - HTTPОтвет** — для получения ответов от сервера Mobile SMARTS.
 - ЧтениеJSON** и метод глобального контекста **ПрочитатьJSON ()** — для преобразования ответа от сервера MS в структуру, массив, соответствие и т. д.
 - ЗаписьJSON**, **ПараметрыЗаписиJSON** и метод глобального контекста **ЗаписатьJSON ()** — для формирования тела запроса из параметров в формат JSON.
- В базе Mobile SMARTS может быть включена авторизация, тогда для выполнения HTTP-запросов необходимо будет каждый раз авторизовываться, есть 2 варианта авторизации:
 - BASIC авторизация** — в этом случае при каждом HTTP-запросе нужно будет отправлять логин и пароль

- Авторизация с использованием токена — при первом HTTP-запросе передаются логин и пароль, а сервер MS возвращается токен — уникальный идентификатор сессии (access token), который при последующих HTTP-запросах можно будет использовать вместо передачи логина и пароля. Срок действия токена (т.е. сессии) ограничен, поэтому для обновления токена при первом запросе сервер Mobile SMARTS дополнительно возвращает «токен обновления» (refresh token) — он нужен для получения нового токена после истечения срока действия текущего токена.
8. Для выполнения всех запросов используется единый метод REST_API_ВыполнитьМетод (), который подготавливает запрос к выполнению, в него передаются URI, имя метода, параметры метода и отправляемые данные. Далее, из этого метода вызывается вспомогательный метод «REST_API_ВыполнитьHTTPЗапрос ()», которые устанавливает соединение с сервером MS, сериализует данные для отправки в JSON, выполняет запрос, получает ответ от сервера MS и десериализует его из JSON.
9. Всего при обмене данными между 1С и Mobile SMARTS используются 5 видов HTTP-запросов: GET, POST, PATCH, PUT и DELETE.
- GET — используется для получения со стороны сервера определенного ресурса (настройку Mobile SMARTS, документ, список документов Mobile SMARTS, метаданные базы Mobile SMARTS). Если вы производите этот запрос, сервер ищет информацию и отправляет ее вам назад. По сути, он производит операцию чтения на сервере. Дефолтный тип запросов.
 - POST — нужен для создания определенного ресурса на сервере. Сервер создает в базе данных новую сущность и оповещает вас, был ли процесс создания успешным. По сути, это операции создания настройки Mobile SMARTS, документа Mobile SMARTS, выгрузки справочников Mobile SMARTS.
 - PUT и PATCH — используются для обновления определенной информации на сервере, например, статуса документа Mobile SMARTS. В таком случае сервер просто изменяет информацию существующих сущностей в базе данных и оповещает об успехе выполнения операции.
 - DELETE — как и следует из названия, удаляет указанную сущность из базы (удаление настройки Mobile SMARTS, документа Mobile SMARTS, очистка справочника Mobile SMARTS) или сигнализирует об ошибке, если такой сущности в базе не было.

Функции для обмена данными и HTTP-методы, которые выполняются в следующих функциях:

Название функции в 1С для обмена данными с Mobile SMARTS
HTTP-метод
Параметры
Общие функции
REST_API_ПодключитьсяКБазеSMARTS
GET
нет
REST_API_ПолучитьОписаниеБазы
GET
BaseInfo

REST_API_ПолучитьТокенSMARTS

GET

username = Логин
password = Пароль

Работа с настройками

REST_API_ПолучитьЗначениеНастройки
БазыSMARTS

GET

«CustomSettings («+ КлючНастройки + «)»

REST_API_ЗаполнитьНастройкиSMARTS

GET

CustomSettings

REST_API_ЗаписатьНастройкиSMARTS

POST

CustomSettings
name = КлючНастройки
value = ЗначениеНастройки

REST_API_УдалитьНастройкиSMARTS

DELETE

«CustomSettings («+ КлючНастройки + «)»

Работа со справочниками

**REST_API_ВыгрузитьТаблицуНаСервер
SMARTS****POST**

Для начала выгрузки:

1. для номенклатуры — «Products/BeginUploadProducts»
2. для ячеек — «Cells/BeginUpdate»
3. для прочих справочников — «Tables/"+ИмяТаблицыENG+
4. «/BeginOverwrite»

Для выгрузки данных:

1. для номенклатуры — «Products/AddProductsToUpload»
2. для ячеек — «Cells»
3. для прочих справочников — «Tables/"+ИмяТаблицыENG

При ошибке в процессе выгрузки:

1. для номенклатуры — «Products/ResetUploadProducts»
2. для ячеек — «Cells/ResetUpdate»
3. для прочих справочников — «Tables/"+ИмяТаблицыENG+
4. «/ResetOverwrite»

Для окончания выгрузки

1. для номенклатуры — «Products/EndUploadProducts»
2. для ячеек — «Cells/EndUpdate»
3. для прочих справочников — «Tables/"+ИмяТаблицыENG+"/EndOverwrite»

**REST_API_ОчиститьТаблицуНаСервере
SMARTS****POST**

Аналогично процессу выгрузки, только без п. 2, т. е., фактически не выгружаем в таблицу никаких данных: Для начала очистки:

1. для номенклатуры — «Products/BeginUploadProducts»
2. для ячеек — «Cells/BeginUpdate»
3. для прочих справочников — «Tables/"+ИмяТаблицыENG+
4. «/BeginOverwrite»

Для окончания очистки

1. для номенклатуры — «Products/EndUploadProducts»
2. для ячеек — «Cells/EndUpdate»
3. для прочих справочников — «Tables/"+ИмяТаблицыENG+
4. «/EndOverwrite»

Работа с документами

REST_API_ПолучитьМетаданные ДокументовMS

GET

Получение списка типов документов — «DocTypes?\$expand=tables»

1. Получение реквизитов шапки документа — «DocTypes ('» +
2. СтруктураДокумента.uni + «')?\$expand=fields»
3. Получение реквизитов табличной части документа — «DocTypes ('» +
4. СтруктураДокумента.uni + «')?\$expand=columns»
5. Получение списка доп.таблиц, которые не определены в метаданных
6. документа, но существуют у самого экземпляра документа — «DocTypes ('» + СтруктураДокумента.uni + «')?\$expand=tables (\$expand=fields)»

Получение списка доп.таблиц — «TablesInfo» Получение списка пользователей — «Users» Получение списка устройств — «Devices»

REST_API_ПолучитьСписокДокументовНа СервереSMARTS

GET

«Docs?\$expand=declaredItems,currentItems,
extendedFields»

REST_API_ПолучитьДанныеДокументаНа СервереSMARTS

GET

«Docs/" +UniДокумента+"(" +ID_Документа+"")?\$expand=declaredItems,
currentItems, combinedItems,extendedFields,tables (\$expand=rows)»

REST_API_ЗаписатьДокументВБазуSMARTS

POST

1. Редактировать/добавить документ — «Docs» + данные документа
2. Выгрузить табличную часть, например, declaredItems — «Docs (" +idДокумента+"")/declaredItems»
3. Принудительное сохранение документа, когда все строки уже загружены — «Docs (" +idДокумента+"")/EndUpdate»

REST_API_УдалитьДокументНаСервере SMARTS

DELETE

«Docs (" +ID_Документа+"")»

REST_API_УстановитьСтатусДокументаНа СервереSMARTS

PATCH

«Docs (" +ID_Документа+"")»

Примеры запросов и ответов, используемые при обмене между 1С и Mobile SMARTS:

10.0.0.29

— пример IP-адреса сервера Mobile SMARTS

e1fc20aa-ff42-47df-9e5b-a94ba38b8935

— пример ID базы Mobile SMARTS

REST_API_ПодключитьсяКБазеSMARTS

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1>

Заголовки:

Content-Type: application/json

Тело запроса:

```
нет
HTTP-метод:
GET
Код состояния:
200
Ответ сервера:
```

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata »;
«value»
Массив
Метаданные базы Mobile SMARTS

REST_API_ПолучитьТокенSMARTS

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/session?username=Логин&pa...>

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«Expires_in»
86 400
Срок действия токена в секундах
«Access_token»
«8ee184cb1eb7b4b945a69bbc5bd198a5»
Токен доступа
«Token_type»
«bearer»
Тип токена
«Refresh_token»
«b8140b77fa61fa5399735ab60f7a9f16»
Токен обновления

REST_API_ПолучитьОписаниеБазы

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/BaseInfo>

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«folder»
«C:\ProgramData\Cleverence\Базы Mobile SMARTS\Склад 15, Полный с Кировкой»
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#BaseInfo »;
«id»
«e1fc20aa-ff42-47df-9e5b-a94ba38b8935»
«name»
«Склад 15, Полный с Кировкой»
«connectionQRCode»
« http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935?altip=VM-WIN10-USR-12 »;
«appld»
«212E5362-5969-40DC-9781-CB7B8AB811DB»
«appName»
«Склад 15, Полный с Кировкой»
«allConnectionStrings»
Массив
Все возможные варианты строки соединения с базой
«comment»
«appDescription»
Соответствие
Информация о продукте: наименование, ID приложения, версия платформы, ссылки на интернет-ресурсы, комментарий
«appInstanceSettings»
Соответствие
Информация о настройках сервера

REST_API_ПолучитьЗначениеНастройкиБазыSMARTS

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/CustomSettings> ('ИмяИнтегр...

Заголовки:

Content-Type: application/json

Тело запроса:

```
нет
HTTP-метод:
GET
Код состояния:
200
Ответ сервера:
```

Ключ
Значение
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#CustomSettings/\$entity »;
«value»
«ИнтеграционнаяОбработка_УТ_11_4.epf»
«name»
«ИмяИнтеграционнойОбработкиКонфигурации1С»

REST_API_ЗаполнитьНастройкиSMARTS

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/CustomSettings?\\$filter=name eq 'ИспользоватьШаблоныВесовогоТовара' or name eq 'ПрефиксКодаВесовогоТовара' or name eq 'ДлинаКодаВесовогоТовара' or name eq 'ЧислоЗнаковПослеЗапятойВесТовара' or name eq 'ПрефиксВключенВКодВесовогоТовара' or name eq 'РазмерПорцииНоменклатуры' or name eq 'ВыгружатьНоменклатуруДокументаПередВыгрузкойДокумента' or name eq 'УстановитьПоискПоНаименованиюИАртикулу' or name eq 'ВыгружатьНоменклатуруСразу' or name eq 'ВыгружатьКонтрагентовСразу'](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/CustomSettings?$filter=name eq 'ИспользоватьШаблоныВесовогоТовара' or name eq 'ПрефиксКодаВесовогоТовара' or name eq 'ДлинаКодаВесовогоТовара' or name eq 'ЧислоЗнаковПослеЗапятойВесТовара' or name eq 'ПрефиксВключенВКодВесовогоТовара' or name eq 'РазмерПорцииНоменклатуры' or name eq 'ВыгружатьНоменклатуруДокументаПередВыгрузкойДокумента' or name eq 'УстановитьПоискПоНаименованиюИАртикулу' or name eq 'ВыгружатьНоменклатуруСразу' or name eq 'ВыгружатьКонтрагентовСразу')

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#CustomSettings »;
«value»
Массив
Массив значений запрашиваемых настроек

REST_API_ЗаписатьНастройкиSMARTS

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/CustomSettings>

Заголовки:

Content-Type: application/json

Тело запроса:

```
{
  "value": true,
  "name": "РаботаБезАлкоголя"
}
```

HTTP-метод:
 POST
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#CustomSettings/\$entity »;

«value»
Истина
Значение настройки
«name»
«РаботаБезАлкоголя»
Ключ настройки

REST_API_УдалитьНастройкиSMARTS

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/CustomSettings> ('Настройки...

Заголовки:

Content-Type: application/json

Тело запроса:

```
нет
HTTP-метод:
DELETE
Код состояния:
204
Ответ сервера:
нет
```

REST_API_ВыгрузитьТаблицуНаСерверSMARTS

Будут выполнены 3 запроса: начать выгрузку, выгрузка, завершить выгрузку (или прервать выгрузку, в случае ошибки), пример дан для выгрузки номенклатуры. В случае выгрузки ячеек или дополнительных таблиц будут изменяться только наименования методов (см. таблицу выше), все остальное выполняется по тому же алгоритму

Запрос 1 — Начать выгрузку

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Products/BeginUploadProducts>

Заголовки:

Content-Type: application/json

Тело запроса:

```
{
  "Ischema": [
    "Product.Name",
    "Product.Barcode",
    "Product.ЭтoГруппa"
```

```
Product.Страна",  
"Product.Marking",  
"Product.Весовой",  
"Product.Маркировка",  
"Packing.Barcode",  
"Packing.ИмяХарактеристики",  
"Packing.Name",  
"Packing.UnitsQuantity",  
"Product.Алко",  
"Packing.АлкоКод",  
"Packing.АлкоОбъем",  
"Packing.АлкоКрепость",  
"Product.АлкоМарк",  
"Packing.АлкоПроизводитель",  
"Packing.АлкоПроизВИНН",  
"Packing.АлкоПроизвКПП",  
"Packing.ТНВЭД",  
"Packing.Бренд",  
"Packing.Верх",  
"Packing.Вид",  
"Packing.Импортер",  
"Packing.Размер",  
"Packing.ИНН",  
"Packing.ИсходныйGTIN",  
"Packing.Низ",  
"Packing.Подкладка",  
"Packing.Производитель",  
"Packing.СтранаПроизводства",  
"Packing.Цвет",  
"Packing.price",  
"Packing.qty",  
"Product.ПоХарактеристикам",  
"Product.ПоСериям",  
"Product.ПоСН",  
"Product.Id",  
"Product.BasePackingId",  
"Packing.ИдХарактеристики",  
"Packing.Id",  
"Packing.ИдЕдиницыИзмерения",  
"Product.КлючХарактеристик",  
"Product.КлючСерий",  
"Packing.ИдСерии"  
],  
"rewriteProductsBook": true,  
"useFullTextSearch": true  
}  
HTTP-метод:  
  POST  
Код состояния:  
  204  
Ответ сервера:  
нет
```

Запрос 2 — Начать выгрузку

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Products/AddProductsToUpload>

Заголовки:

Content-Type: application/json

Тело запроса:

```
{
  "flds": [
    "X-77890 Стинол Завод бытовой техники",
    "0000000094",
    "false",
    "X-77890",
    "false",
    "",
    "2000000058078",
    "",
    "ШТ",
    "1",
    "false",
    "",
    "0",
    "0",
    "false",
    "",
    "",
    "",
    "",
    "Стинол",
    "",
    "",
    "",
    "",
    "",
    "",
    "2000000058078",
    "",
    "",
    "Завод бытовой техники",
    "",
    "",
    "0",
    "8",
    "false",
    "false",
    "false",
    "cbcf4934-55bc-11d9-848a-00112f43529a",
```

```
"bd72d926-55bc-11d9-848a-00112f43529a",  
"  
"bd72d926-55bc-11d9-848a-00112f43529a",  
"bd72d926-55bc-11d9-848a-00112f43529a",  
"  
"9c556d55-720f-11df-b436-0015e92f2802",  
"  
]  
}
```

HTTP-метод:

POST

Код состояния:

204

Ответ сервера:

нет

Запрос 3 — Завершить выгрузку / прервать выгрузку

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Products/EndUploadProducts> (для завершения выгрузки)

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Products/ResetUploadProducts> (для прерывания выгрузки)

Заголовки:

Content-Type: application/json

Тело запроса:

```
нет  
HTTP-метод:  
POST  
Код состояния:  
204  
Ответ сервера:  
нет
```

REST_API_ОчиститьТаблицуНаСервереSMARTS

Аналогично процессу выгрузки, только без запроса 2, т. е., фактически, не выгружаем в таблицу никаких данных:

REST_API_ПолучитьМетаданныеДокументовMS — состоит из 7 запросов

Запрос1 — Получение списка типов документов

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes?\\$expand=tables](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes?$expand=tables)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
HTTP-метод:
GET
Код состояния:
200
Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#DocTypes »;
«value»
Массив
Массив метаданных документов Mobile SMARTS

Запрос 2 — Получение метаданных и реквизитов шапки документа на примере документа «Агрегация»**Строка запроса:**[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes \('Агрегация&...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes ('Агрегация&...)**Заголовки:**

Content-Type: application/json

Тело запроса:

нет
HTTP-метод:
GET
Код состояния:
200
Ответ сервера:

Ключ
Значение
Комментарий

«@odata.context»
«http://localhost:18686/MobileSMARTS/api/v1/\$metadata#DocTypes/\$entity»;
«fields»
Массив
Массив реквизитов шапки документа
«uni»
«Agregaciya»
Транслитерация имени документа
«name»
«Агрегация»
Имя документа
«allowRemoveDocuments»
Истина
Разрешить удаление документа с ТСД
«removeAfterFinish»
Ложь
Удалять после завершения с ТСД
...
...
и т. д.

Запрос 3 — Получение метаданных и реквизитов табличной части документа на примере документа «Агрегация»

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes \('Agregaciya&...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes ('Agregaciya&...)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
HTTP-метод:
GET
Код состояния:
200
Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#DocTypes/\$entity »;
«columns»
Массив
Массив реквизитов табличной части документа
«uni»
«Агрегация»
Транслитерация имени документа
«name»
«Агрегация»
Имя документа
«allowRemoveDocuments»
Истина
Разрешить удаление документа с ТСД
«removeAfterFinish»
Ложь
Удалять после завершения с ТСД
...
...
и т. д.

Запрос 4 — Получение списка доп.таблиц, которые не определены в метаданных документа, но существуют у самого экземпляра документа на примере документа «Агрегация»

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes \('Агрегация&...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/DocTypes ('Агрегация&...)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#DocTypes/\$entity »;
«columns»
Массив
Массив табличных частей документа и их реквизитов
«uri»
«Агрегация»
Транслитерация имени документа
«name»
«Агрегация»
Имя документа
«allowRemoveDocuments»
Истина
Разрешить удаление документа с ТСД
«removeAfterFinish»
Ложь
Удалять после завершения с ТСД
...
...
и т. д.

Запрос 5 — Получение списка дополнительных таблиц

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/TablesInfo>

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#TablesInfo »;
«value»
Массив
Массив метаданных дополнительных таблиц Mobile SMARTS

Запрос 6 — Получение списка пользователей — «Users»

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Users>

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#Users »;

«value»
Массив
Массив метаданных пользователей Mobile SMARTS

Запрос 7 — Получение списка устройств — «Devices»

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Devices>

Заголовки:

Content-Type: application/json

Тело запроса:

нет
HTTP-метод:
GET
Код состояния:
200
Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#Devices ";
«value»
Массив
Массив метаданных устройств Mobile SMARTS

REST_API_ПолучитьСписокДокументовНаСервереSMARTS

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs?\\$expand=declaredItems,curren...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs?$expand=declaredItems,curren...)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«@odata.context»
« http://localhost:18686/MobileSMARTS/api/v1/\$metadata#Devices »;
«value»
Массив
Массив данных документов Mobile SMARTS

REST_API_ПолучитьДанныеДокументаНаСервереSMARTS

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs/PrihodNaSklad \('7e38b...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs/PrihodNaSklad ('7e38b...)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
 HTTP-метод:
 GET
 Код состояния:
 200
 Ответ сервера:

Ключ
Значение
Комментарий
«id»
«7e38bff7-8faf-11ea-a40c-00155d03e00e»
GUID документа

«name»
«Приобретение товаров и услуг ТД00-000026 от 06.05.2020 18:37:37»
Имя документа 1С
«nastrojkaBiznesProcessa»
«BPe8d6ec5a-7a78-41ba-8b01-820f78b31cdb»
GUID бизнес-процесса, по которому был выгружен документ
«имяBP»
«Приобретение товаров и услуг (проверка документа на ТСД)»
Имя бизнес-процесса
«states»
Массив
Состояние документа — Выгружен, В работе, Завершен
«declaredItems»
Массив
Табличная часть СтрокиПлан
«currentItems»
Массив
Табличная часть СтрокиФакт
«combinedItems»
Массив
Табличная часть Совмещенные
«extendedFields»
Структура
Дополнительные реквизиты шапки
«tables»
Массив
Дополнительные табличные части
...
...
И т.д.

REST_API_ЗаписатьДокументВБазуSMARTS

Запрос 1 — Выгрузка шапки документа

Строка запроса:

<http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs>

Заголовки:

Content-Type: application/json

Тело запроса:

```

{
  "id": "7e38bff7-8faf-11ea-a40c-00155d03e00e",
  "warehouseId": "1",
  "ИдКонтрагента": "cbcf497b-55bc-11d9-848a-00112f43529a",
  "ИдСклада1С": "03ce4b6f-3ff7-11e0-af05-0015e9b8c48d",
  "ИмяСклада": "Продуктовая база",
  "ИмяКонтрагента": "База \"Продукты\"",
  "DistributeByBarcode": true,
  "barcode": "167777389647035791856893807099027644430",
  "РежимУпаковок": "БезУпаковок",
  "Name": "Приобретение товаров и услуг ТД00-000026 от 06.05.2020 18:37:37",
  "documentTypeName": "ПриходНаСклад",
  "НастройкаБизнесПроцесса": "BPe8d6ec5a-7a78-41ba-8b01-820f78b31cdb",
  "ИмяБП": "Приобретение товаров и услуг (проверка документа на ТСД)",
  "КонтрольКолва": true
}

```

HTTP-метод:
 POST

Код состояния:
 201

Ответ сервера:

Ключ
Значение
Комментарий
«id»
«7e38bff7-8faf-11ea-a40c-00155d03e00e»
GUID документа
«name»
«Приобретение товаров и услуг ТД00-000026 от 06.05.2020 18:37:37»
Имя документа в 1С
«createdOnPDA»
Ложь
Создан на ТСД
«documentTypeName»
«ПриходНаСклад»
Тип документа Mobile SMARTS

«modified»
Ложь
Изменен
«inProcess»
Ложь
В работе
«finished»
Ложь
Завершен
«barcode»
«167777389647035791856893807099027644430»
Штрихкод документа
«serverHosted»
Ложь
Коллективная работа
«idKontragenta»
«cbcf497b-55bc-11d9-848a-00112f43529a»
«idSklada1S»
«03ce4b6f-3ff7-11e0-af05-0015e9b8c48d»
«imyaSklada»
«Продуктовая база»
«imyaKontragenta»
«База «Продукты»»
«rezhimUakovok»
«БезУпаковок»
«nastrojkaBiznesProcessa»
«BPe8d6ec5a-7a78-41ba-8b01-820f78b31cdb»
«imyaBP»
«Приобретение товаров и услуг (проверка документа на ТСД)»

«kontrolKolva»
Истина
Контроль количества
...
...
И т. д.

Запрос 2 — Выгрузка табличных частей документа, например, declaredItems

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs \('7e38bff7-8faf-11ea-...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs ('7e38bff7-8faf-11ea-...)

Заголовки:

Content-Type: application/json

Тело запроса:

```
{
  "value": [
    {
      "packingId": "bd72d8fb-55bc-11d9-848a-00112f43529a",
      "productId": "e8a71fa5-55bc-11d9-848a-00112f43529a",
      "ИдЕдиницыИзмерения": "bd72d8fb-55bc-11d9-848a-00112f43529a",
      "Цена": 41,
      "declaredQuantity": 5
    }
  ]
}
```

HTTP-метод:
POST

Код состояния:
200

Ответ сервера:
нет

Запрос 3 — Принудительное сохранение документа, когда все строки уже загружены

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs \('7e38bff7-8faf-11ea-...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs ('7e38bff7-8faf-11ea-...)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
HTTP-метод:
POST
Код состояния:
204
Ответ сервера:
нет

REST_API_УдалитьДокументНаСервереSMARTS

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs \('7e38bff7-8faf-11ea-...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs ('7e38bff7-8faf-11ea-...)

Заголовки:

Content-Type: application/json

Тело запроса:

нет
HTTP-метод:
DELETE
Код состояния:
204
Ответ сервера:
нет

REST_API_УстановитьСтатусДокументаНаСервереSMARTS

Строка запроса:

[http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs \('7e38bff7-8faf-11ea-...](http://10.0.0.29:10501/e1fc20aa-ff42-47df-9e5b-a94ba38b8935/api/v1/Docs ('7e38bff7-8faf-11ea-...)

Заголовки:

Content-Type: application/json

Тело запроса:

```
{  
  "Status": "ОшибкаЗаписи"  
}
```

HTTP-метод:
PATCH
Код состояния:
204
Ответ сервера:
нет

Не нашли что искали?



Задать вопрос в техническую поддержку

Подключение к Mobile SMARTS через REST API

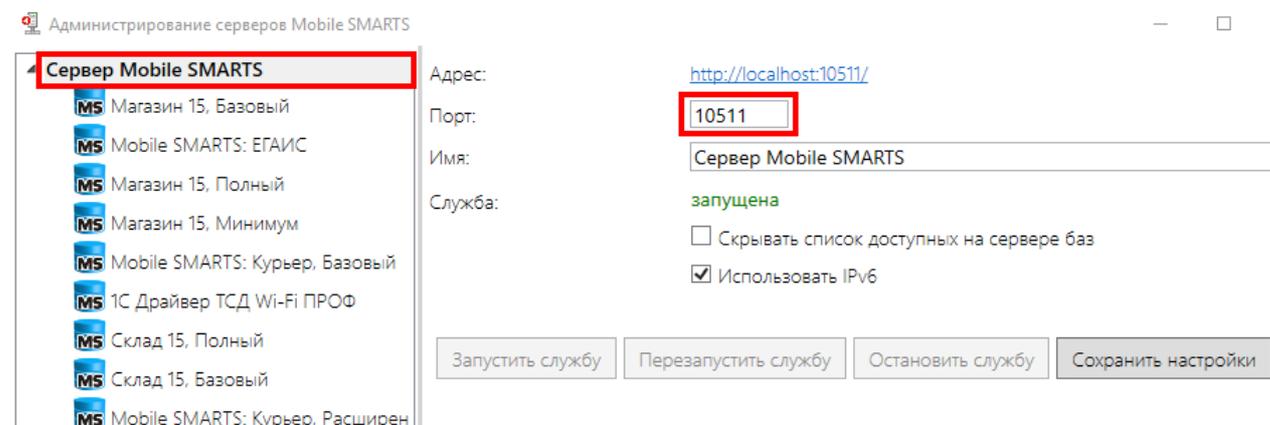
Последние изменения: 2024-03-26

Точкой входа для доступа к API в нашей системе является url вида:

[http\(s\)://{имя сервера}:{порт базы}/api/v1](http(s)://{имя сервера}:{порт базы}/api/v1)

По нему можно получить список методов апи.

Чтобы узнать порт сервера, порт базы или идентификатор базы, надо воспользоваться [приложением для администрирования сервера Mobile SMARTS](#).



Администрирование серверов Mobile SMARTS

Сервер Mobile SMARTS

- Магазин 15, Базовый
- Mobile SMARTS: ЕГАИС
- Магазин 15, Полный
- Магазин 15, Минимум
- Mobile SMARTS: Курьер, Базовый
- 1С Драйвер ТСД Wi-Fi ПРОФ
- Склад 15, Полный
- Склад 15, Базовый
- Mobile SMARTS: Курьер, Расширен

Адрес: <http://localhost:10511/>

Порт: 10511

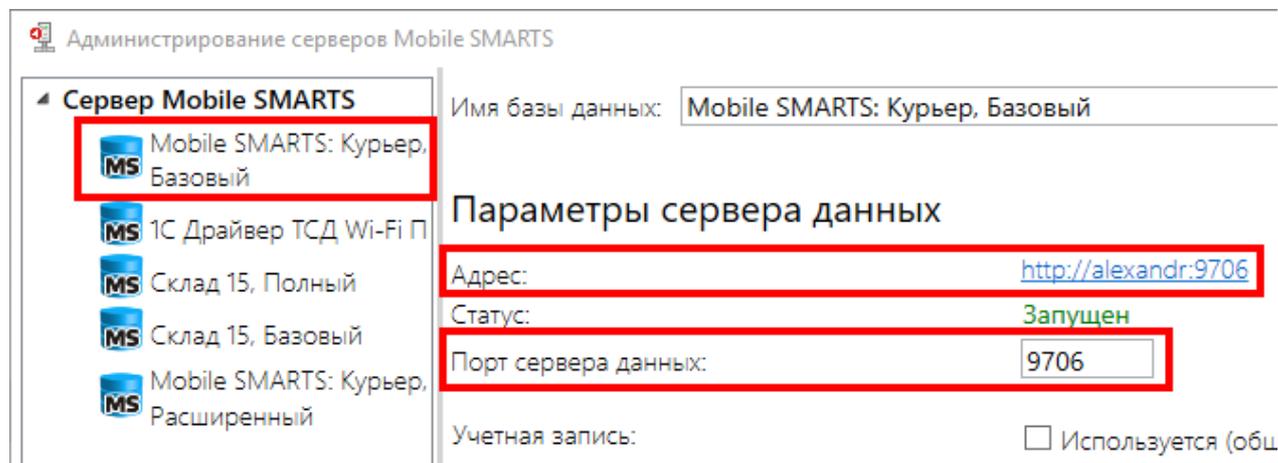
Имя: Сервер Mobile SMARTS

Служба: **запущена**

Скрывать список доступных на сервере баз

Использовать IPv6

Запустить службу | Перезапустить службу | Остановить службу | Сохранить настройки



Администрирование серверов Mobile SMARTS

Сервер Mobile SMARTS

- Mobile SMARTS: Курьер, Базовый
- 1С Драйвер ТСД Wi-Fi П
- Склад 15, Полный
- Склад 15, Базовый
- Mobile SMARTS: Курьер, Расширенный

Имя базы данных: Mobile SMARTS: Курьер, Базовый

Параметры сервера данных

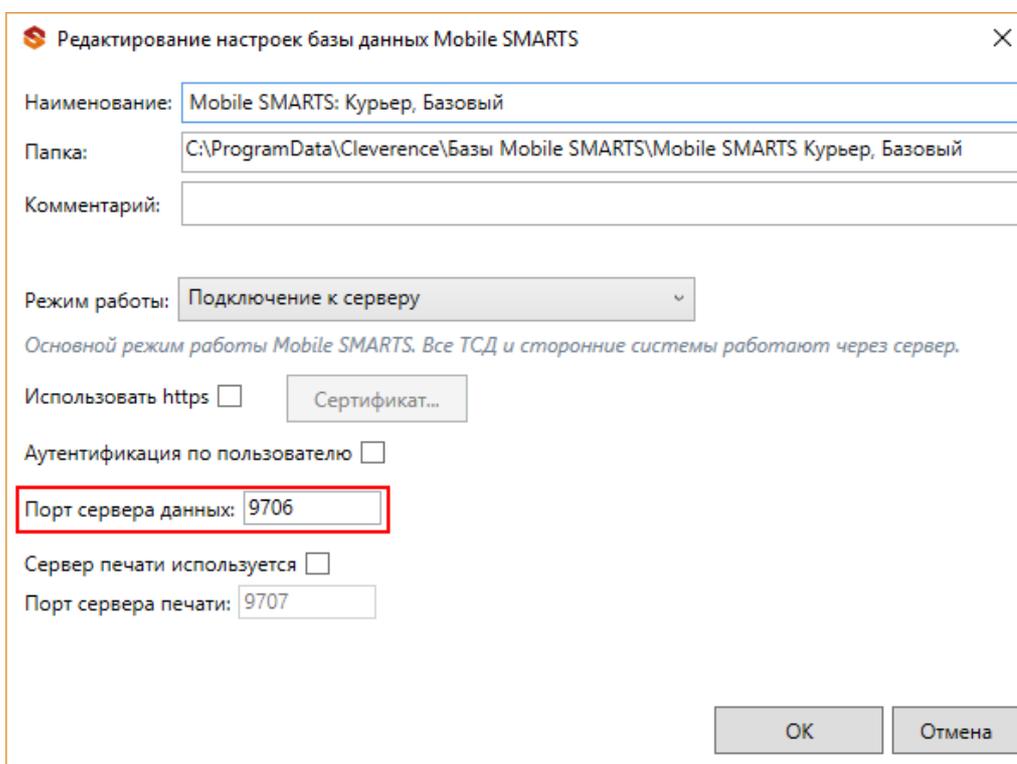
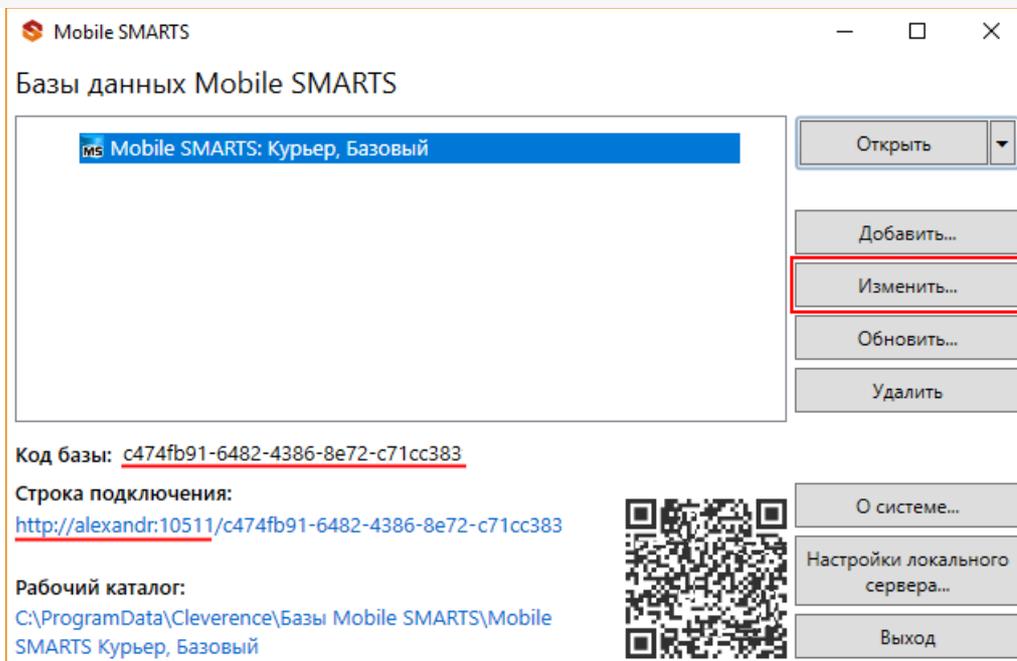
Адрес: <http://alexandr:9706>

Статус: **Запущен**

Порт сервера данных: 9706

Учетная запись: Используется (общ)

Или менеджером баз данных Mobile SMARTS.



Для приведенных скриншотов url получаются такими:

[http\(s\)://localhost:10511/c474fb91-6482-4386-8e72-c71cc383/api/v1](http(s)://localhost:10511/c474fb91-6482-4386-8e72-c71cc383/api/v1)

или

[http\(s\)://localhost:9706/api/v1](http(s)://localhost:9706/api/v1)

 интеграция, REST API

Не нашли что искали?



Задать вопрос в техническую поддержку

Проверка запросов через Swagger

Последние изменения: 2024-03-26

Примеры Swagger для Mobile SMARTS:
 Swagger для «Mobile SMARTS: Магазин 15»
 Swagger для «Mobile SMARTS: Курьер»

В нашей системе также предусмотрена возможность просматривать структуру API и выполнять простые запросы, используя Swagger. Для доступа к нему необходимо в браузере зайти по адресу:

<http://xxx.xxx.xxx.xxx:9000/MobileSMARTS/swagger/ui/index>

По этому адресу открывается страница с описанием всех методов.

 swagger

Mobile SMARTS: Курьер, Базовый API

Devices	Показать/Скрыть	Операции кратко	Операции подробно
Docs	Показать/Скрыть	Операции кратко	Операции подробно
Docs/ChekKorrekcii	Показать/Скрыть	Операции кратко	Операции подробно
Docs/PoluchenieTovara	Показать/Скрыть	Операции кратко	Операции подробно
Docs/SdachaTovara	Показать/Скрыть	Операции кратко	Операции подробно
Docs/Zakaz	Показать/Скрыть	Операции кратко	Операции подробно
DocTypes	Показать/Скрыть	Операции кратко	Операции подробно
Licenses	Показать/Скрыть	Операции кратко	Операции подробно
Products	Показать/Скрыть	Операции кратко	Операции подробно
Tables/Kontragenty	Показать/Скрыть	Операции кратко	Операции подробно
Tables/Oshibki	Показать/Скрыть	Операции кратко	Операции подробно
Tables/Ostatki	Показать/Скрыть	Операции кратко	Операции подробно
Tables/TipyPredmetov	Показать/Скрыть	Операции кратко	Операции подробно
Tables/Valyuty	Показать/Скрыть	Операции кратко	Операции подробно
TablesInfo	Показать/Скрыть	Операции кратко	Операции подробно
UserGroups	Показать/Скрыть	Операции кратко	Операции подробно
Users	Показать/Скрыть	Операции кратко	Операции подробно
Warehouses	Показать/Скрыть	Операции кратко	Операции подробно

[BASE URL: /MobileSMARTS , Версия API: v1]

Каждая группа в этом списке разворачивается и отображает все методы этой группы.

Mobile SMARTS: Курьер, Базовый API

Devices

Показать/Скрыть | Операции кратко | Операции подробно

GET	/api/v1/Devices	Список устройств
POST	/api/v1/Devices	Добавить/отредактировать устройство
DELETE	/api/v1/Devices('{deviceId}')	Удалить устройство
GET	/api/v1/Devices('{deviceId}')	Получить устройство по идентификатору
PATCH	/api/v1/Devices('{deviceId}')	Изменить устройство
PUT	/api/v1/Devices('{deviceId}')	Добавить/отредактировать устройство по известному идентификатору

Docs

Показать/Скрыть | Операции кратко | Операции подробно

Docs/ChekKorrekcii

Показать/Скрыть | Операции кратко | Операции подробно

В каждом методе можно просматривать все входящие и исходящие параметры. Также можно посмотреть описание сущностей, которые отправляются/получаются при использовании этого метода.

Devices

Показать/Скрыть | Операции кратко | Операции подробно

GET	/api/v1/Devices	Список устройств
-----	-----------------	------------------

Пример ответа (Статус 200)
OK

Описание | **Пример**

```

{
  "@odata.context": "string",
  "value": [
    {
      "appInstanceId": "string",
      "deviceId": "string",
      "batteryStatus": "string",
      "lastInfoTime": "2018-03-20T06:49:48.982Z",
      "userId": "string",
      "warehouseId": "string",
      "documentId": "string"
    }
  ]
}

```

Content Type ответа

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
\$expand	<input type="text"/>	Expands related entities inline.	query	string
\$filter	<input type="text"/>	Filters the results, based on a Boolean condition.	query	string
\$select	<input type="text"/>	Selects which properties to include in the response.	query	string
\$orderby	<input type="text"/>	Sorts the results.	query	string
\$top	<input type="text"/>	Returns only the first n results.	query	integer
\$skip	<input type="text"/>	Skips the first n results.	query	integer
\$count	<input type="text"/>	Includes a count of the matching results in the response.	query	boolean

Также можно получить json схемы всех методов api и по нему генерировать модели в своем приложении.

<http://localhost:{порт базы}/swagger/docs/v1> - получение json сваггера.

Не нашли что искали?



Задать вопрос в техническую поддержку

Авторизация в системе через REST API

Последние изменения: 2024-03-26

Если в системе включена авторизация, то для начала работы с API необходимо пройти авторизацию в системе.

Реализованы несколько вариантов авторизации:

- BASIC авторизация.
- Авторизация методом GET по адресу `/api/session` с получением token.
- Авторизация методом POST по адресу `/api/session` с получением token.

REST API работает только если при работе с ТСД используется сервер Mobile SMARTS. При прямом подключении ТСД к компьютеру через кабель/кредл или при обмене с учетной системой через папку использовать REST API не получится.

В базе Mobile SMARTS должна быть включена **авторизация**, тогда для выполнения HTTP-запросов необходимо будет каждый раз авторизовываться, есть 2 варианта авторизации:

- BASIC авторизация — в этом случае при каждом HTTP-запросе нужно будет отправлять логин и пароль;
- авторизация с использованием токена — при первом HTTP-запросе передаются логин и пароль, а сервер MS возвращается токен — уникальный идентификатор сессии (access token), который при последующих HTTP-запросах можно будет использовать вместо передачи логина и пароля. Срок действия токена (т.е. сессии) ограничен, поэтому для обновления токена при первом запросе сервер Mobile SMARTS дополнительно возвращает «токен обновления» (refresh token) — он нужен для получения нового токена после истечения срока действия текущего токена.

Обратите внимание. Для защиты передаваемых данных рекомендуется использовать режим с включенной аутентификацией и доступом по https.

Для работы через протокол https необходимо указать это в настройках базы данных, для этого зайдите в менеджер базы, справа кнопки «Добавить» нажмите на выпадающий список и выберите пункт «Настройки», далее в окне «Редактирование настроек» поставьте галочку «Использовать http» и нажмите «ОК».

Mobile SMARTS

Базы данных Mobile SMARTS

Без сервера
 Склад 15, Демо
 1С Драйвер ТСД

Локальный сервер
 Магазин 15, Расширенный с Кировкой
 Магазин 15, Демо
 Склад 15, Демо 2
 Склад 15, Полный с Кировкой
 Магазин 15, Расширенный
 Склад 15, Демо 3
 Склад 15, Расширенный с Ки...
 Склад 15, Омни с Кировкой
 Склад 15, Демо 4
 MobileSMARTS Тестовая конф...

Код базы: rtl15
 Строка подключения:
<http://win-3an96cq4nr:10501/rtl15>
 Рабочий каталог:
 C:\ProgramData\Cleverage\Базы Mobile

Открыть
 Добавить
 Настройка...

Редактирование настроек базы данных Mobile SMARTS

Код базы: rtl15

Наименование: Магазин 15, Расширенный с Кировкой

Папка: C:\ProgramData\Cleverage\Базы Mobile SMARTS\Магазин 15, Расширенный с Кировкой

Комментарий:

Режим работы: Подключение к серверу

Основной режим работы Mobile SMARTS. Все мобильные устройства и сторонние системы работают через сервер.

Использовать https Сертификат...

Аутентификация по пользователю

Порт сервера данных: 9000

Сервер печати используется

Порт сервера печати: 9001

OK Отмена

Для работы в **режиме с включенной аутентификацией** необходимо завести пользователей, задать им логины и пароли. В панели управления Mobile SMARTS узел «Пользователи и группы» содержит данные о пользователях и группах, в которых они состоят.

Конфигурация

- Магазин 15 с Кировкой
 - Типы документов
 - Операции
 - Структура номенклатуры
 - Общие вычисляемые поля
 - Структура таблиц
 - Серверные события и расширения
 - Пользователи и группы**
 - Группы пользователей
 - Пользователи
 - Структура складов
 - Штрихкоды контейнеров
 - Оборудование
 - Этикетки
 - Данные

Инструкция по конфигурации настроек пользователей и групп пользователей, как определить роль пользователей и список типов документов, доступных для обработки пользователям такой группы описана в одноименной [статье на сайте](#).

Пример запроса на авторизацию приложения методом POST

1. Для начала в настройках базы Mobile SMARTS включите авторизацию/аутентификацию по пользователю и нажмите на кнопку «OK»

Редактирование настроек базы данных Mobile SMARTS

Код базы: 15911b0e-830f-433b-8d8c-048988df6cdf

Наименование: MS-5748 Магазин 15

Папка: D:\Work\MS-5748\Clever\CLEVER\01fd935c-ce47-4cb6-a3c7-7ce03b12bd29

Комментарий:

Режим работы: Подключение к серверу

Основной режим работы Mobile SMARTS. Все мобильные устройства и сторонние системы работают через сервер.

Использовать https Сертификат...

Аутентификация по пользователю ▾

Порт сервера данных: 51434

Сервер печати используется

Порт сервера печати: 51435

OK Отмена

2. Пример запроса на авторизацию <http://localhost:51434/api/v1/session?username={username}&password={password}> в ответ приходит Access_token

v1/D
78b7

nam

nam

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize HTML

```
1 {"Access_token": "a856aa43587cf22dabc638005b035873", "Token_type": "bearer", "Expires_in": 2147483647, "Refresh_token":
2 "e4c0f0ce16ae352e474f78e03d295441"}

```

it_ty
it=us

it_ty
it=us

it tv

3. В последующих запросах в заголовке указываем

{Authorization: Bearer {Access_token}}

Если не указать Authorization, в ответ приходит статус ошибки 401 — Unauthorized

The screenshot shows a Postman request configuration for a GET request to `http://localhost:51434/MobileSMARTS/api/v1/Docs['new_07400ce3-5f35-46a6-816a-c87e278b71f5']`. The Headers tab is active, showing a list of headers. The 'Authorization' header is present with the value `Bearer a856aa43587cf22dabc638005b035873`. However, the 'Status' field in the bottom right corner is highlighted in red and displays `Status: 401 Unauthorized`, indicating that the request failed because the Authorization header was not correctly processed or was missing in the context of the test.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.26.8	
<input checked="" type="checkbox"/> Accept	*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input type="checkbox"/> Authorization	Bearer a856aa43587cf22dabc638005b035873	
Key	Value	Description

4. При успешной авторизации приходит ответ и статус 202 — OK

The screenshot shows the same Postman request configuration as above, but now the 'Status' field in the bottom right corner is highlighted in red and displays `Status: 202 OK`, indicating a successful response. The 'Authorization' header is checked and its value is visible in the table below.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.26.8	
<input checked="" type="checkbox"/> Accept	*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> Authorization	Bearer a856aa43587cf22dabc638005b035873	
Key	Value	Description

BASIC авторизация

Данный вид авторизации чаще всего используется браузером для доступа к функциям API.

При использовании данного метода необходимо в заголовке каждого запроса указывать:

```
Authorization: Basic {login}:{password}
```

Допускается base64 при формировании строки {login}:{password}

Авторизация методом GET

Еще один способ авторизации — отправить GET запрос по адресу /api/v1/session, при этом в url запросе указать параметры login и password:

```
https://localhost:9000/api/v1/session?username=${Username}&password=${Password}
```

Ответ сервера:

```
{
  Access_token:"123123123",
  Token_type:"bearer",
  Expires_in:86400,
  Refresh_token:"321321321",
}
```

Авторизация методом POST

Авторизация по логину и паролю происходит путем отправки POST запроса на сервер, в результате которого возвращается access_token и refresh token в формате JSON.

Пример запроса:

```
POST /oauth/token HTTP/1.1
Host: mobilesmares.ru/api/session
Content-Type: application/x-www-form-urlencoded
```

```
grant_type=password&client_id=123&client_secret=user&username=user@domain.ru&password=123456
```

Ответ сервера:

```
{
  Access_token:"123123123",
  Token_type:"bearer",
  Expires_in:86400,
  Refresh_token:"321321321",
}
```

Восстановление после окончания срока действия сессии

Восстановление после окончания срока действия сессии происходит путем отправки refresh_token на сервер, в результате приходит новый access_token и refresh_token

Пример:

```
POST /oauth/token HTTP/1.1
```

Host: mobilesmares.ru/api/session
Content-Type: application/x-www-form-urlencoded

grant_type:refresh_token&client_id=123&client_secret=user&refresh_token=321321321

Ответ:

HTTP/1.1 200 OK
Content-Type: application/json

```
{  
  Access_token:"99999",  
  Token_type:"bearer",  
  Expires_in:86400,  
  Refresh_token:"789789789",  
}
```

Вызов функций с использованием token

Для того чтобы обратиться к функциям (если не используется Basic авторизация), для которых необходима авторизация, необходимо в заголовке Authorization передавать токен:

Authorization: Bearer <token>

Иначе сервер вернет ошибку авторизации 401.

 интеграция, REST API

Не нашли что искали?



Задать вопрос в техническую поддержку

Формат запросов в REST API

Последние изменения: 2024-03-26

Формат передачи входящих и исходящих данных — JSON, в кодировке UTF-8. Входящие параметры должны передаваться в теле POST запроса или в виде query-строки (?field=value) для GET запросов. Нужно учитывать, что GET запросы имеют ограничение на длину URL — 2048 символов.

В случае, если указан неверный адрес — возвращается ошибка 404.

В случае возникновения ошибки авторизации, возвращается ошибка 401.

Запросы строятся в формате OData, подробнее можно посмотреть по адресу

<http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part2-url-conventions.html>

Пагинация

Функции, возвращающие массив значений, поддерживают параметры для порционной загрузки, которые передаются в виде query-строки (?\$top=1&\$skip=10).

Пример:

[https://localhost:9000/api/v1/Docs?\\$top=5&\\$skip=10](https://localhost:9000/api/v1/Docs?$top=5&$skip=10)

При выполнении данного запроса мы получим максимум 5 записей начиная с 11 индекса в списке.

Если в запросе на получение списка указать параметр count=true, в результате вернется дополнительно поле count, которое содержит общее число записей в списке на сервере.

Фильтрация

Для фильтрации данных в запросе можно указывать параметр \$filter, например:

[https://localhost:9000/api/v1/Docs?\\$filter=lastChangeDate gt 2017-10-06T17:41:10Z and documentTypeName eq 'Заказ'](https://localhost:9000/api/v1/Docs?$filter=lastChangeDate gt 2017-10-06T17:41:10Z and documentTypeName eq 'Заказ')

При выполнении данного запроса мы получим список документов, дата изменения которых более указанной в запросе и тип документа — «Заказ».

Выборка только нужных свойств

Также можно указывать список полей, которые необходимо отобразить в результате, например:

[https://localhost:9005/api/v1/Docs?\\$select=id,tables](https://localhost:9005/api/v1/Docs?$select=id,tables)

Результатом выполнения данного запроса будет список документов, состоящих только из 2-х полей — id и tables.

Выборка записи по идентификатору

Для получения нужной записи необходимо составить запрос с указанием идентификатора этой записи, например:

[https://localhost:9005/api/v1/Docs \('165'\)](https://localhost:9005/api/v1/Docs ('165'))

Сортировка

Чтобы отсортировать результат необходимо указать параметр `$orderby`, например:

[https://localhost:9005/api/v1/Docs?\\$orderby=id](https://localhost:9005/api/v1/Docs?$orderby=id)

В результате получим список документов, отсортированный по идентификатору.

Отображение подтаблиц и строк документов

Для документов реализован вывод записей строк документа и таблиц документа.

Пример:

Если в документе с идентификатором «1» существует таблица «ОплатыВозвраты», то получить строки таблицы можно отправив запрос:

[https://localhost:9000/api/v1/Docs \('1'\)/ОплатыВозвраты](https://localhost:9000/api/v1/Docs ('1')/ОплатыВозвраты)

Для получения только строк документа с идентификатором «123» необходимо выполнить запрос

[https://localhost:9000/api/v1/Docs \('123'\)/declaredItems](https://localhost:9000/api/v1/Docs ('123')/declaredItems) это запрос плановой части документа

[http://localhost:9000/api/v1/Docs \('123'\)?\\$expand=currentItems](http://localhost:9000/api/v1/Docs ('123')?$expand=currentItems) запрос фактической части документа

Развертывание сложных свойств

Чтобы отобразить все документы, включая его таблицы со строками, нужно сформировать запрос в таком виде:

[https://localhost:9000/api/v1/Docs?\\$expand=tables \(\\$expand=rows\)](https://localhost:9000/api/v1/Docs?$expand=tables ($expand=rows))

Поиск без учета регистра

[https://localhost:9000/api/v1/Docs?\\$filter=Contains \(tolower \(name\), 'фраза'\)](https://localhost:9000/api/v1/Docs?$filter=Contains (tolower (name), 'фраза'))

Запросы DELETE

Для запросов DELETE требуется параметр `If-Match`, нужно указывать `*`.

Добавление массива данных

В некоторых методах предусмотрено добавление массива

{ «value»:[{первый элемент}, {второй элемент}]}

 интеграция, REST API

Не нашли что искали?



Задать вопрос в техническую поддержку

Примеры запросов в REST API

Последние изменения: 2024-03-26

Получение информации о базе

Для получения информации по текущей базе нужно выполнить запрос:

GET [/api/v1/BaseInfo](#)

Ответ:

```
{
"@odata.context": "http://172.19.0.30:9000/MobileSMARTS/api/v1/$metadata#BaseInfo",
"id": "rtl15",
"name": "Магазин 15, Базовый",
"folder": "C:ProgramDataCleverenceБазы Mobile SMARTSМагазин 15, Базовый",
"appId": "F42C7B5F-405C-4076-AE07-9348F189EE71",
"appName": "Магазин 15, Базовый",
"comment": null,
"allConnectionStrings": [
"https://172.19.0.30:10502/rtl15",
"Доступ к swagger:"
"https://172.19.0.30:10502/rtl15/swagger"
],
"appDescription": {
"appId": "F42C7B5F-405C-4076-AE07-9348F189EE71",
"appName": "Магазин 15, Базовый",
"desktopReqPayment": false,
"configId": "8DC2C8BA-77CE-46AD-93CB-6CFFCA7B96D7",
"serverModeSupported": true,
"onlineCallsSupported": false,
"batchModeSupported": true,
"appLink": "http://cleverence.ru/software/mobile-smarts/rtl15/",
"appSupportLink": "https://www.cleverence.ru/support/category:295/",
"aboutLicLink": "http://cleverence.ru/software/mobile-smarts/rtl15/#spec",
"appVersionsLink": "http://cleverence.ru/software/mobile-smarts/rtl15/#spec",
"comment": "Основная поставка Магазин 15",
"appVersion": "1.1.1.155",
"clientVersion": "3.0.0.100",
"panelVersion": "1.0",
"mainServerVersion": "1.0",
"appServerVersion": "3.0.0.2699",
"minPlatformVersion": "1.0",
"minPlatform35Version": "1.0"
},
"appInstanceSettings": {
"mode": "Server",
"hasServerAuth": false,
"serverSettings": {
```

```
"dataService": {
  "enabled": true,
  "port": 9000,
  "useHttps": false,
  "deviceAuth": false,
  "passwordAuth": false
},
"printService": {
  "enabled": false,
  "port": 9001,
  "useHttps": false,
  "deviceAuth": false,
  "passwordAuth": false
}
}
}
```

Группы пользователей

Работа с массивом

GET [/api/v1/UserGroups](#)

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": "string",
      "name": "string",
      "documentTypeNames": [
        "string"
      ],
      ": true,
      "role": 0,
      "serverSideInventory": true,
      "autorunDocumentTypeName": "string",
      "onStartHandlerName": "string",
      "onFinishHandlerName": "string"
    }
  ]
}
```

Работа с записями по идентификатору

POST [/api/v1/UserGroups \('{key}'\)](#)

PUT [/api/v1/UserGroups \('{key}'\)](#)

PATCH [/api/v1/UserGroups \('{key}'\)](#)

DELETE [/api/v1/UserGroups \('{key}'\)](#)

UserGroup {

id (*string, optional*): Id пользователя ,

name (*string, optional*): Уникальное наименование группы ,

documentTypeNames (*Array[string], optional, read only*),

batchMode (*boolean, optional*): Свойство, позволяющее задать тип обмена данными с сервером. Если true - обмен производится вручную, по запросу пользователем. Иначе, обмен будет производиться периодически, с интервалом, заданным в настройках клиентского приложения ,

role (*integer, optional*): Пользовательская роль = ['0', '1', '2', '3'],

serverSideInventory (*boolean, optional*): Указывает источник номенклатурного справочника. Если true - позиции номенклатуры будут искаться на сервере(необходимо наличие постоянной связи с сервером), иначе - справочник номенклатуры будет загружаться на терминал при операции обмена данными и использоваться локально ,

autorunDocumentTypeName (*string, optional*): Тип документа (виртуальный), который будет открыт автоматически, сразу после запуске программы на ТСД ,

onStartHandlerName (*string, optional*): Имя операции, запускающейся при входе пользователя данной группы ,

onFinishHandlerName (*string, optional*): Имя операции, запускающейся при выходе пользователя данной группы

}

Получение списка пользователей группы

GET [/api/v1/UserGroups \('{key}'\)/users](#)

Пользователи

Работа с массивом

GET [/api/v1/Users](#)

Работа с записями по идентификатору

POST [/api/v1/Users](#)

PUT [/api/v1/Users](#)

PATCH [/api/v1/Users](#)

DELETE [/api/v1/Users](#)

User {

```

id (string, optional): Id пользователя ,
name (string, optional): Имя пользователя ,
password (string, optional): (write only) Пароль пользователя ,
description (string, optional): Описание пользователя ,
barcode (string, optional): (write only) Штрихкод пользователя ,
groupId (string, optional): Имя группы в которой состоит пользователь. Подробнее смотрите
UserGroup ,
groupName (string, optional): Для совместимости со старыми обработками ,
warehouseIds (Array[string], optional, read only)
}

```

Ячейки

GET [/api/v1/Cells](#) — Список ячеек

Работа с записями по идентификатору

POST [/api/v1/Cells](#) — Добавить/редактировать ячейку

DELETE [/api/v1/Cells \('{id}'\)](#) — Удалить ячейку

GET [/api/v1/Cells \('{id}'\)](#) — Получить ячейку по идентификатору

PUT [/api/v1/Cells \('{id}'\)](#) — Добавь/редактировать ячейку по идентификатору

Обновление справочника со сбросом всех записей

POST [/api/v1/Cells/BeginOverwrite](#) — начинает процедуру пакетной выгрузки ячеек на сервер. Все позиции будут накапливаться и не будут доступны до вызова функции `EndOverwrite`.

POST [/api/v1/Cells/EndOverwrite](#) — завершает процедуру пакетной выгрузки номенклатуры. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Старый справочник товаров при этом будет полностью очищен.

Обновление записей

POST [/api/v1/Cells/BeginUpdate](#) — начинает процедуру пакетного обновления ячеек на сервере. Все позиции будут накапливаться и не будут доступны до вызова функции `EndUpdate`.

POST [/api/v1/Cells/EndUpdate](#) — завершает процедуру пакетного обновления ячеек. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Выгруженная номенклатура будет слита с существующим на сервере справочником.

Сброс обновления/ перезаписывания

POST [/api/v1/Cells/ResetUpdate](#) — сбрасывает процедуру пакетного обновления ячеек.

Номенклатура

Получение схемы

Получение списка всех полей номенклатуры:

[https://localhost:9000/api/v1/ProductSchema?\\$expand=allfields](https://localhost:9000/api/v1/ProductSchema?$expand=allfields)

Получение полей номенклатуры:

[https://localhost:9000/api/v1/ProductSchema?\\$expand=fields](https://localhost:9000/api/v1/ProductSchema?$expand=fields)

Получение списка фиксированных полей:

[https://localhost:9000/api/v1/ProductSchema?\\$expand=defaultFields](https://localhost:9000/api/v1/ProductSchema?$expand=defaultFields)

Работа с массивом

GET [/api/v1/Products](#) — список номенклатуры.

Работа с записями по идентификатору

POST [/api/v1/Products](#) — добавить/ редактировать номенклатуру.

DELETE [/api/v1/Products \('{id}'\)](#) — удалить номенклатуру.

GET [/api/v1/Products \('{id}'\)](#) — получить номенклатуру по идентификатору.

PUT [/api/v1/Products \('{id}'\)](#) — добавить/ редактировать номенклатуру по идентификатору.

Обновление справочника со сбросом всех записей

POST [/api/v1/Products/BeginOverwrite](#) — начинает процедуру пакетной выгрузки номенклатуры на сервер. Все позиции будут накапливаться и не будут доступны до вызова функции EndOverwrite.

POST [/api/v1/Products/EndOverwrite](#) — завершает процедуру пакетной выгрузки номенклатуры. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Старый справочник товаров при этом будет полностью очищен.

Обновление записей

POST [/api/v1/Products/BeginUpdate](#) — начинает процедуру пакетного обновления номенклатуры на сервере. Все позиции будут накапливаться и не будут доступны до вызова функции EndUpdate.

POST [/api/v1/Products/EndUpdate](#) — завершает процедуру пакетного обновления номенклатуры. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Выгруженная номенклатура будет слита с существующим на сервере справочником.

Сброс обновления/ перезаписывания

POST [/api/v1/Products/ResetUpdate](#) — сбрасывает процедуру пакетного обновления номенклатуры.

Обновление номенклатуры таблицей значений

POST [/api/v1/Products/BeginUploadProducts](#) — начинает выгрузку позиций номенклатуры.

POST [/api/v1/Products/AddProductToUpload](#) — добавляет в выгрузку товаров товар с упаковкой.

POST [/api/v1/Products/AddProductsToUpload](#) — добавляет в выгрузку товаров товаров с упаковками.

POST [/api/v1/Products/EndUploadProducts](#) — завершает выгрузку товаров.

Таблицы

Работа с массивом

GET [/api/v1/Tables/BiznesProcessy](#) — получить все записи таблицы.

Работа с записями по идентификатору

POST [/api/v1/Tables/BiznesProcessy](#) — редактировать/ добавить запись.

DELETE [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — удалить запись из таблицы.

GET [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — получить запись по идентификатору.

PATCH [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — редактировать запись.

PUT [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — редактировать/добавить запись по известному идентификатору.

Обновление справочника со сбросом всех записей

POST [/api/v1/Tables/BiznesProcessy/BeginOverwrite](#) — начинает процедуру пакетной выгрузки строк таблицы на сервер. Все позиции будут накапливаться и не будут доступны до вызова функции EndOverwrite.

POST [/api/v1/Tables/BiznesProcessy/EndOverwrite](#) — завершает процедуру пакетной выгрузки строк таблицы. Только после вызова этой функции сервер завершит обработку переданных позиций и они попадут в таблицу. Старое содержимое при этом будет полностью очищено.

Обновление записей

POST [/api/v1/Tables/BiznesProcessy/BeginUpdate](#) — начинает процедуру пакетного обновления строк таблицы на сервере. Все передаваемые будут накапливаться и не будут доступны до вызова функции EndUpdate.

POST [/api/v1/Tables/BiznesProcessy/EndUpdate](#) — завершает процедуру пакетного обновления строк таблицы. Только после вызова этой функции сервер завершит обработку переданных позиций и они попадут таблицу. Выгруженные позиции будут слиты с существующей на сервере таблицей.

Сброс обновления\перезаписывания

POST [/api/v1/Tables/BiznesProcessy/ResetUpdate](#) — сбрасывает процедуру пакетного обновления строк таблицы.

Документы

Получение списка типов документов

GET [/api/v1/DocTypes](#) — список типов документов.

GET [/api/v1/DocTypes \('{uni}'\)](#) — получить тип документа по идентификатору.

Получение всех полей типа документа

[https://localhost:9000/api/v1/DocTypes?\\$expand=allfields](https://localhost:9000/api/v1/DocTypes?$expand=allfields)

Работа с массивом

GET [/api/v1/Docs](#) — список документов.

POST [/api/v1/Docs](#) редактировать/ добавить документ.

Работа с записями по идентификатору

DELETE [/api/v1/Docs \('{id}'\)](#) — удалить документ.

GET [/api/v1/Docs \('{id}'\)](#) — получить документ по идентификатору.

PATCH [/api/v1/Docs \('{id}'\)](#) — редактировать документ.

PUT [/api/v1/Docs \('{id}'\)](#) — редактировать/добавить документ по известному идентификатору.

Обновление записей

При любом редактировании документа, он сразу не сохраняется в систему. Сохранение происходит, если вызвать принудительное сохранение ([EndUpdate](#)), либо через 30 сек от последнего изменения.

POST [/api/v1/Docs \('{id}'\)/EndUpdate](#) — принудительно сохраняет документ, когда все строки уже загружены (не дожидаясь сохранения через 30 сек, как указано выше).

Получение строк документа

GET [/api/v1/Docs \('{key}'\)/declaredItems](#) — получить строк документа.

POST [/api/v1/Docs \({key}\)/declaredItems](#) — редактировать/добавить строку документа.

[https://localhost:5001/api/v1/Docs \('{id}'\) ?\\$expand=currentItems](https://localhost:5001/api/v1/Docs ('{id}') ?$expand=currentItems) — получить CurentlItems строк документа по его ID.

Блокировка документа

POST [/api/v1/Docs \('{id}'\)/Block](#) — блокирует документ для совместной работы.

Тело запроса:

```
{"timeout»:1000}
```

, где timeout — время блокировки документа

POST [/api/v1/Docs \('{id}'\)/Unblock](#) — разблокирует документ для совместной работы.

Склады

Работа с массивом

GET [/api/v1/Warehouses](#) — получить список складов.

Работа с записями по идентификатору

POST [/api/v1/Warehouses](#) — добавление/ редактирование склада.

DELETE [/api/v1/Warehouses \('{id}'\)](#) — удаление склада.

GET [/api/v1/Warehouses \('{id}'\)](#) — получить конкретный склад.

PATCH [/api/v1/Warehouses \('{id}'\)](#) — редактирование существующего склада.

PUT [/api/v1/Warehouses \('{id}'\)](#) — добавление/ редактирование склада по существующему идентификатору.

Warehouse {

```

storaged (string, optional): Идентификатор склада, для хранения (не меняется) ,
id (string, optional): Уникальный идентификатор склада ,
name (string, optional): Наименование склада ,
cells (Array[Cell], optional, read only): Коллекция ячеек склада
}
Cell {
barcode (string, optional): Штрихкод ячейки. Может быть шаблонированным. Подробнее про
применение шаблонов для ячеек, смотрите Руководство разработчика ,
id (string, optional): Id ячейки, искусственный ключ ,
name (string, optional): Наименование ячейки ,
description (string, optional): Описание ячейки
}

```

Получение информации о подключенных устройствах

GET [/api/v1/Devices](#) — список устройств.

DeviceInfo {

```

appInstanceId (string, optional),
deviceId (string, optional): Уникальный идентификатор терминала. Заполняется самим
устройством при регистрации его в системе ,
batteryStatus (string, optional): Уровень заряда батарей ,
lastInfoTime (string, optional): Последнее время получения информации о терминале ,
userId (string, optional): Id пользователя, работавший с устройством в момент последнего
получения информации о нем ,
warehouseId (string, optional): Идентификатор склада, на котором работает пользователь ,
documentId (string, optional): Идентификатор документа, с которым работает пользователь ,
cellId (string, optional): Идентификатор последней ячейки, с которой работал пользователь ,
deviceName (string, optional): Пользовательское имя устройства ,
documentTypeName (string, optional),
serverHostedDocument (boolean, optional),
deviceIp (string, optional),
userGroupId (string, optional)

```

}

Настройки

GET [/api/v1/CustomSettings](#) — список настроек

Результат:

```
{
"@odata.context": "http://localhost:9000/MobileSMARTS/api/v1/$metadata#CustomSettings";
"value": [
{
"name": "testProp",
"value": "testValue"
}
]
}
```

POST /api/v1/CustomSettings — Добавить/отредактировать настройку

Тело запроса:

```
{
"name": "testProp",
"value": "testValue"
}
```

Также можно добавлять массив настроек:

```
{
"value":[
{
"name": "testProp",
"value": "testValue"
},
{
"name": "testProp1",
"value": "testValue1"
}
]
}
```

GET /api/v1/CustomSettings ('{name}') — получить настройку

Тело ответа:

```
{
"@odata.context":
"http://localhost:9000/MobileSMARTS/api/v1/$metadata#CustomSettings/$entity";
"name": "testProp",
"value": "testValue"
}
```

PUT /api/v1/CustomSettings ('{name}') — Добавить/отредактировать настройку по известному идентификатор.

Тело запроса:

```
{
  "name": "testProp",
  "value": "testValue"
}
```

DELETE [/api/v1/CustomSettings \('{name}'\)](#) — удалить настройку

Сообщения

Действия с сообщениями:

GET [/api/v1/Messages](#) — список сообщений.

POST [/api/v1/Messages](#) — добавить/отредактировать сообщение.

GET [/api/v1/Messages \('{id}'\)](#) — получить сообщение по идентификатору.

MessageInfo {

```
  applInstanceid (string, optional),
  deviceId (string, optional): Уникальный идентификатор терминала. Заполняется самим
  устройством при регистрации его в системе ,
  batteryStatus (string, optional): Уровень заряда батарей ,
  lastInfoTime (string, optional): Последнее время получения информации о терминале ,
  userId (string, optional): Id пользователя, работавший с устройством в момент последнего
  получения информации о нем ,
  warehouseId (string, optional): Идентификатор склада, на котором работает пользователь ,
  documentId (string, optional): Идентификатор документа, с которым работает пользователь ,
  cellId (string, optional): Идентификатор последней ячейки, с которой работал пользователь ,
  deviceName (string, optional): Пользовательское имя устройства ,
  documentTypeName (string, optional),
  serverHostedDocument (boolean, optional),
  deviceIp (string, optional),
  userGroupId (string, optional)
}
```

 REST API

Не нашли что искали?



Задать вопрос в техническую поддержку

Особенности загрузки данных через REST API

Последние изменения: 2024-03-26

Отложенное внесение изменений

Обновление данных с помощью REST API всегда имеет отложенное срабатывание. Это значит, что большинство справочников (номенклатура, таблицы) и документы в базе данных на сервере обновляются не мгновенно, а с задержкой (30 секунд).

Для чего это делается:

- Для предотвращения мгновенной отправки измененного справочника на мобильное устройство, так как требуется время, чтобы успеть обновить его версию на мобильном устройстве.
- Для того, чтобы избежать генерации лишнего промежуточного трафика из-за порционности POST-запросов.
- Относительно справочника номенклатуры это решает проблему лишней нагрузки на сервер по формированию справочника и его индексов, так как данный справочник имеет специфический внутренний формат (используемый для ТСД на Windows CE).

Обновление документа

При загрузке большого числа строк в документе включается режим обновления (BeginUpdate), который завершится автоматически через 30 секунд и документ сохранится на диск. В режиме обновления документ не сохраняется при добавлении каждой строки. Если после загрузки последней строки требуется сразу записать документ на диск, необходимо выполнить запрос EndUpdate. Такая схема предотвращает выполнения события на сервере до завершения загрузки всех строк документа.

Пример:

```
/api/v1/Docs/ChekKorrekcii ('{id}')/EndUpdate
```

Обновление таблицы и справочника номенклатуры

При загрузке большого числа строк в какую-либо таблицу системы необходимо включить режим обновления таблицы. Существуют 2 режима обновления — добавление новых записей и полная перезапись таблицы. При перезаписи старые данные заменяются новыми, т. е. таблица очистится и в нее добавятся новые записи.

Для перезаписи необходимо вызвать запросы:

```
/api/v1/Tables/Kontragenty/BeginOverwrite — включить режим перезаписи для таблицы Kontragenty
```

{Далее необходимо отправить все новые записи}

```
/api/v1/Tables/Kontragenty/EndOverwrite — завершить режим перезаписи и заменить предыдущую таблицу Kontragenty
```

Для обновления записей таблицы необходимо вызвать запросы:

```
/api/v1/Tables/Kontragenty/BeginUpdate — начать обновление таблицы
```

{Далее необходимо отправить все новые записи}

`/api/v1/Tables/Kontragenty/EndUpdate`

Для справочника номенклатуры запросы формируются следующим образом:

`/api/v1/Products/BeginOverwrite`

`/api/v1/Products/EndOverwrite`

`/api/v1/Products/BeginUpdate`

`/api/v1/Products/EndUpdate`

Пакетная загрузка номенклатуры на сервер REST API

Пакетную загрузку номенклатуры можно реализовать следующим способом:

POST `/api/v1/Products/ResetUpdate` — сбрасываем процедуру пакетного обновления номенклатуры на сервере

Тело сообщения:

```
{}
```

POST `/api/v1/Products/BeginUpdate` — начинаем процедуру пакетного обновления номенклатуры на сервере

Тело сообщения:

```
{}
```

POST `/api/v1/Products` — Добавляем номенклатуру на сервер двумя пакетами

Пакет 1. Тело сообщения:

```
{
  "value": [
    {
      "id": "1f738be6-0433-4a85-866f-3479f7ec1eda",
      "name": "Товар1",
      "barcode": "1001",
      "basePackingId": "шт",
      "packings": [
        {
          "id": "шт"
        }
      ]
    },
    {
      "id": "2f738be6-0433-4a85-866f-3479f7ec1eda ",
      "name": "Товар2",
      "barcode": "1002",
      "basePackingId": "шт",
      "packings": [
        {
          "id": "шт"
        }
      ]
    }
  ]
}
```

Пакет 2. Тело сообщения:

```
{
  "value": [
    {
      "id": "3f738be6-0433-4a85-866f-3479f7ec1eda ",
      "name": "Товар3",
      "barcode": "1003",
      "basePackingId": "шт",
      "packings": [
        {
          "id": "шт"
        }
      ]
    },
    {
      "id": "4f738be6-0433-4a85-866f-3479f7ec1eda ",
      "name": "Товар4",
      "barcode": "1004",
      "basePackingId": "шт",
      "packings": [
        {
          "id": "шт"
        }
      ]
    }
  ]
}
```

Для тестовых сообщений сформированы сообщения в формате json по базовым полям в количестве 2 единиц в одном пакете

POST /api/v1/Products/EndUpdate — завершаем процедуру пакетного обновления номенклатуры на сервере

Тело сообщения:

```
{}
```

 интеграция, REST API

Не нашли что искали?



Задать вопрос в техническую поддержку

Расширение API через коннектор

Последние изменения: 2024-03-26

Есть возможность расширить API, создав свой коннектор в серверной части платформы.

Более подробно о создании коннектора можно посмотреть [здесь](#).

Для этого при создании коннектор нужно унаследовать и от интерфейса `IApiExtenderPlugin`, например:

```
public class ApiExtenderConnector : IConnector, IApiExtenderPlugin
{
    public IList<IApiGroup> ApiGroup { get; private set; }
}
```

Свойство `ApiGroup` возвращает массив классов, унаследованных от `IApiGroup`, которые будут обрабатывать все запросы расширения из внешней системы.

```
public interface IApiGroup
{
    string Id { get; }
    object CreateNewEntity();
    string GetEntityIdFieldName();
    IEnumerable<object> GetList();
    object Get(string id);
    object Post(object input);
    object Put(string id, object input);
    void Delete(string id);
}
```

`Id` — идентификатор, который должен быть уникальным в системе. Он может содержать как буквы, так и цифры. Для обращения к этому расширению в url строку нужно будет вставить этот идентификатор, например:

Создаем класс коннектора:

```
public class ApiExtenderConnector : IConnector, IApiExtenderPlugin
{
    public ApiExtenderConnector() {
        ApiGroup = new List<IApiGroup>();
        ApiGroup.Add(new ApiTestGroup());
    }
    public bool IsSelfTimeoutBehavior { get { return false; } }
    private string id;
    [System.ComponentModel.Description("Идентификатор.")]
    public string Id
    {
```

```
get { return this.id; }
set { this.id = value; }
}
bool initialized = false;
public bool Initialized
{
get
{
return this.initialized;
}
}
private int timeout = 0;
public int Timeout
{
get { return this.timeout; }
set { this.timeout = value; }
}
private bool enabled = true;
public bool Enabled
{
get
{
return this.enabled;
}
set
{
this.enabled = value;
}
}
private TimeoutBehavior timeoutBehavior = TimeoutBehavior.ThrowException;
public TimeoutBehavior TimeoutBehavior
{
get { return this.timeoutBehavior; }
set { this.timeoutBehavior = value; }
}
public bool IsSupportDeviceInfoInArgs
{get { return false; }
}
[Cleverence.DataCollection.Xml.XmlSerializable(Cleverence.DataCollection.Xml.XmlSerializationType.
None)]
public IList<IApiGroup> ApiGroup { get; private set;}
public void CheckLicenseLimitations(List<LicenseExternalSystem> supportedSystems)
{
}
public void Initialize()
{
}
public object InvokeMethod(string methodName, object[] args)
{
return null;
}
}
```

Далее необходимо откомпилировать и положить готовую библиотеку в папку сервера. В результате после удачного запуска сервера проверяем ответ сервера по пути

<https://localhost:<порт базы>/api/v1/>

В ответе должна содержаться группа

```
{"name»: "Plugins/ApiTestGroup», "kind»: "EntitySet», "url»: "Plugins/ApiTestGroup"}
```

Далее при переходе по пути

<https://localhost:<порт базы>/api/v1/Plugins/ApiTestGroup>

получаем список наших сущностей (результат выполнения функции GetList ())

```
{
"@odata.context": "https://localhost:9005/MobileSMARTS/api/v1/$metadata#Plugins&quot;,
"value": [
{"id": "1", "name": "1111", "valueInt": 0, "valueBool": false},
{"id": "2", "name": "2223", "valueInt": 0, "valueBool": false},
{"id": "3", "name": "3333", "valueInt": 0, "valueBool": false}]
}
```

Далее POST запрос можем проверить в swagger. Для этого заходим

<https://localhost:<порт базы>/swagger>

Находим там нашу группу Plugins/ApiTestGroup, выбираем POST, вставляем в параметр body:

```
{"id»: "4», "name»: "new4"}
```

Нажимаем кнопку «Попробовать!»

Plugins/ApiTestGroup

Показать/Скрыть | Операции кратко | Операции подробно

GET /api/v1/Plugins/ApiTestGroup [Получить все записи](#)POST /api/v1/Plugins/ApiTestGroup [Отправить значения в функцию](#)

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
objectWrapper	<pre>{"id": "4", "name": "new4"}</pre>		body	Описание Пример <pre>{ "id": "string" }</pre>

Content Type параметра: [Попробовать!](#)[Спрятать ответ](#)

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"id": "4", "name": "new4"}' 'https://
```

URL запроса

```
https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup
```

Тело ответа

```
{
  "@odata.context": "https://localhost:9005/MobileSMARTS/api/v1/$metadata#Plugins/$entity",
  "id": "4",
  "name": "new4",
  "valueInt": 0,
  "valueBool": false
}
```

HTTP код ответа

201

Далее заходим в блок GET и видим что наша новая сущность сохранилась:

[Попробовать!](#) [Спрятать ответ](#)

Curl

```
curl -X GET --header 'Accept: application/json' 'https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup'
```

URL запроса

```
https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup
```

Тело ответа

```
{
  "id": "2",
  "name": "2223",
  "valueInt": 0,
  "valueBool": false
},
{
  "id": "3",
  "name": "3333",
  "valueInt": 0,
  "valueBool": false
},
{
  "id": "4",
  "name": "new4",
  "valueInt": 0,
  "valueBool": false
}
]
```

Для удаления заходим в блок DELETE, вводим наш новый идентификатор 4 и выполняем.

DELETE /api/v1/Plugins/ApiTestGroup('{id}') Запрос на удаление значений

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
id	4	key: id	path	string
If-Match	*	If-Match header	header	string

Что может прийти в ответ

HTTP код	Причина	Структура ответа	Заголовки
204	NoContent		

[Попробовать!](#)

Проверяем в блоке GET и видим, что новая сущность исчезла.

[Попробовать!](#) [Скрыть ответ](#)

Curl

```
curl -X GET --header 'Accept: application/json' 'https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup'
```

URL запроса

```
https://localhost:9005/MobileSMARTS/api/v1/Plugins/ApiTestGroup
```

Тело ответа

```
{
  {
    "id": "2",
    "name": "2223",
    "valueInt": 0,
    "valueBool": false
  },
  {
    "id": "3",
    "name": "3333",
    "valueInt": 0,
    "valueBool": false
  }
}
```

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

Блокировка документа через REST API

Последние изменения: 2024-03-26

Для редактирования коллективного документа через API нужно его заблокировать.

Блокировка методом block

Для этого нужно использовать метод block с идентификатором документа

[http://localhost:12471/MobileSMARTS/api/v1/Docs \('6007dccb-43c9-40f9-86ff-f66e413b0e77'\)/Block](http://localhost:12471/MobileSMARTS/api/v1/Docs ('6007dccb-43c9-40f9-86ff-f66e413b0e77')/Block)

POST /api/v1/Docs('{id}')/Block Call operation

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных				
id	<input type="text" value="6007dccb-43c9-40f9-86ff-f66e413b0e77"/>	key: id	path	string				
parameters	<pre>{ "timeout": 9000 }</pre>	Block action parameters	body	<table border="1"> <thead> <tr> <th>Описание</th> <th>Пример</th> </tr> </thead> <tbody> <tr> <td></td> <td><pre>{ "timeout": 0 }</pre></td> </tr> </tbody> </table>	Описание	Пример		<pre>{ "timeout": 0 }</pre>
Описание	Пример							
	<pre>{ "timeout": 0 }</pre>							

В параметрах нужно обязательно указывать таймаут (через какое время документ автоматически разблокируется, если его не разблокировали вручную). Диапазон от 500 до 10000 мс.

Разблокировка методом unblock

Для ручной разблокировки нужно использовать метод unblock с указанием идентификатора документа. После ручной разблокировки — автоматическая разблокировка не произойдет.

[http://localhost:12471/MobileSMARTS/api/v1/Docs \('6007dccb-43c9-40f9-86ff-f66e413b0e77'\)/Unblock](http://localhost:12471/MobileSMARTS/api/v1/Docs ('6007dccb-43c9-40f9-86ff-f66e413b0e77')/Unblock)

POST /api/v1/Docs('{id}')/UnBlock Разблокирует документ для совместной ра

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
id	<input type="text" value="6007dccb-43c9-40f9-86ff-f66e413b0e77"/>	key: id	path	string
UnBlockActionParameters	<input type="text"/>	UnBlock action parameters	body	Inline Model {}

Добавления новой строки методом post

Для добавления новой отдельной строки документа можно использовать метод post в DeclaredItems.

Пример

Method POST Request URL [http://localhost:12471/MobileSMARTS/api/v1/Docs\('6007dccb-43c9-40f9-86ff-f66e413b0e77'\)/DeclaredItems](http://localhost:12471/MobileSMARTS/api/v1/Docs('6007dccb-43c9-40f9-86ff-f66e413b0e77')/DeclaredItems) SEND

Parameters ^

Headers Body Variables

Body content type application/json Editor view Raw input

FORMAT JSON MINIFY JSON

```
{
  "uid": "047df020-9b57-4814-80b0-f35104d8f359",
  "createdBy": "Unknown",
  "productId": "697b6490-3453-4a86-b3a4-d05172213c81",
  "declaredQuantity": 3,
  "currentQuantity": 0,
  "currentQuantityWithBinding": 0,
  "firstStorageId": null,
  "secondStorageId": null,
  "firstStorageBarcode": null,
  "secondStorageBarcode": null,
  "firstCellId": null,
  "secondCellId": null,
  "packingId": "266e3c86-0aa2-4291-8b9e-bde72f2b6809",
  "sscc": null,
```

Редактирование методом patch

Для редактирования существующей строки можно пользоваться методом patch.

Пример

[http://localhost:12471/MobileSMARTS/api/v1/Docs\('6007dccb-43c9-40f9-86ff-f66e413b0e77'\)/DeclaredItems](http://localhost:12471/MobileSMARTS/api/v1/Docs('6007dccb-43c9-40f9-86ff-f66e413b0e77')/DeclaredItems)

Method Request URL
PATCH [http://localhost:12471/MobileSMARTS/api/v1/Docs\('6007dccb-43c9-40f9-86ff-f66e413b0e77'\)/DeclaredItems](http://localhost:12471/MobileSMARTS/api/v1/Docs('6007dccb-43c9-40f9-86ff-f66e413b0e77')/DeclaredItems) SEND

Parameters ^

Headers Body Variables

Body content type Editor view
application/json Raw input

FORMAT JSON MINIFY JSON

```
{
  "uid": "047df020-9b57-4814-80b0-f35104d8f359",
  "packingId": "266e3c86-0aa2-4291-8b9e-bde72f2b6809"
}
```

При патче можно указывать только те поля, которые необходимо обновить в нужной строке. В теле обязательно нужно указывать `uid` нужной строки!

Удаление строки методом Delete

Для удаления нужной строки у документа можно использовать метод Delete.

Пример

DELETE [http://localhost:12471/MobileSMARTS/api/v1/Docs \('6007dccb-43c9-40f9-86ff-f66e413b0e77'\)/DeclaredItems \('047df020-9b57-4814-80b0-f35104d8f359'\)](http://localhost:12471/MobileSMARTS/api/v1/Docs('6007dccb-43c9-40f9-86ff-f66e413b0e77')/DeclaredItems('047df020-9b57-4814-80b0-f35104d8f359')),

тут в урле необходимо указать как идентификатор документа, так и `uid` строки из `declaredItems`.

Не нашли что искали?



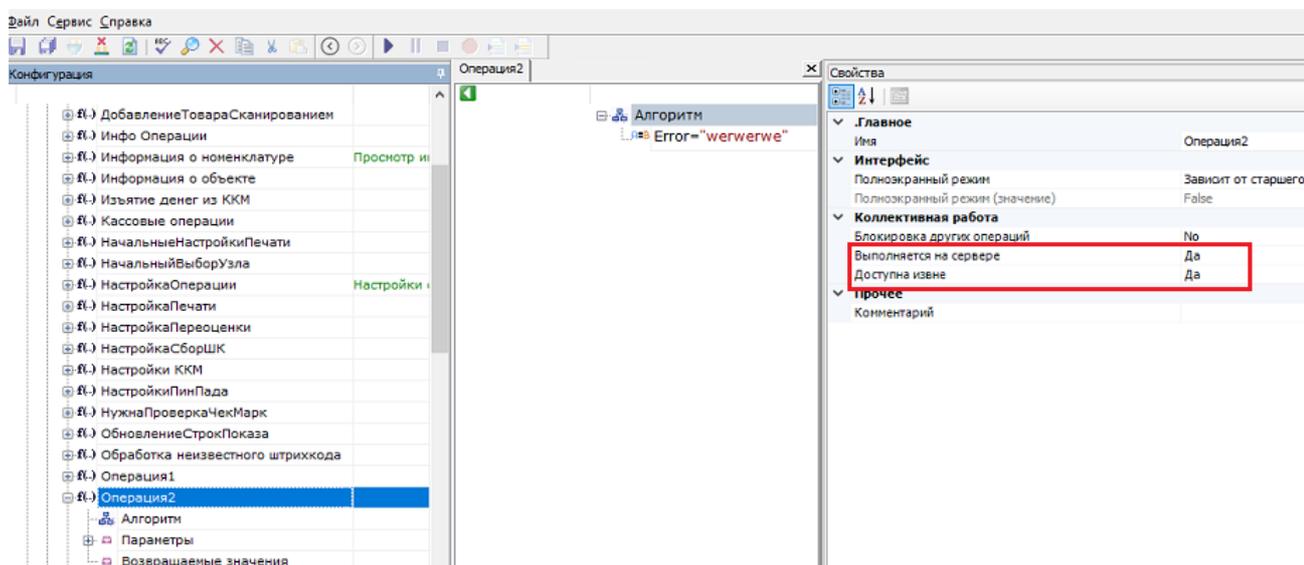
Задать вопрос в техническую поддержку

Вызов серверных операций через REST API

Последние изменения: 2024-03-26

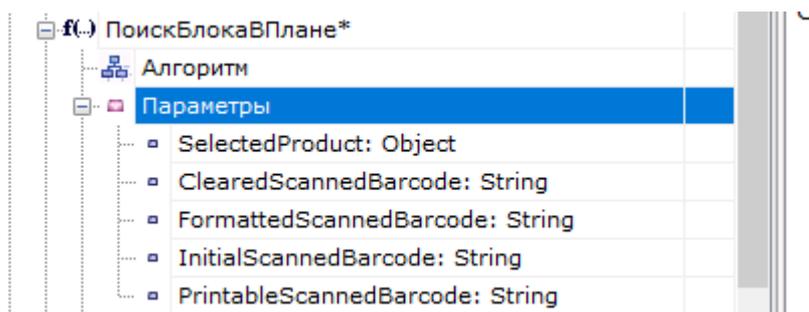
Начиная с версии 3.2.46.20453 у платформы Mobile SMARTS появилась возможность вызывать **серверные операции** с помощью REST API. Для этого необходимо:

1. В панели управления **Mobile SMARTS** для требуемой операции установить значения «Да» у параметров «Выполняется на сервере» и «Доступна извне».



2. Выполнить запрос [http://localhost:9000/MobileSMARTS/api/v1/Operations\('{OperationName}'\)](http://localhost:9000/MobileSMARTS/api/v1/Operations('{OperationName}')), где {OperationName} - имя требуемой операции.

3. Входные параметры, необходимые для выполнения операции указываются тут:



4. Если в операцию необходимо передать параметры, то они перечисляются в теле запроса в формате json:

```
{
  "a": "параметр1",
  "b": "параметр2"
}
```

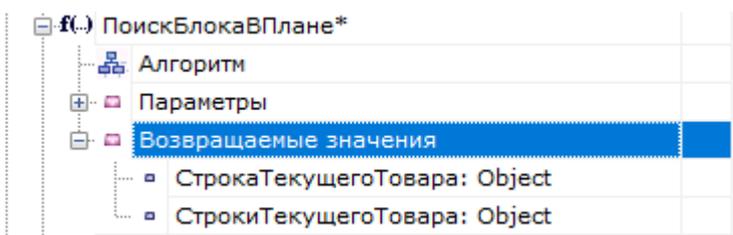
5. Если в параметрах необходимо передать сложный тип, то при описании параметра в json необходимо добавить поле "@odata.type", в котором указывается тип передаваемого параметра. Например:

```
{
  "field0": "11",
  "field1": {
    "@odata.type": "#Cleverence.Warehouse.Document",
    "id": "7c342252-de63-42fe-9742-b47b0a40a7ee"
  }
}
```

6. В ответе сервер вернет результат выполнения операции в виде json, в котором будут перечислены выходные параметры после выполнения операции. Например:

```
{
  vvodCeny: true,
  sklad: true
}
```

7. Выходные параметры указываются при настройке операции тут:



8. Выполнение серверной операции можно протестировать в Swagger:

Operations Показать/Скрыть | Операции кратко | Операции подробно

POST /api/v1/Operations('{operationName}') Вызов серверной операции

Пример ответа (Статус 200)
ОК

Описание Пример

```
{}
```

Content Type ответа:

Параметры

Параметр	Значение	Описание	Тип параметра	Тип данных
operationName	<input type="text" value="Операция1"/>	key: operationName	path	string
InvokeOperation	<pre>{ vvodCeny: true, sklad: true }</pre>	The entity to put	body	Описание Пример <pre>{}</pre>

Content Type параметра:

Не нашли что искали?



Задать вопрос в техническую поддержку

Принцип онлайн-обмена документами и справочниками между 1С и ТСД

Последние изменения: 2024-03-26

В работе Mobile SMARTS, помимо ручной выгрузки/загрузки документов и справочников, предусмотрен [онлайн-режим](#), который позволяет брать необходимые данные из 1С при работе на ТСД.

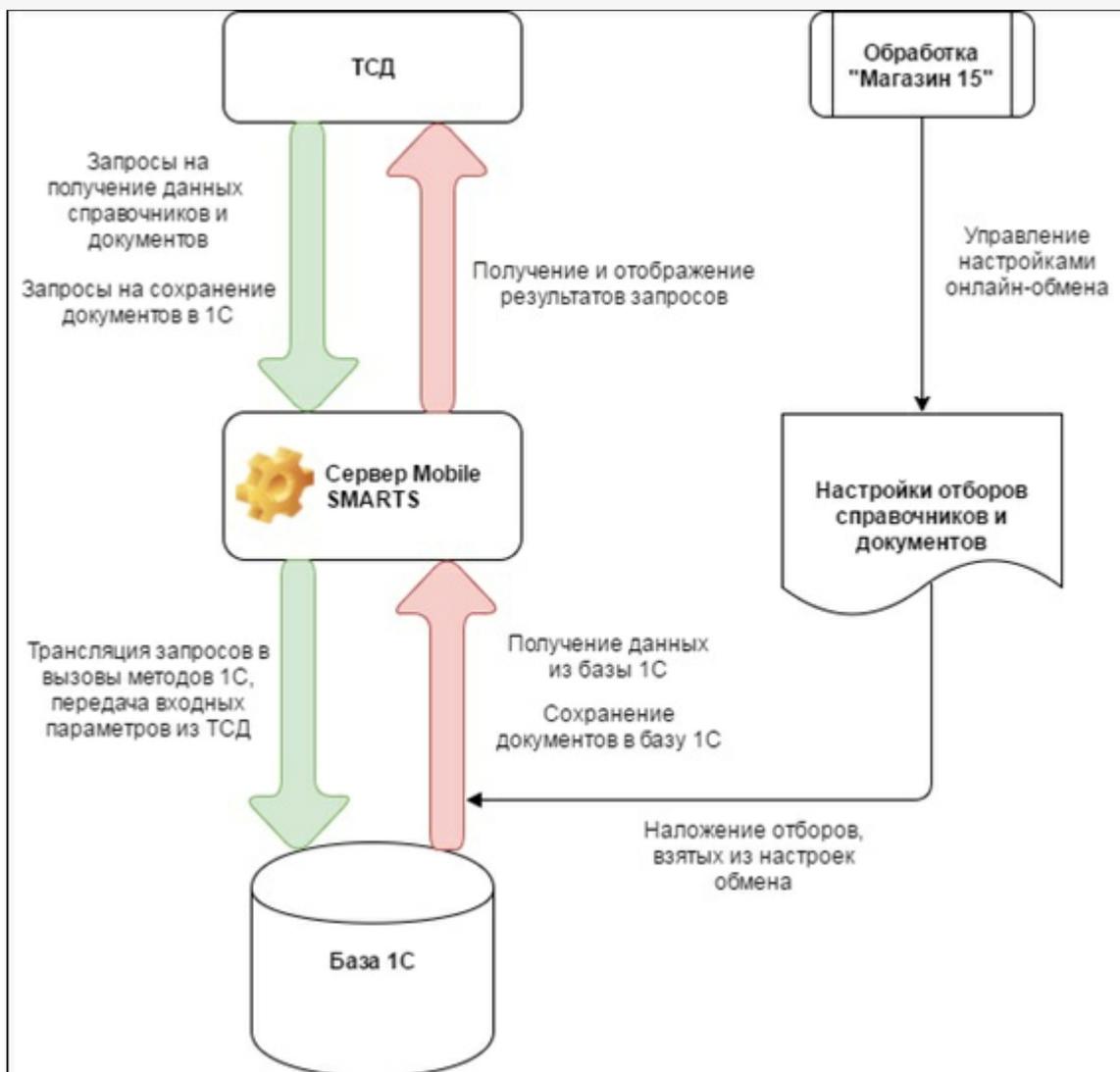
В данной статье подробно рассматриваются принципы работы терминала сбора данных (далее — ТСД) в режиме онлайн-обмена с учетной системой на базе 1С. В качестве примера используется типовая конфигурация «1С:Розница 2.2». После прочтения данной статьи пользователь сможет самостоятельно настроить работу онлайн-обмена в своей базе.

Теоретическая часть и подготовительные действия

1. Онлайн-обмен предназначен для получения данных справочников и документов на терминале непосредственно из базы 1С, БЕЗ ручной загрузки/ выгрузки данных. Таким образом, пользователю, работающему с ТСД, нет необходимости после этапа сканирования документов возвращаться к рабочему месту, запускать 1С и вручную загружать/выгружать данные с помощью внешней обработки.
2. Сервер Mobile SMARTS обменивается данными с базой 1С через «V8.ComConnector». В этом случае подключение осуществляется напрямую к серверу 1С, даже если мы подключаемся к локальной файловой базе.

Для старых продуктов на платформе Mobile SMARTS (ПРОФ драйвер) можно использовать еще один способ обмена — с помощью объекта «V8.Application» (подключение осуществляется через запуск толстого клиента 1С, но в «невидимом» режиме, без показа окна приложения). Данный метод не подходит для новых продуктов («Магазин 15», «Склад 15» и др.)

Подробности представлены на структурной диаграмме:



3. В панели управления Mobile SMARTS это можно увидеть здесь:

The screenshot shows the configuration for the 1C connector. The 'Прочее' (Other) section is highlighted with a red box, containing the following settings:

Обработка драйвера	C:/ProgramData/Cleverence/Базы MobileSMARTS/2a9
Пароль	
Пароль администратора кластера	
Пользователь	Cleverence
Тип подключения	Менеджер COM-соединений (V8x.COMConnector)

4. Включение/выключение режима «Онлайн» доступно в окне «Настройки подключения».

Начало работы

Склад 15, Омни, v.1.3.2.29

База данных «Склад 15, Омни» [подробнее](#)

Имя базы: Управление торговлей, редакция 11 v.
 Версия 1С: 8.3.19.1229
 Папка/Адрес: Srvr="VM-WS2019ST-1C";Ref="ut114demo";

 Онлайн

Открыть базу 1С...

Настройки подключения...

Папка с обработками 1С:

<C:\ProgramData\Cleverence\Базы Mobile SMARTS\aff6021c-120f-468d-b32a-d25b8aa3ff84\Обработки 1С>

Если данные до работы с ТСД были выгружены на сервер в ручном режиме, то ТСД будет использовать их, а не обращаться к базе 1С посредством онлайн-обмена. Недостатком данного метода будет то, что пользователь ТСД не будет видеть на терминале актуальные данные из 1С, для этого нужно будет с определенной периодичностью обмениваться данными с 1С в ручном режиме и, таким образом, актуализировать данные об остатках, ценах, товарах и т. д.

- Для работы ТСД с 1С в режиме «Онлайн» используется механизм обработчиков различных событий, возникающих во время работы с ТСД (поиск элемента справочника по наименованию/штрихкоду, получение списка документов, получение товарного состава документа и т. д.), при этом все обработчики событий должны располагаться в модуле объекта обработки и быть экспортируемыми.
- Список событий и сопоставленных им обработчиков доступен на вкладке «События сервера»:

Событие	Обработчик
Misc	
Добавлять объект сессии в вызов событ	Да
События документов	
Документ возвращен без изменений	
ДокументДобавлен	
ДокументЗавершен	OneC_Connector:ДокументЗавершен
ДокументИзменен	
ДокументПолученНаОбработку	
ДокументУдален	
ПолучитьДокумент	OneC_Connector:ПолучитьДокумент
ПолучитьСписокДокументов	OneC_Connector:ПолучитьСписокДокументов
События номенклатуры	
НайтиНоменклатуруПоЧастиНаименова	OneC_Connector:НайтиНоменклатуруПоЧастиНаименования
ПолучитьСписокНоменклатуры	OneC_Connector:ПолучитьСписокНоменклатуры
ТоварНеНайден	OneC_Connector:ПолучитьТовар
ТоварыНеНайден	OneC_Connector:ПолучитьТовары
События таблиц	
Обработать запрос	OneC_Connector:ОбработатьЗапрос

Как видим на скриншоте, далеко не все обработчики на данный момент используются, это значит, что, в дальнейшем, подразумевается увеличение функциональности онлайн-обмена.

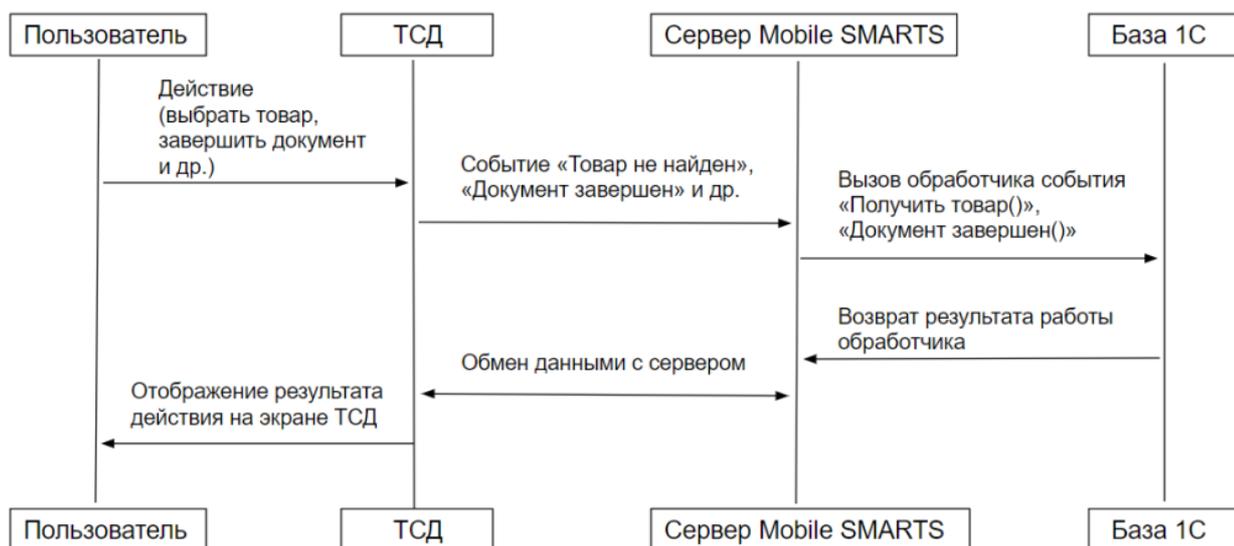
Для примера, список нескольких обработчиков в модуле объекта 1С:

```

// +++
// Режим 0 - получение товара с упаковка по по ID
// Режим 1 - получение товара по штрихкоду/Коду/Артикулу
// Режим 2 - получение ID Товара + ID Упаковки (наименование)
// Режим 3 - получение упаковки по ID упаковки и ID товара
// Режим 4 - получение получение списка товаров по ID и по ID упаковок
// Режим 5 - получение получение списка товаров по родителям
// Режим 6 - получение получение списка товаров по наименованию
// +++
Функция ПолучитьТовар(Код, КодУпаковки, UserId, Режим, mXmlDoc=Неопределено) Экспорт
Функция ОтобразитьДокументы(НастройкиКомпоновки, ТипДокумента1С)
Функция ПолучитьБизнесПроцессы(ТипДокументаMS, ПараметрыВызовSMARTC = Неопределено)
Функция ПолучитьНастройкиБазыНаСервере (БазаMS)
Функция ПолучитьНастройкиДокументовНаСервере ()
Функция ПолучитьСписокДокументов(UserId, ТипДокумента=Неопределено, xmlСессии = Неопределено) Экспорт
Функция ПолучитьДокументПоШтрихкоду(UserId, КодДокумента, ТипДокумента, xmlСессии=Неопределено) Экспорт
Функция ПолучитьНастройкиОтборовПоУмолчанию()
Функция ПолучитьДокументПоКоду(UserId, КодДокумента, ОперацияТСД, xmlСессии=Неопределено) Экспорт
Функция ПолучитьДокумент(UserId, КодДокумента, ОперацияТСД, Режим, xmlСессии=Неопределено) Экспорт
Функция ПодготовитьСтруктуруДанных1(СтруктураНастроек, СхемаКомпоновкиДанных, Компонент) Экспорт

```

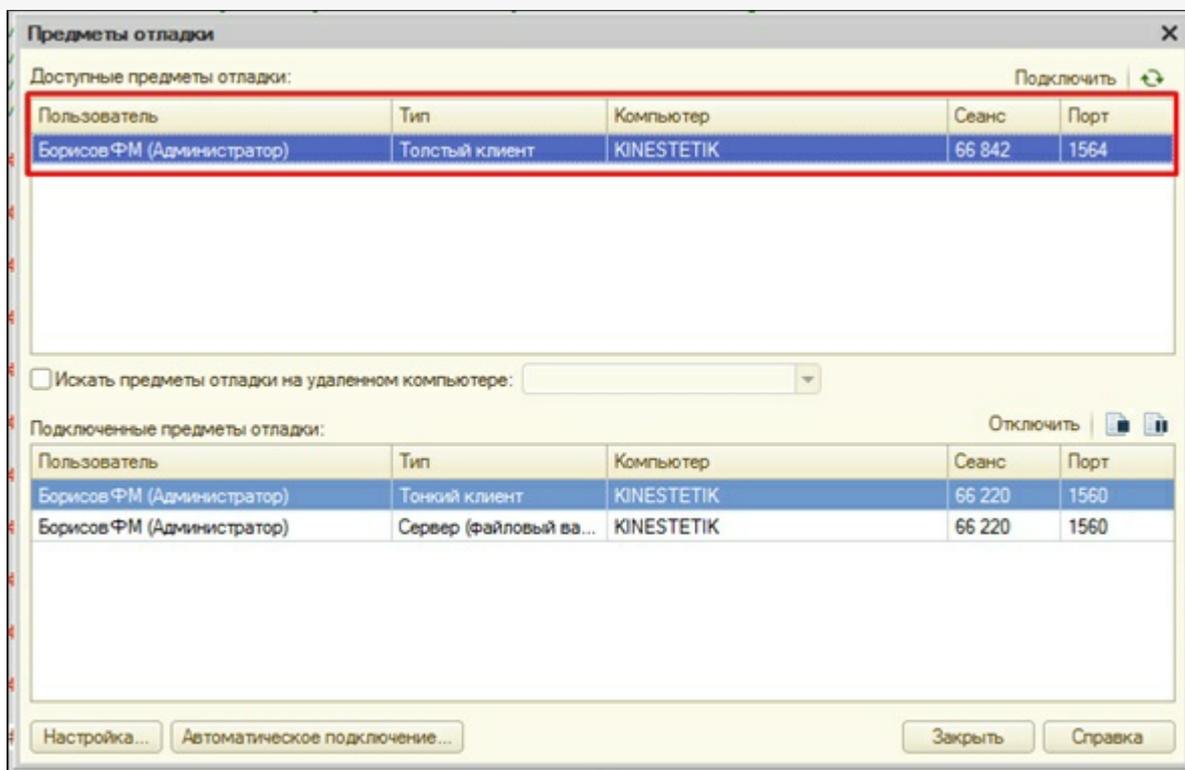
7. Для наглядности происходящих процессов представлена динамическая диаграмма:



Более подробную информацию о том, для чего нужен каждый из обработчиков и при каких событиях на ТСД они вызываются, можно узнать у сотрудников компании «Клеверенс»: Сергея Шаширова и Михаила Пузырёва.

Демонстрация работы и отладки онлайн-обмена

1. После того как режим «Онлайн» запущен, на компьютере запускается COM-соединение, посредством которого будут получаться данные из 1С. Этот сеанс доступен для подключения к отладчику, т.к. это может потребоваться для отладки онлайн-обмена:



2. После этого можно запускать ТСД или «Клиент Mobile SMARTS для ПК», ставить точки останова и отлаживать, например, метод «ПолучитьСписокДокументов». На клиенте я зашел в список документов «Поступление»:



В итоге, толстый клиент зашел в метод «ПолучитьСписокДокументов»:

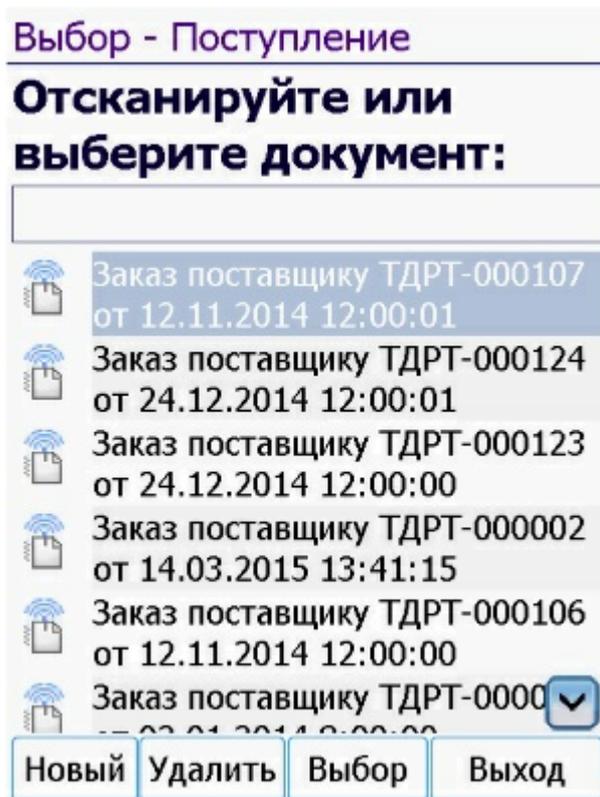
```
// обработчик события "ПолучитьСписокДокументов"
Функция ПолучитьСписокДокументов(UserId, ТипДокумента=Неопределено, xmlСессии = Неопределено) Экспорт
    storageConnector = Новый СОМОбъект("Cleverence.Warehouse.StorageConnector");
    ИдВазыСМАРТС = "";
    Если ЗначениеЗаполнено(xmlСессии) Тогда
        СессияСМАРТС = ПолучитьОбъектСМАРТС(storageConnector, xmlСессии);
        Если СессияСМАРТС <> Неопределено Тогда
            ИдВазыСМАРТС = СессияСМАРТС.ServerAppInstanceId;
        КонечЕсли;
    КонечЕсли;

    Подключиться(ИдВазыСМАРТС);

    СтрокиБП = ПолучитьБизнесПроцессы(ТипДокумента, ЭтотОбъект.СмартсНастройки);
    Если (СтрокиБП = Неопределено) ИЛИ (СтрокиБП <> Неопределено И СтрокиБП.Количество() = 0) Тогда
        Возврат Неопределено;
    КонечЕсли;

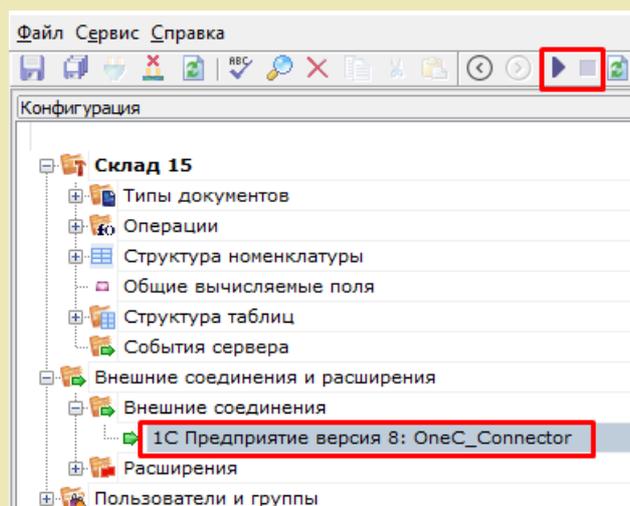
```

3. По результатам работы метода на клиенте отобразился список документов, полученных напрямую из 1С:



По аналогичным принципам работают все остальные онлайн-методы.

Если в обработку вносились изменения и она была пересохранена, то необходимо **ПЕРЕЗАПУСКАТЬ** «Внешнее соединение» в панели управления. Это связано с тем, что при запущенном толстом клиенте, обработка также остаётся запущенной. А после перезапуска она заново запустится с диска.



Настройки онлайн-обмена во внешней обработке

1. Понятие «торговый объект» означает, что все настройки хранятся в разрезе торговых объектов (как правило, это магазины, и для них на ТСД есть специальная таблица «Магазины»). Если ни один из магазинов не заведен, то его не требуется выбирать, и в таком случае используются глобальные настройки для всех магазинов. В противном случае, для каждого магазина будут браться свои настройки. Данный режим необходим, когда в одной базе 1С ведутся продажи по нескольким магазинам, тогда пользователь ТСД будет видеть информацию о документах, остатках, складах и т. д. только для своего магазина.
2. Среди настроек реализованы отбор справочников и отбор документов для отображения на ТСД, которые настраиваются в специальной обработке 1С. Отбор справочников для онлайн-обмена:

← → Клеверенс: Склад 15, Расширенный

Торговый объект: ×

Выгрузить
справочники

Открыть окно обмена
справочников

Открыть окно
обмена документами

Показать
настройки

[Информация по работе с обработкой](#)

← → Клеверенс: Настройка обмена справочников

Сохранить и закрыть

Переписывать номенклатуру на ТСД при выгрузке Выгружать номенклатуру документа при выгрузке документа

Разбить номенклатуру для выгрузки по: позиций (0 - не разбивать)

Установить поиск по наименованию и артикулу

Использовать шаблоны весового товара

Выбрать все

Снять все

Выгружать	Наименование	Выгрузка	Предпросмотр	Очистка справочника	Настройка
<input checked="" type="checkbox"/>	Номенклатура	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Характеристики номенклатуры	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Контрагенты	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Склады (места хранения)	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Остатки номенклатуры на складах	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input type="checkbox"/>	Цены номенклатуры	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Серии номенклатуры	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Ячейки	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input type="checkbox"/>	Дополнительные реквизиты	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить
<input checked="" type="checkbox"/>	Транспортные упаковки	Выгрузка...	Предпросмотр...	Очистка справочника...	Настроить

Если у нас в обработке выбран какой-то магазин (торговый объект) в качестве текущего, то настройка будут сохраняться для него. Иначе они будут считаться глобальными.

3. С помощью кнопки «Настроить» добавляем и сохраняем настройки для каждого из возможных справочников («Номенклатура», «Склады», «Контрагенты», «Серии»):

← → Клеверенс: Настройка и выгрузка справочника Номенклатура на ТСД

Параметры

Параметр	Значение
<input type="checkbox"/> Склад	
<input type="checkbox"/> Вид цены	
<input type="checkbox"/> Дата цен	Произвольная дата
Выгружать артикул	Да
Выгружать код	Да
Выгружать цены	Да
Выгружать остатки	Да

Отбор справочника (позволяет ограничить список выгрузки)

Представление	Значение
⊖ Отбор	
<input type="checkbox"/> = Номенклатура	Равно
<input type="checkbox"/> = Упаковка	Равно

Для проверки отбора жмём кнопку «Предпросмотр» и видим результат.

В итоге отбор настроен и на ТСД он будет использоваться в методе «ОбработатьЗапрос»:

Настройка отбора справочника Номенклатура для узла Центральная база детские товары (ТС.Предприятие)

Вернуться к настройкам

Добавить Ещё

Код	Артикул	Номенклатура	Характеристика	Упаковка	Штрихкод
00000000003		Бытовая техника			
00000000002		Кухонные элект...			
00000000023	КВ-900	Комбайн кухонн...			2000015
00000000023	КВ-900	Комбайн кухонн...			100 054
00000000027	К-7778	Кофеварка BRA...			2000015
00000000027	К-7778	Кофеварка BRA...			100 055
00000000004	КА-879	Кофеварка JAC...			2000015
00000000004	КА-879	Кофеварка JAC...			100 056
00000000037	M-678	Миксер BINATO...			2000015
00000000037	M-678	Миксер BINATO...			100 057

Параметры запроса:

Имя	Значение
ИспользоватьХарактерис...	Нет
П	Бытовая техника
СтрокаПоиска	
ЧислоПоиска	0

Текст запроса:

```

ВЫБРАТЬ
    СПРНоменклатура.Ссылка КАК Номенклатура,
    СПРНоменклатура.ВидНоменклатуры КАК ВидНоменклатуры,
    СПРНоменклатура.НаборУпаковок КАК НаборУпаковок,
    СПРНоменклатура.АлкогольнаяПродукция КАК АлкогольнаяПродукция,
    ЕСТЬNULL(СПРНоменклатура.ЕдиницаИзмерения.Наименование, "шт") КАК ЕдиницаИзмерения,
    ЕСТЬNULL(СПРНоменклатура.Производитель.Наименование, "") КАК Производитель,
    СПРНоменклатура.Артикул КАК Артикул,
    СПРНоменклатура.Весовой КАК Весовой,
    СПРНоменклатура.Код КАК Код,
    СПРНоменклатура.Наименование КАК Наименование,
    СПРНоменклатура.Родитель КАК Родитель,
    СПРНоменклатура.ЭтоГруппа КАК ЭтоГруппа,
    СПРНоменклатура.ВидНоменклатуры.ИспользоватьСерии КАК ИспользоватьСерии,
    СПРНоменклатура.ВидНоменклатуры.НастройкаИспользованияСерий КАК НастройкаИспользованияСерий,
    СПРНоменклатура.ВидНоменклатуры.ИспользованиеХарактеристик КАК ИспользованиеХарактеристик
ПОМЕСТИТЬ ВТ_Номенклатура
  
```

4. Отбор документов. Возможен только для тех бизнес-процессов, у которых указан исходный документ, т. е. те бизнес-процессы, которые подразумевают сперва выгрузку документа на ТСД, потом его загрузку обратно в 1С. Аналогично, отбор может быть глобальным либо для конкретного торгового объекта.

← → Клеверенс: Склад 15, Расширенный

Торговый объект: x Подключенная база Mobile SMARTS: Склад 15, Расширенный, версия: 1.2.4.43

Выгрузить справочники
Используйте бизнес процессы для настройки обмена документами в которых описаны правила выгрузки и загрузки документов. Список бизнес процессов общий для всех торговых объектов.

Открыть окно обмена справочников

Открыть окно обмена документами
Настройте список отбираемых типов документов из 1С, с которыми требуется дальнейшая работа на ТСД. Отборы могут быть настроены только по имеющимся настроенным бизнес-процессам с заполненным "Исходным документом".

Скрыть настройки

[Информация по работе с обработкой](#) [Используйте торговые объекты для разделения настроек справочников и отборов документов между](#)

5. При нажатии на «Настроено» будет выдано окно для настройки отборов по аналогии с отбором справочников.

← → Клеверенс: Настройки отборов бизнес-процессов (глобальные)

Онлайн

Выбрать все Снять все

	Название БП	Настройка	Исходный документ 1С (источник)
<input checked="" type="checkbox"/>	Приобретение товаров и услуг (проверка документа на ТСД)	Настроено	Приобретение товаров и услуг
<input checked="" type="checkbox"/>	Приобретение товаров и услуг (на основании "Заказа поставщику")	Настроено	Заказ поставщику
<input checked="" type="checkbox"/>	Приходный ордер на товары (на основании "Заказа поставщику")	Настроено	Заказ поставщику
<input checked="" type="checkbox"/>	Приходный ордер на товары (на основании "Поступления товаров")	Настроено	Приобретение товаров и услуг
<input checked="" type="checkbox"/>	Пересчет товаров (проверка документа на ТСД)	Настроено	Пересчет товаров
<input checked="" type="checkbox"/>	Пересчет товаров адресного склада (проверка на ТСД)	Настроено	Пересчет товаров
<input checked="" type="checkbox"/>	Перемещение товаров (Заказ на перемещение)	Настроено	Заказ на перемещение
<input checked="" type="checkbox"/>	Перемещение товаров (проверка документа на ТСД)	Настроено	Перемещение товаров
<input checked="" type="checkbox"/>	Реализация товаров и услуг (проверка документа на ТСД)	Настроено	Реализация товаров и услуг

Клеверенс: Настройка отбора документов ⋮ □ ×

Статусы
Использование отбора: Включено

ID Торгового объекта: Торговый объект:

ID Бизнес-процесса: ВРe8d6ec5a-7a78-41ba-8b01-820f78b31cdb Тип документа 1С: ПриобретениеТоваровУслуг

Тип обмена: Онлайн Наименование БП: Приобретение товаров и услуг (проверка документа на ТСД)

Вариант отбора: Только конструктор Свой запрос:

В отбор попадут документы, удовлетворяющие заданным параметрам компоновщика (см. Обязательные параметры и Параметры отбора).

Обязательные параметры

Параметр	Значение
Глубина отбора	
Комментарий отсеечения	

Параметры отбора

Представление			
Отбор			
<input checked="" type="checkbox"/>	= Проведен	Равно	Да
<input checked="" type="checkbox"/>	= Пометка удаления	Не равно	Да
<input checked="" type="checkbox"/>	= Согласован	Равно	Да

Сортировка

В итоге отбор настроен и на ТСД он будет использоваться в методе «ПолучитьСписокДокументов»:

Клеверенс: терминалы сбора данных (1С.Предприятие)

Клеверенс: терминалы сбора данных

[Вернуться к настройкам](#)

Результат запроса:

Ссылка
Поступление товаров ТДРТ-000001 от 03.01.2014 12:00:00
Поступление товаров ТДРТ-000002 от 05.01.2014 12:00:00
Поступление товаров ТДРТ-000003 от 11.01.2014 0:00:00
Поступление товаров ТДРТ-000004 от 12.01.2014 0:00:00
Поступление товаров ТДРТ-000005 от 15.01.2014 0:00:00
Поступление товаров ТДРТ-000006 от 16.01.2014 0:00:00
Поступление товаров ТДРТ-000007 от 16.01.2014 0:00:00
Поступление товаров ТДРТ-000008 от 19.01.2014 0:00:00
Поступление товаров ТДРТ-000009 от 22.01.2014 0:00:00

Параметры запроса:

Имя	Значение
ГлубинаОтбора	1 500
ДокументПроведен	Да
ТекущаяДата	20.03.2017 20:40:08

Текст запроса:

```

ВЫБРАТЬ
    ПоступлениеТоваров.Ссылка КАК Ссылка
ИЗ
    Документ.ПоступлениеТоваров КАК ПоступлениеТоваров
ГДЕ
    НАЧАЛОПЕРИОДА(ПоступлениеТоваров.Дата, ДЕНЬ) >= ДОБАВИТЬКДАТЕ(НАЧАЛОПЕРИОДА(ТекущаяДата, ДЕНЬ), ДЕНЬ, -ГлубинаОтбора)
    И ПоступлениеТоваров.Проведен = ДокументПроведен
  
```

6. Все данные о настройках хранятся в файле «customsettings.xml», в каталоге с базой Mobile SMARTS. При этом, настройки отборов хранятся либо с постфиксом «Глобальный», либо с ID торгового объекта. Аналогично — настройки отборов справочников, см. скрин.

3	<SettingsValueNode name="ИспользоватьШаблоныВесовогоТовара">
8	<SettingsValueNode name="ПрефиксКодаВесовогоТовара">
13	<SettingsValueNode name="ЧислоЗнаковПослеЗапятойВесТовара">
18	<SettingsValueNode name="ДлинаКодаВесовогоТовара">
23	<SettingsValueNode name="ВыгрузитьКонтрагентовСразу">
28	<SettingsValueNode name="ВыгрузитьНоменклатуруСразу">
33	<SettingsValueNode name="ВыгрузитьОстаткиСразу">
38	<SettingsValueNode name="ВыгрузитьСерииСразу">
43	<SettingsValueNode name="ВыгрузитьСкладыСразу">
48	<SettingsValueNode name="ВыгрузитьФормыАСразу">
53	<SettingsValueNode name="ВыгрузитьХарактеристикиСразу">
58	<SettingsValueNode name="ВыгрузитьЦеныСразу">
63	<SettingsValueNode name="МобильнаяПечать">
68	<SettingsValueNode name="ПерегисоватьНоменклатуруПриВыгрузке">
73	<SettingsValueNode name="иСоединениеКлевиренс">
78	<SettingsValueNode name="НастройкиБизнесПроцессов">
14600	<SettingsValueNode name="СписокОтборовБизнесПроцессов Глобальный Онлайн">
14800	<SettingsValueNode name="СписокОтборовБизнесПроцессов Глобальный Ручной">
14964	<SettingsValueNode name="СписокОтборовБизнесПроцессов Глобальный Автообмен">
15128	<SettingsValueNode name="ИдПриложения">
15133	<SettingsValueNode name="Ид">
15138	<SettingsValueNode name="ПодключеннаяБазаИмяКонфигурации1С">
15143	<SettingsValueNode name="ПодключеннаяБазаСинонимКонфигурации1С">
15148	<SettingsValueNode name="ПодключеннаяБазаСтрокаСоединения">
15153	<SettingsValueNode name="ПодключеннаяБазаВерсияПлатформы1С">
15158	<SettingsValueNode name="ПодключеннаяБазаВерсияКонфигурации1С">
15163	<SettingsValueNode name="СтрокаПодключенияСервера">
15168	<SettingsValueNode name="ВыполнятьНаСервере">
15173	<SettingsValueNode name="НастройкиКомпоновщикаНастроекВВыгрузкеНоменклатуры">
15290	<SettingsValueNode name="НастройкиКомпоновщикаНастроекХарактеристики">
15343	<SettingsValueNode name="ПользовательОсновнойБД">
15348	<SettingsValueNode name="Узлы">
15850	<SettingsValueNode name="ПараметрыУзлов">
16571	<SettingsValueNode name="ЗначенияПараметровУзлов">
16946	<SettingsValueNode name="Пользователи">
17410	<SettingsValueNode name="ЗначенияПроизвольныхПараметровУзлов">
18135	<SettingsValueNode name="ПроизвольныеКоды">
18187	<SettingsValueNode name="ID Узла">
18192	<SettingsValueNode name="НаименованиеУзла">
18197	<SettingsValueNode name="Узел1С">
18202	<SettingsValueNode name="СписокОтборовБизнесПроцессов ND896cf79162af40cbb970cbe6e5dd85f1 Онлайн">
18384	<SettingsValueNode name="НастройкиКомпоновщикаНастроекОстатки">
18430	<SettingsValueNode name="НастройкиКомпоновщикаНастроекЦены">
18492	<SettingsValueNode name="НастройкиКомпоновщикаНастроекСклады">
18549	<SettingsValueNode name="OneCProcessingFileName">
18554	<SettingsValueNode name="НастройкиОтборовДокументов">
19014	<SettingsValueNode name="ПарольПользователяОсновнойБД">
19019	<SettingsValueNode name="НастройкаОтбораСправочников ND896cf79162af40cbb970cbe6e5dd85f1 Номенклатура">
19136	</SettingsValueList>

Заключение

В данной статье мы показали, как можно работать с ТСД в режиме онлайн-обмена, а также рассмотрели все этапы по подготовке и настройке базы Mobile SMARTS и конфигурации 1С для работы в этом режиме.

Данный руководство дает основную базу для понимания принципов работы онлайн-обмена, который очень удобен в работе и при автоматизации рабочих мест.

Магазин 15, Алкоголь, ЕГАИС3, 1С, обмен данными, онлайн

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументДобавляется»

Последние изменения: 2024-03-26

Событие, возникающее в процессе добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы.

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который выгружается на сервер.
ИдПользователя
string (строка)
Передается null
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document). Если выполнение серверной операции прерывается (переход [прервать операцию]), добавление документа на сервер отменяется. Если отмена добавления не требуется, операция должна завершиться по переходу [завершить операцию].

В случае обычных коннекторов, если обработчик события вернул null, добавление документа на сервер отменяется.

Описание в [панели управления](#):

C#

<ид. коннектора>:ДобавлениеДокумента

Ид. коннектора — задается в панели управления.

Например: OneC_DriverConnector:ДобавлениеДокумента

Внешние соединения и расширения		Обработчики событий документов	
Внешние соединения		Документ возвращен с ТСД без обработки	
1С Предприятие версия 8: OneC_Connector		Документ добавлен	
		Документ добавляется	ДобавлениеДокумента
Структура номенклатуры		Порядок вызова обработчиков событий документов	
Общие вычисляемые поля		Документ возвращен с ТСД без обработки	
Структура таблиц		Документ добавлен	
События сервера		Документ добавляется	OneC_Connector:ДобавлениеДокумента

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументДобавлен»

Последние изменения: 2024-03-26

Событие, возникающее после добавления документа на сервере. Вызывается при выгрузке документа в базу Mobile SMARTS из внешней системы.

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который выгружен на сервер.
ИдПользователя
string (строка)
Передается null.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Обработчик может внести изменения в документ. После вызова обработчика документ сохраняется на сервере.

Описание в [панели управления](#):

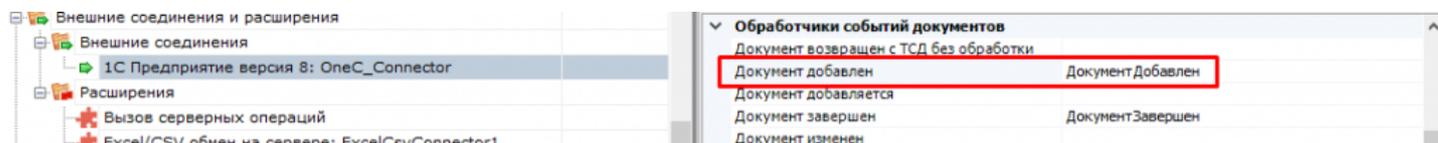
C#

<ид. коннектора>:ДокументДобавлен

Описание в панели управления:

Ид. коннектора - задается в панели управления.

Например: OneC_Connector:ДокументДобавлен



Структура таблиц	
События сервера	
Внешние соединения и расширения	
Внешние соединения	
1С Предприятие версия 8: OneC_Connector	

Порядок вызова обработчиков событий документов	
Документ возвращен с ТСД без обработки	
Документ добавлен	OneC_Connector:ДокументДобавлен
Документ добавляется	
Документ завершен	OneC_Connector:ДокументЗавершен



события сервера, коннекторы

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументНазначаетсяПользователю»

Последние изменения: 2024-03-26

Событие о том, что документ готов передаваться на мобильное устройство. Вызывается при запросе с ТСД получения документа для работы.

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который получает пользователь для работы на ТСД.
ИдПользователя
string (строка)
Ид. пользователя ТСД, который выполнил запрос.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document). Если выполнение серверной операции прерывается (переход [прервать операцию]), назначение документа пользователю ТСД отменяется. Если отмена назначения не требуется, операция должна завершиться по переходу [завершить операцию].

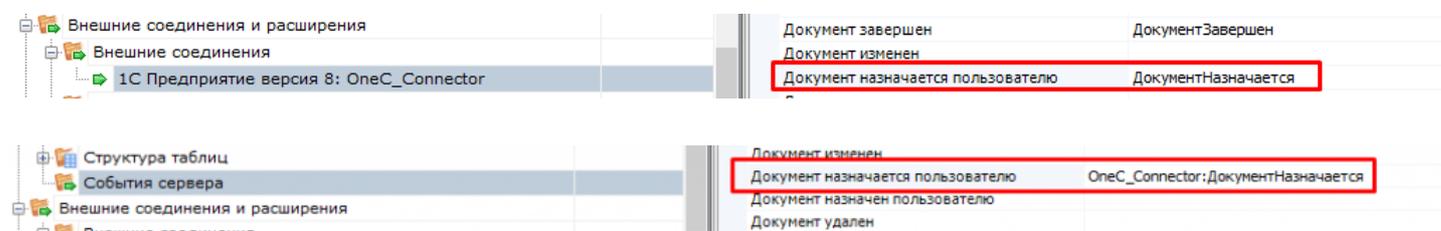
В случае обычных коннекторов, если обработчик события вернул null, назначение документа отменяется.

Описание в [панели управления](#):

```
C#
<ид. коннектора>;ДокументНазначается
```

Ид. коннектора - задается в панели управления.

Например: OneC_Connector:ДокументНазначается



Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументНазначенПользователю»

Последние изменения: 2024-03-26

Событие о том, что документ захвачен на обработку. Вызывается в момент, когда документ был передан на ТСД для работы с ним.

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который отправлен для работы на ТСД.
ИдПользователя
string (строка)
Ид. пользователя ТСД, который выполнил запрос.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Обработчик может внести изменения в документ. После вызова обработчика документ сохраняется на сервере.

Описание в [панели управления](#):

```
C#
.....
<ид. коннектора>:ДокументНазначен
```

Ид. коннектора - задается в панели управления.

Например: OneC_Connector:ДокументНазначен

<ul style="list-style-type: none"> Внешние соединения <ul style="list-style-type: none"> 1С Предприятие версия 8: OneC_Connector Расширения <ul style="list-style-type: none"> Вызов серверных операций 	<ul style="list-style-type: none"> Документ изменен Документ назначается пользователю Документ назначен пользователю Документ удален
<ul style="list-style-type: none"> Структура таблиц События сервера Внешние соединения и расширения 	<ul style="list-style-type: none"> Документ изменен Документ назначается пользователю Документ назначен пользователю OneC_Connector:ДокументНазначен Документ удален

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументИзменен»

Последние изменения: 2024-03-26

Событие об изменении документа. Вызывается при сохранении документа на сервер в процессе работы на ТСД при использовании в конфигурации Mobile SMARTS действия «Сохранение документа на сервер».

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который был изменен.
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Описание в [панели управления](#):

C#

<ид. коннектора>;ДокументИзменен

Ид. коннектора - задается в панели управления.

Например: OneC_DriverConnector:ДокументИзменен



события сервера, коннекторы

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументЗавершен»

Последние изменения: 2024-03-26

Событие о завершении обработки документа. Вызывается при получении сервером завершенного документа с ТСД. Может использоваться для автоматической загрузки завершенных документов в учетную систему. Учетная система может выполнить загрузку документа по ID с помощью функции GetDocument компоненты Cleverence.Warehouse.StorageConnector, после загрузки документ может быть удален из базы с помощью вызова RemoveDocument. Подробнее описано [здесь](#).

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который был завершен.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Описание в [панели управления](#):

C#
<ид. коннектора>;ДокументЗавершен

Ид. коннектора - задается в панели управления.

Например: OneC_Connector;ДокументЗавершен

<ul style="list-style-type: none"> Внешние соединения и расширения <ul style="list-style-type: none"> Внешние соединения <ul style="list-style-type: none"> 1С Предприятие версия 8: OneC_Connector Расширения 	<ul style="list-style-type: none"> Документ добавлен Документ добавляется Документ завершен Документ изменен 	<ul style="list-style-type: none"> ДокументДобавлен ДокументЗавершен
<ul style="list-style-type: none"> Структура таблиц События сервера Внешние соединения и расширения Пользователи и группы 	<ul style="list-style-type: none"> Документ добавляется Документ завершен Документ изменен Документ назначается пользователю 	<ul style="list-style-type: none"> OneC_Connector;ДокументЗавершен



Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS «ДокументВозвращенТсДБезОбработки»

Последние изменения: 2024-03-26

Вызывается, когда документ с терминала был возвращен пользователем вызовом release (вернуть документ без изменений).

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который получает пользователь для работы на ТСД.
ИдПользователя
string (строка)
Ид. пользователя ТСД, который выполнил запрос.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Описание в [панели управления](#):

C#

<ид. коннектора>;ДокументВозвращен

Ид. коннектора - задается в панели управления.

Например: OneC_Connector;;ДокументВозвращен

The screenshot shows the configuration interface for the OneC Connector. It displays two event handler configurations for the 'ДокументВозвращен' event:

- Обработчики событий документов**: This section shows two handlers: 'Документ возвращен с ТСД без обработки' (highlighted with a red box) and 'Документ добавлен'.
- Порядок вызова обработчиков событий документов**: This section shows the order of handlers, with 'Документ возвращен с ТСД без обработки' (highlighted with a red box) and 'OneC_Connector;ДокументВозвращен' (highlighted with a red box) listed.

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ДокументУдален»

Последние изменения: 2024-03-26

Событие об удалении документа с сервера. Удаление выполняется внешней системой.

Параметры

Имя параметра
Тип данных
Описание
ИдДокумента
string (строка)
Код документа, который получает пользователь для работы на ТСД.
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

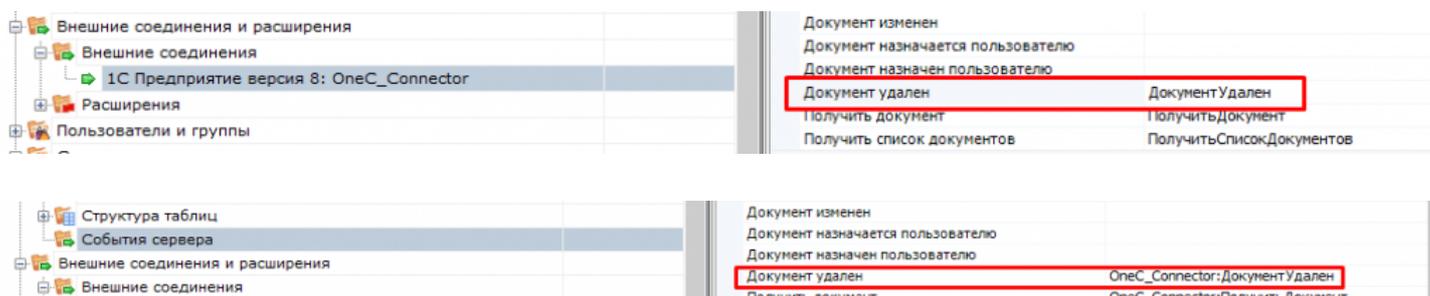
Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию передается сам документ (переменная сессии Document).

Описание в [панели управления](#):

```
C#
.....
<ид. коннектора>:ДокументУдален
```

Ид. коннектора - задается в панели управления.

Например: OneC_DriverConnector:ДокументУдален



Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ПолучитьДокумент»

Последние изменения: 2024-03-26

Событие о запросе получения документа по идентификатору или штрихкоду с сервера. Событие может использоваться совместно с событием «Получить список документов». Обработчик события «Получить список документов» возвращает список описаний документов (Cleverence.Warehouse.DocumentDescriptionCollection). Список отображается в окне выбора документа на ТСД, пользователь ТСД выбирает позицию из списка, идентификатор выбранного документа передается сервером в обработчик события «Получить документ». Обработчик «Получить документ» возвращает документ Mobile SMARTS (объект Cleverence.Warehouse.Document, сериализованный в xml). Также получение документа может выполняться по штрихкоду, отсканированному в окне выбора документов. В этом случае в параметрах типа документа должна быть включена настройка «Выбирать по штрихкоду с сервера — Да».

Параметры

Имя параметра
Тип данных
Описание
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
КодДокумента
string (строка)
Идентификатор или штрихкод запрашиваемого документа.
ТипДокумента
string (строка)
Тип документов, которые запрашиваются с терминала.
Режим
int (целое число)
=0 — получить документ по коду; =1 — получить документ по штрихкоду.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Возвращаемое значение

Функция должна вернуть объект типа Cleverence.Warehouse.Document в виде XML или пустую строку, если документ не найден.

Событие может быть обработано модулем «Вызов серверных операций». В этом случае в серверную операцию в переменную сессии Document передается результат вызова предыдущего обработчика в цепочке обработчиков данного события (если задано несколько обработчиков). Если предыдущих обработчиков в цепочке нет или результат вызова предыдущего обработчика = null, то создается новый документ с указанным типом. Также в сессию заносятся следующие значения: DocumentTypeName (тип запрошенного документа), DocumentId (ид. или штрихкод запрошенного документа), GetDocumentMode (режим получения документа: 0 — по ид., 1 — по штрихкоду). Возвращаемое значение (документ) операция должна занести в переменную Result.

Описание в [панели управления](#):

C#

<ид. коннектора>:ПолучитьДокумент

Ид. коннектора — задается в панели управления.

Например: OneC_Connector:ПолучитьДокумент

Пример функции

Для «1С:Предприятия 8»:

1С

Функция ПолучитьДокумент(UserId, КодДокумента, ОперацияТсд, Режим, mXmlDoc = Неопределено) Экспорт

//код обработки поиска документа

...

ДокументТсд = Новый СОМОбъект ("Cleverence.Warehouse.Document");

//заполнение документа Тсд данными из документа 1С

...

Результат = StorageConnector.ToXml (ДокументТсд);

Возврат Результат;

КонецФункции



события сервера, коннекторы

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS «ПолучитьСписокДокументов»

Последние изменения: 2024-03-26

Событие о запросе получения списка документов с сервера. Вызывается при открытии списка документов на терминале и периодически при нахождении внутри списка. Позволяет реализовать поиск и отбор документов по параметрам непосредственно в базе учетной системы, без предварительной выгрузки.

Вызов события может происходить только в том случае, если включен режим отображения серверных документов в списке документов терминала.

Если в панели управления в типах документов «Показывать в списке документы на сервере» проставлено «Нет», то никакого события на сервере не происходит. В случае, когда проставлено «Да», то терминал по необходимости будет запрашивать документы с сервера, заодно вызывая событие получения списка.

Параметры

Имя параметра
Тип данных
Описание
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
ТипДокумента
string (строка)
Тип документов, которые запрашиваются с терминала. Если тип документа null (неопределено) или пустая строка — запрашиваются все документы, вне зависимости от их типа.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий»

Возвращаемое значение

В качестве возвращаемого значения ожидается коллекция описаний документов Cleverence.Warehouse.DocumentDescriptionCollection, сохраненная в виде XML.

Событие может быть обработано модулем «Вызов серверных операций», в этом случае в серверную операцию в параметр Result передается результат вызова предыдущего обработчика в цепочке обработчиков данного события (если задано несколько обработчиков). Результат также возвращается в переменной Result. Серверная операция может изменить полученный из внешней системы список документов или вернуть свой собственный.

Описание в [панели управления](#):

C#

<ид. коннектора>:ПолучитьСписокДокументов

Ид. коннектора — задается в панели управления.

Например: OneC_Connector:ПолучитьСписокДокументов

Пример функции для «1С:Предприятия 8»

Ниже приведен пример метода «ПолучитьСписокДокументов» для использования в «1С:Предприятии» с целью получения списка документов в нужном формате. Процесс формирования списка документов на стороне 1С подробно не рассматривается (в данном примере он происходит в рамках функции «ОтобратитьДокументы (...)»).

1С

```

Функция ПолучитьСписокДокументов(UserId, ТипДокумента, mXmlDoc = Неопределено)
Экспорт
StorageConnector = Новый COMObject("Cleverence.Warehouse.StorageConnector");
ДокументыТСД = Новый COMObject("Cleverence.Warehouse.DocumentDescriptionCollection");
ОтобранныеДокументы = ОтобратитьДокументы(UserId, ТипДокумента);
Для Каждого СтрокаДок из ОтобранныеДокументы Цикл
DocDescr = Новый COMОбъект("Cleverence.Warehouse.DocumentDescription");
DocDescr.Id = XMLСтрока(СтрокаДок.Ссылка);
сDescr.Name = Строка(СтрокаДок.Ссылка);
DocDescr.DocumentTypeName = ТипДокумента;
DocDescrs.Add(DocDescr);
КонецЦикла;
Результат = StorageConnector.ToXml(DocDescrs);
Возврат Результат;
КонецФункции

```

В готовой интеграции «1С:Предприятия» и Mobile SMARTS рассматриваемая функция уже добавлена в модуль основной обработки «КлеверенсТСД_ОсновнаяОбработка.erf». Если вы делаете самостоятельную интеграцию Mobile SMARTS с «1С:Предприятием» с помощью основной обработки, то вы можете использовать готовые методы.

```

ОбработатьЗапрос
КонецПопытки;
Глядро_ЗаписатьСобытиеВЖурналРегистрации(ИмяСобытия, "Информация", "Конец, Ид пользователя = "+ Строка(UserId), НачалоСо
Возврат Результат;
КонецФункции
// Обрабатывает запрос сервера Mobile Smarts на получение списка документов //...
функция ПолучитьСписокДокументов UserId = Неопределено, ТипДокумента = Неопределено, mXmlDoc = Неопределено, АдресаПеременны
НачалоСобытия = ТекущаяУниверсальнаяДатаВМиллисекундах();
ИмяСобытия = "ПолучитьСписокДокументов";
Попытка
ТипИнтерфейса = Глядро_ПолучитьДанныеИзХранилища1С("Cleverence_ТипИнтерфейса");
Если АдресаПеременных = Неопределено Тогда
    АдресаПеременных = Комядро_СоздатьСтруктуруАдресаПеременных();
КонецЕсли;
НастройкиИнициализированы = Комядро_ПолучитьЗначениеПеременной("НастройкиИнициализированы", АдресаПеременных);
Глядро_ЗаписатьСобытиеВЖурналРегистрации(ИмяСобытия, "Информация", "Начало, Тип документа: "+ Строка(ТипДокумента)

```

Итогом выполнения функции «ПолучитьСписокДокументов» будет отображение документов из обработки 1С на экране мобильного устройства, подключенного к учетной системе (при работе в [онлайн-режиме](#)).

 события сервера, коннекторы

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ПолучитьТовар»

Последние изменения: 2024-03-26

Событие о том, что товар был запрошен терминалом по идентификатору или каким-либо реквизитам (штрихкод, артикул, код) с сервера и не был найден в серверном справочнике. Позволяет реализовать поиск товаров непосредственно в базе учетной системы, без предварительной выгрузки.

Для того, чтобы выполнялся вызов обработчика события, в настройках номенклатуры должен быть включен «Поиск на сервере» и «Поиск во внешней системе».

+	Типы документов	
+	Операции	
+	Структура номенклатуры	
+	Общие вычисляемые поля	
+	Структура таблиц	
+	События сервера	
	Поиск во внешней системе	Да
	Поиск локально на устройстве	Нет
	Поиск на сервере	Да
	Сервер в приоритете	Да
	Общие	
	Общие товарные шаблоны	<Список...>

Параметры

Имя параметра
Тип данных
Описание
ИдТовара
string (строка)
Идентификатор или атрибут для поиска (штрихкод, артикул, код) для поиска.
ИдУпаковки
string (строка)
Идентификатор упаковки товара. Заполняется только для режимов 2 и 3.
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
Режим
int (целое число)
Режим поиска =0 — поиск по коду товара, без указания конкретной упаковки. =1 — поиск по штрихкоду, артикулу или любой другой характеристике товара. =2 — поиск по коду товара с заданной упаковкой. =3 — поиск упаковки для товара.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий»

Возвращаемое значение

В качестве возвращаемого значения ожидаются разные типы данных, в зависимости от режима:

[Cleverence.Warehouse.Product](#), [Cleverence.Warehouse.Packing](#), [Cleverence.Warehouse.PackedProduct](#) или [Cleverence.Warehouse.PackedProductCollection](#) в виде XML.

Режим 0:

Требуется найти товар по его идентификатору. Возвращаемое значение [Cleverence.Warehouse.Product](#) в виде XML. Если товар не найден, необходимо вернуть null (неопределено).

Следует возвращать товар с базовым типом упаковки. Если другие упаковки потребуются в процессе работы, программа запросит их с помощью режима 2 или 3.

В данном режиме событие вызывается, когда на ТСД было вызвано получение товара по идентификатору. Например, при отображении строк документа в списке выполняется получение товаров по их идентификаторам из строк документа.

Режим 1:

Требуется найти товар по штрихкоду, артикулу или иному реквизиту, по которому предусмотрен поиск. Возвращаемое значение [Cleverence.Warehouse.PackedProduct](#) или [Cleverence.Warehouse.PackedProductCollection](#) в виде XML. Если товар не найден, необходимо вернуть null (неопределено).

В данном режиме событие вызывается при сканировании на ТСД штрихкода товара в действии Выбор номенклатуры или при вводе пользователем значения, по которому требуется найти товар.

Режим 2:

Требуется найти товар по коду товара с заданной упаковкой. Возвращаемое значение [Cleverence.Warehouse.Product](#) в виде XML. Если товар не найден, необходимо вернуть null (неопределено). Отличие от режима 0 состоит в том, что в режиме 0 должен возвращаться товар с базовой упаковкой, а в данном режиме запрашивается товар с конкретной упаковкой.

Режим 3:

Требуется найти упаковку товара. Возвращаемое значение [Cleverence.Warehouse.Packing](#) в виде XML. Если товар не найден, необходимо вернуть null (неопределено).

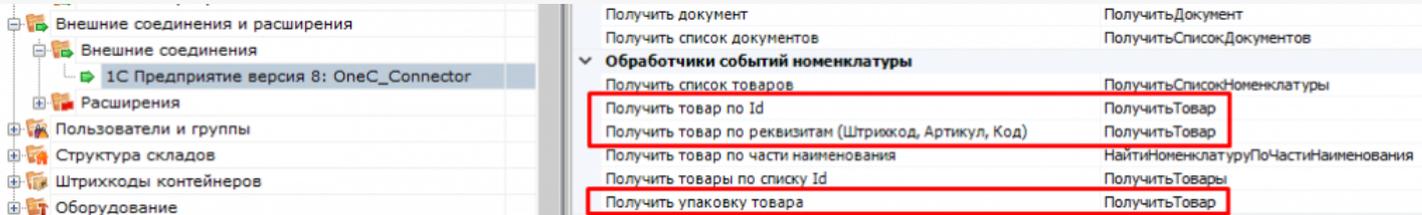
Описание в [панели управления](#):

C#

<ид. коннектора>:ПолучитьТовар

Ид. коннектора — задается в панели управления.

Например: OneC_Connector:ПолучитьТовар



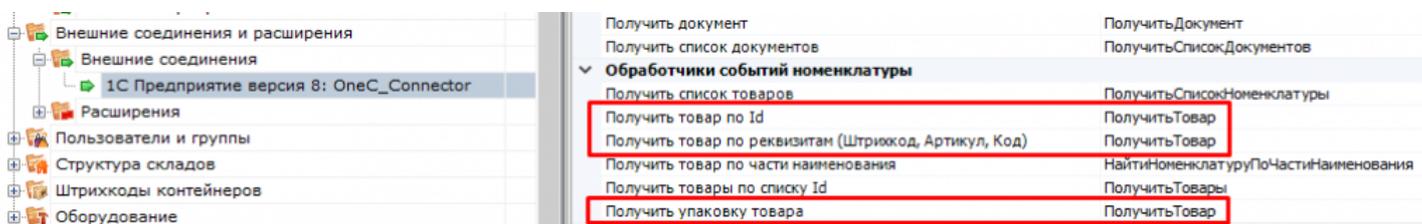
При использовании коннекторов, у которых имена обработчиков указываются в свойствах самого коннектора (производные от ConnectorTypical), есть возможность указать разные обработчики для разных режимов получения товара:

Режим 0 и Режим 2 — «Получить товар по Id»,

Режим 1 — «Получить товар по реквизитам (Штрихкод, Код, Артикул)»,

Режим 3 — «Получить упаковку товара».

Параметры обработчиков те же, что описаны выше.



Пример функции

Для «1С:Предприятия 8»:

```
1С
.....

Функция ПолучитьТовар(ПараметрНоменклатуры, ПараметрУпаковки, UserId, Режим = 0,
mXmlDoc = Неопределено) Экспорт
Результат = Неопределено;
//код обработки поиска товара
...
StorageConnector = Новый СОМОбъект("Cleverence.Warehouse.StorageConnector");
//создание объекта драйвера для передачи данных в компоненту
PackedProductCollection = Новый СОМОбъект("Cleverence.Warehouse.PackedProductCollection");
...
//преобразование объекта драйвера в формат XML
Результат = StorageConnector.ToXml (PackedProductCollection);
//передача данных на сервер Mobile SMARTS
Возврат Результат ;
КонецФункции
```

Кроме изложенного варианта возврата результата из обработчика в виде xml-представления объектов Cleverence.Warehouse.Product, ProductCollection, PackedProduct и т. п., есть возможность возвращать из 1С таблицу значений. Наименования колонок таблицы значений должны начинаться на «Product_», если поле относится к товару и на «Packing_», если это поле упаковки. Например, «Product_Id», «Packing_Barcode», «Packing_Характеристика». Поле объекта PackedProduct (возвращается в режиме 2), не относящиеся

ни к товару, ни к упаковке, указывается без префиксов «Product_» и «Packing_» (например, «Quantity»).

Пример:

Product_Id	Packing_Marking	Product_Marking	Product_Barcode	Packing_Barcode	Product_Name	Product_BasePackingId	Packing_Id	Packing_Name
"dee6e1d9-5...	"K-120001"	"K-120001"	"00000000018"	"2000019017158"	"Кроссовки мужские, кожа"	"пара"	"пара"	"пара"

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS «ПолучитьТоварыПоСпискуId»

Последние изменения: 2024-03-26

Обработчик события вызывается при запросе с терминала группы товаров с конкретными упаковками, если товары не найдены в выгруженном справочнике.

Для того, чтобы выполнялся вызов обработчика события, в настройках номенклатуры должен быть включен «Поиск на сервере» и «Поиск во внешней системе» .

Параметры

Имя параметра
Тип данных
Описание
ИдТоваров
Массив строк (COMSafeArray)
Массив идентификаторов запрошенных товаров.
ИдУпаковок
Массив строк (COMSafeArray)
Массив идентификаторов упаковок.
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий».

Массивы «ИдТоваров» и «ИдУпаковок» имеют одинаковую длину и содержат пары «ИдТовара-ИдУпаковки» данного товара.

Событие аналогично событию «Получить товар» в режиме 2 (поиск по коду товара с заданной упаковкой). Отличие в том, что «Получить товар принимает одну пару ИдТовара-ИдУпаковки» , а «Получить товары по списку Id» набор таких пар.

Возвращаемое значение

Результат, возвращаемый функцией, должен быть в виде Xml-представления объекта Cleverence.Warehouse.ProductCollection (коллекция товаров, см. [справочник](#)). Xml-представление объектов Mobile SMARTS следует получать с помощью функции ToXml компоненты StorageConnector.

В случае 1С функция может возвращать таблицу значений, которая на сервере Mobile SMARTS преобразуется в объект коллекции. Наименования колонок таблицы значений должны начинаться на "Product_", если поле

относится к товару и на “Packing_”, если это поле упаковки. Например, “Product_Id”, “Packing_Barcode”, “Packing_Характеристика”.

Обработчик события может найти не все запрошенные товары, тогда в таблице должны быть только строки, соответствующие найденным товарам. Если не найдено ни чего, возвращается пустая таблица.

Описание в [панели управления](#):

C#
<ид. коннектора>:ПолучитьТовары

Ид. коннектора - задается в панели управления.

Например: OneC_Connector:ПолучитьТовары

Пример функции

Для «1С:Предприятия 8»:

C#

```
Функция ПолучитьТовары(ИдТоваров, ИдУпаковок, UserId, mXmlDoc=Неопределено) Экспорт
ЗапросНоменклатуры = Новый Запрос;
...
СписокНоменклатуры = Новый СписокЗначений;
Для Каждого Ид из ИдТоваров Цикл
СписокНоменклатуры.Добавить(ПолучитьСсылку("Номенклатура", Ид));
КонецЦикла;
ЗапросНоменклатуры.УстановитьПараметр("П", СписокНоменклатуры);
ТаблицаТоваров = ЗапросНоменклатуры.Выполнить().Выгрузить();
Результат = Новый ТаблицаЗначений;
Результат.Колонки.Добавить("Product_Id");
Результат.Колонки.Добавить("Product_Marking" );
Результат.Колонки.Добавить("Product_Barcode" ); Результат.Колонки.Добавить("Product_Name"
);
Результат.Колонки.Добавить("Product_BasePackingId" );
...
Результат.Колонки.Добавить("Packing_Id" );
Результат.Колонки.Добавить("Packing_Name" );
Результат.Колонки.Добавить("Packing_Marking" );
Результат.Колонки.Добавить("Packing_Barcode" );
...
Для Каждого СтрокаТовара Из ТаблицаТоваров Цикл
СтрокаРезультата = Результат.Добавить();
СтрокаРезультата["Product_Id"] = СтрокаТовара["ИдНоменклатуры"];
...
КонецЦикла;
Возврат Результат;
КонецФункции
```

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS

«ПолучитьТоварПоЧастиНаименования»

Последние изменения: 2024-03-26

Обработчик события вызывается, когда пользователь на терминале вводит в окне выбора номенклатуры из списка текст для поиска позиций номенклатуры.

Для того, чтобы выполнялся вызов обработчика события, в настройках номенклатуры должен быть включен «Поиск на сервере» и «Поиск во внешней системе».

Параметры

Имя параметра
Тип данных
Описание
ТекстДляПоиска
string (строка)
Текст, введенный пользователем на терминале.
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.
XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS (Cleverence.Warehouse.ServerSession см. справочник), сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий»

Возвращаемое значение

Результат, возвращаемый функцией, должен быть в виде Xml-представления объекта Cleverence.Warehouse.PackedProductCollection (коллекция товаров с упаковками, см. [справочник](#)). Xml-представление объектов Mobile SMARTS следует получать с помощью функции ToXml компоненты StorageConnector.

В случае 1С функция может возвращать таблицу значений, которая на сервере Mobile SMARTS преобразуется в объект коллекции. Наименования колонок таблицы значений должны начинаться на «Product_», если поле относится к товару и на «Packing_», если это поле упаковки. Например, «Product_Id», «Packing_Barcode», «Packing_Характеристика».

Описание в [панели управления](#):

```
C#
```

```
<ид. коннектора>:НайтиНоменклатуруПоЧастиНаименования
```

Ид. коннектора — задается в панели управления.

Например: OneC_DriverConnector:НайтиНоменклатуруПоЧастиНаименования

Пример функции

Для «1С:Предприятия 8»:

```
1С
```

```
Функция НайтиНоменклатуруПоЧастиНаименования(ТекстДляПоиска, userId,
mXmlDoc=Неопределено) Экспорт
//код обработки поиска товара
...
StorageConnector = Новый СОМОбъект("Cleverence.Warehouse.StorageConnector");
//создание объекта драйвера для передачи данных в компоненту
PackedProductCollection = Новый СОМОбъект("Cleverence.Warehouse.PackedProductCollection")
...
//преобразование объекта драйвера в формат XML
Результат = StorageConnector.ToXml (PackedProductCollection);
//передача данных на сервер Mobile SMARTS
Возврат Результат;
КонецФункции
```



события сервера, коннекторы

Не нашли что искали?



Задать вопрос в техническую поддержку

Событие сервера Mobile SMARTS «ПолучитьСписокТоваров»

Последние изменения: 2024-03-26

Обработчик события вызывается при отображении на терминале справочника номенклатуры с сервера, позволяет организовать вывод иерархического справочника номенклатуры из учетной системы.

Аргументы:

Имя параметра	Тип данных	Описание
ИдРодителя		
string (строка)		
Идентификатор позиции (папки), вложенные элементы которой следует вернуть. Для верхнего уровня - пустая строка или Null.		
ИдПользователя		
string (строка)		
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.		
XmlСессии		
string (строка)		
Xml представление объекта Cleverence.Warehouse.ServerSession (см. справочник) или Null (если в настройках событий сервера в конфигурации Mobile SMARTS отключено добавление сессии в обработчики событий). Объект может быть загружен с помощью функции FromXml объекта Cleverence.Warehouse.StorageConnector.		

В панели управления:

Свойства	
Общее	
Добавлять объект сессии в вызов событий	Да
События	
Документ возвращен без изменений	
ДокументДобавлен	
ДокументЗавершен	OneC_Connector:ДокументЗавершен
ДокументИзменен	
ДокументПолученНаОбработку	
ДокументУдален	
НайтиНоменклатуруПоЧастиНаименования	OneC_Connector:НайтиНоменклатуруПоЧастиНаименования
Обработать запрос	OneC_Connector:ОбработатьЗапрос
ПолучитьДокумент	OneC_Connector:ПолучитьДокумент
ПолучитьСписокДокументов	OneC_Connector:ПолучитьСписокДокументов
ПолучитьСписокНоменклатуры	OneC_Connector:ПолучитьСписокНоменклатуры
ТоварНеНайден	OneC_Connector:ПолучитьТовар

<id коннектора>:<имя функции-обработчика>

При запросе позиций верхнего уровня ИдРодителя пустой. Когда пользователь на терминале при просмотре

списка номенклатуры выбирает элемент, являющийся группой, в параметр ИдРодителя передается идентификатор выбранной группы. При этом функция должна вернуть элементы, вложенные в группу (элементы могут сами являться группами или позициями номенклатуры).

Результат, возвращаемый функцией, должен быть в виде Xml-представления объекта `Cleverence.Warehouse.PackedProductCollection` (коллекция товаров с упаковками, см. справочник). Xml-представление объектов Mobile SMARTS следует получать с помощью функции `ToXml` компоненты `StorageConnector`.

В случае 1С функция может возвращать таблицу значений, которая на сервере Mobile SMARTS преобразуется в объект коллекции. Наименования колонок таблицы значений должны начинаться на "Product_", если поле относится к товару и на "Packing_", если это поле упаковки. Например, "Product_Id", "Packing_Barcode", "Packing_Характеристика".

Если возвращаемый элемент является группой, у объекта `PackedProduct`, добавляемого в коллекцию, не должно быть заполнено свойство `Packing` (упаковка), этот признак используется клиентом Mobile SMARTS для определения того, что внутрь элемента при выборе его из списка можно зайти. В случае таблицы значений в строке, представляющей группу, должны быть заполнены только колонки товара (`Product`) и не заполнены колонки упаковки (`Packing`).

Пример

1С

Функция ПолучитьСписокНоменклатуры(ИдРодителя, userId, mXmlDoc=Неопределено) Экспорт

Родитель = Неопределено;

Если Не ЗначениеЗаполнено(ИдРодителя) Тогда

Родитель = Справочники.Номенклатура.ПустаяСсылка();

Иначе

Попытка

Гуид = Новый УникальныйИдентификатор(ИдРодителя);

Родитель = Справочники.Номенклатура.ПолучитьСсылку(Гуид);

Исключение

Родитель = Справочники.Номенклатура.ПустаяСсылка();

КонецПопытки;

КонецЕсли;

ЗапросНоменклатуры = Новый Запрос;

...

ЗапросНоменклатуры.УстановитьПараметр("Родитель", Родитель);

```
ТаблицаТоваров = ЗапросНоменклатуры.Выполнить().Выгрузить();
```

```
Результат = Новый ТаблицаЗначений;
```

```
Результат.Колонки.Добавить("Product_Id");
```

```
Результат.Колонки.Добавить("Product_Marking" );
```

```
Результат.Колонки.Добавить("Product_Barcode" ); Результат.Колонки.Добавить("Product_Name" );
```

```
Результат.Колонки.Добавить("Product_BasePackingId" );
```

```
...
```

```
Результат.Колонки.Добавить("Packing_Id" );
```

```
Результат.Колонки.Добавить("Packing_Name" );
```

```
Результат.Колонки.Добавить("Packing_Marking" );
```

```
Результат.Колонки.Добавить("Packing_Barcode" );
```

```
...
```

```
Для Каждого СтрокаТовара Из ТаблицаТоваров Цикл
```

```
СтрокаРезультата = Результат.Добавить();
```

```
СтрокаРезультата["Product_Id"] = СтрокаТовара["ИдНоменклатуры"]; 
```

```
СтрокаРезультата["Product_Name"] = Строка(СтрокаТовара["Номенклатура"]);
```

```
Если СтрокаТовара["ЭтоГруппа"] Тогда
```

```
СтрокаРезультата["Product_ЭтоГруппа"] = Истина;
```

```
Иначе
```

```
...
```

```
СтрокаРезультата["Product_Marking"] = Артикул;
```

```
СтрокаРезультата["Product_Barcode"] = Код;
```

```
СтрокаРезультата["Packing_Barcode"] = ШК;
```

```
СтрокаРезультата["Product_BasePackingId"] = ФлагБазовойЕдиницы;
```

```
СтрокаРезультата["Packing_Id"] = ЕдиницаИзмерения;
```

```
СтрокаРезультата["Packing_Name"] = ЕдиницаИзмерения;
```

...

КонецЕсли;

КонецЦикла;

Возврат Результат;

КонецФункции



события сервера, коннекторы

Не нашли что искали?



[Задать вопрос в техническую поддержку](#)

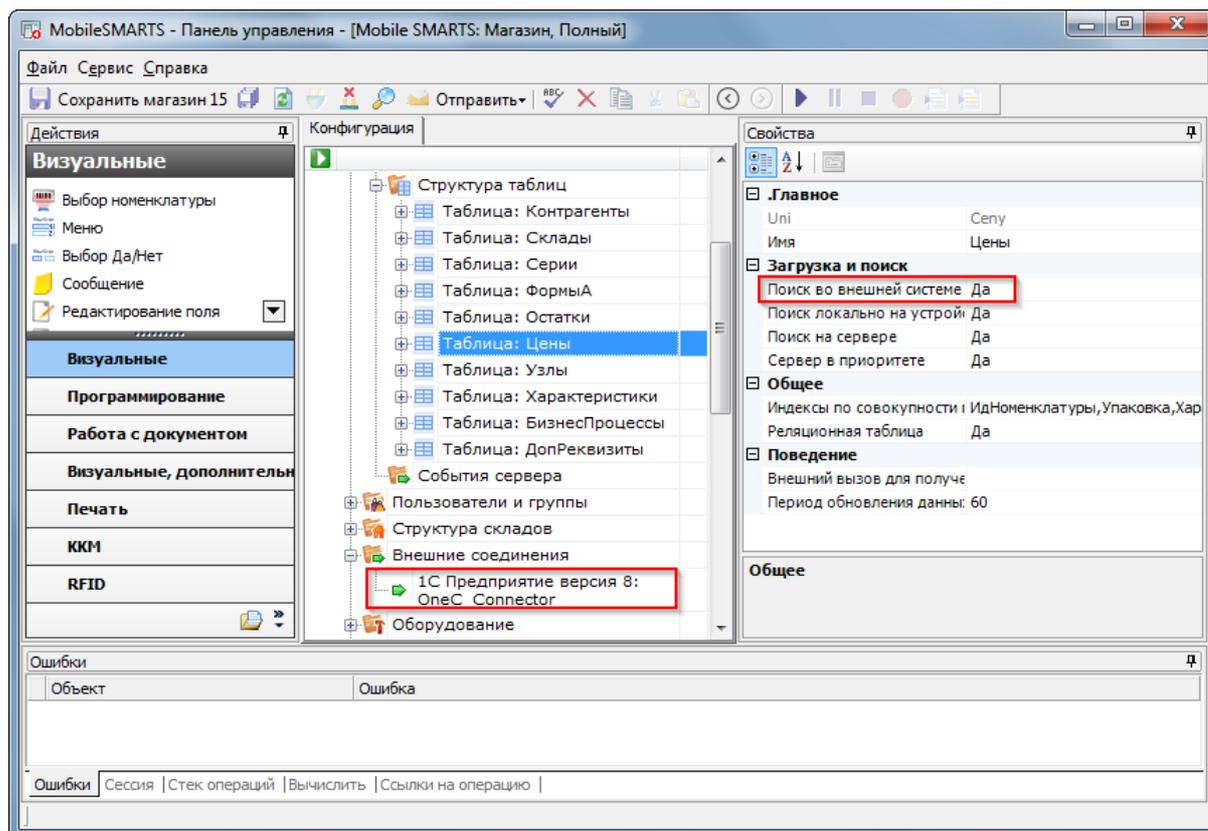
Событие сервера Mobile SMARTS

«ОбработатьЗапрос»

Последние изменения: 2024-03-26

Обработчик события вызывается при выполнении запроса к серверной таблице, имеющейся в конфигурации Mobile SMARTS. Данный механизм позволяет организовать получение по запросу с терминала различных данных, которые при работе без онлайн вызовов, хранятся в выгружаемых таблицах (например, цены, остатки, серии).

Чтобы обработчик вызывался при запросе данных из таблицы, в настройках таблицы должно быть включено «Поиск на сервере — Да», «Поиск во внешней системе — Да».



Параметры

Имя параметра
Тип данных
Описание
ЗапросXML
string (строка)
Массив идентификаторов запрошенных товаров.
ИдПользователя
string (строка)
Идентификатор пользователя Mobile SMARTS, при работе которого на ТСД, произошел вызов обработчика.

XmlСессии
string (строка)
Объект сессии сервера Mobile SMARTS, сериализованный в xml. Параметр передается, только если включена настройка «Добавлять объект сессии в вызов событий»

Для обработки запроса необходимо загрузить из xml объект Cleverence.Warehouse.DocumentQuery с помощью функции FormXML объекта Cleverence.Warehouse.StorageConnector.

DocumentQuery содержит следующие свойства, используемые при обработке запроса:

Свойство
Тип данных
Описание
From
Строка
Наименование таблицы, из которой выполняется выборка данных (например, Остатки). Равно наименованию одной из таблиц, имеющихся в конфигурации Mobile SMARTS.
WhereRootElement
Ссылка на объект IQueryElement
Корневой элемент синтаксического дерева выражения условия в запросе.

По значению, заданному в свойстве From, обработчик должен определить, откуда следует выбрать данные. Далее нужно обойти элементы дерева выражения условия и наложить соответствующее условие на запрос учетной системы. IQueryElement содержит:

Свойство
Тип данных
Описание
IsGroup
Boolean
Признак, является ли элемент группой (И, ИЛИ, НЕ)
Группа: IsGroup = Истина например, (ИдНоменклатуры == «cbcf493e-55bc-11d9-848a-00112f43529a» && Характеристика == «светло-серый» && ВидЦеныВнутр == 1)
GroupTypeStr
Строка
Тип группы (какой логической операцией объединены условия в группе). Одно из следующих значений: «Or» (Или), «And» (И), «Not» (Не).
Elements
QueryElementCollection
Коллекция элементов группы, может содержать как отдельные элементы условия (IsGroup=Ложь), так и другие группы (IsGroup=Истина).

Элемент: IsGroup = Ложь например, ИдНоменклатуры == «cbcf493e-55bc-11d9-848a-00112f43529a»
--

ComparisonTypeStr
Строка
Вид сравнения. Одно из следующих значений: «==» (равно), «!=» (не равно), «<» (меньше), «>» (больше), «<=» (меньше или равно), «>=» (больше или равно), «Contains» (содержит), «StartsWith» (начинается с).
LeftValue
Строка
Наименование сравниваемого поля
RightValue
Object
Значение сравниваемого поля (число, строка).

Возвращаемое значение

Результат, возвращаемый функцией, должен быть в виде Xml-представления объекта `Cleverence.Warehouse.RowCollection` (коллекция строк таблицы, см. [справочник](#)). Xml-представление объектов Mobile SMARTS следует получать с помощью функции ToXml компоненты StorageConnector.

В случае 1С функция может возвращать таблицу значений. Наименования полей (как в случае строк `Cleverence.Warehouse.Row`, так и в случае колонок таблицы значений 1С) должны соответствовать наименованиям полей таблицы из конфигурации Mobile SMARTS.

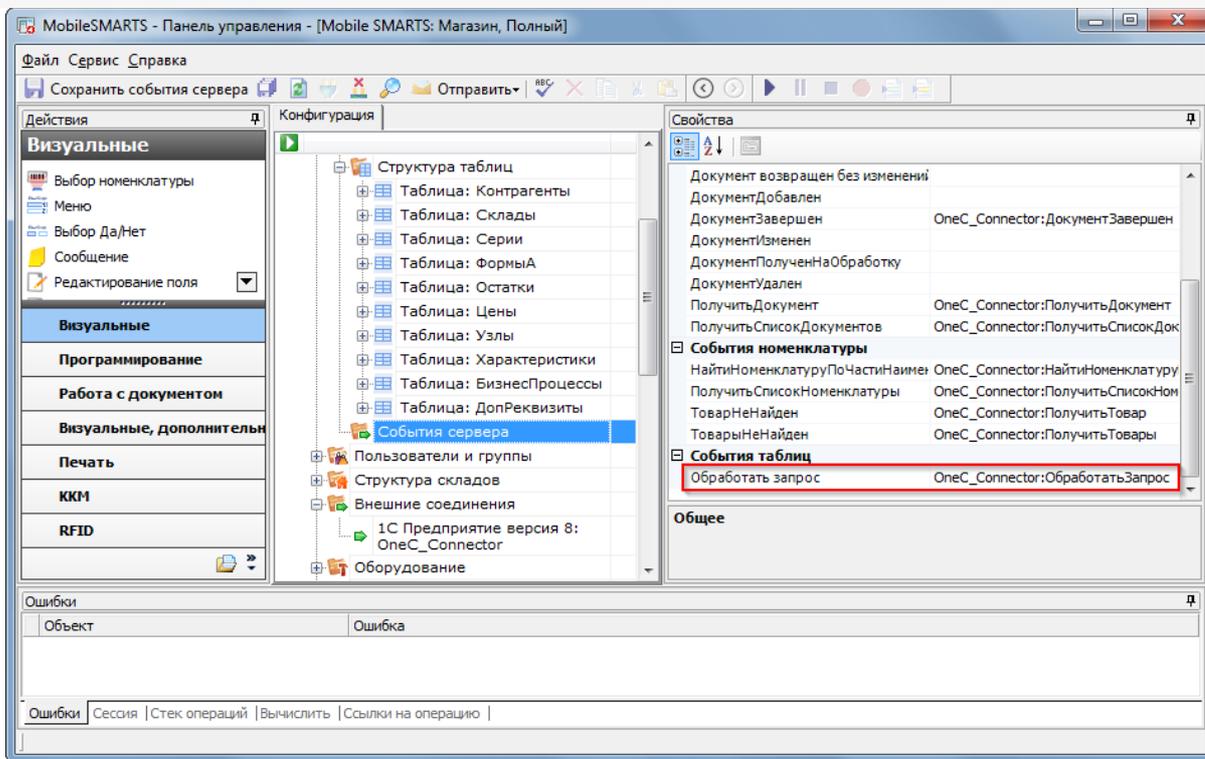
Описание в [панели управления](#):

1С

<ид. коннектора>:ОбработатьЗапрос

Ид. коннектора — задается в панели управления.

Например: `OneC_Connector:ОбработатьЗапрос`



Пример функции

1С

Функция ОбработатьЗапрос(запросXML, userId, mXmlDoc=Неопределено) Экспорт

...

docQuery = connector.FromXML(запросXML);

ИмяМакетаСхемыКомпоновки = ПолучитьИмяМакета(docQuery.From);

СхемаКомпоновки = ЭтотОбъект.ПолучитьМакет(ИмяМакетаСхемыКомпоновки);

...

КомпоновщикНастроекКомпоновкиДанных = Новый

КомпоновщикНастроекКомпоновкиДанных;

...

ЗаполнитьОтбор(docQuery.From, docQuery.WhereRootElement,

КомпоновщикНастроекКомпоновкиДанных.Настройки.Отбор.Элементы);

ПроцессорВывода = Новый

ПроцессорВыводаРезультатаКомпоновкиДанныхВКоллекциюЗначений;

ТабВыгрузки = ПроцессорВывода.Вывести(ПроцессорКомпоновки);

Возврат ТабВыгрузки;

КонецФункции

Процедура ЗаполнитьОтбор(ИмяОтбора, queryElement, Элементы)

Если queryElement = NULL Или queryElement = Неопределено Тогда

Возврат;

КонецЕсли;

Если queryElement.IsValueItem Тогда

Возврат;

КонецЕсли;

Если Не queryElement.IsGroup Тогда

ЭлементОтбора = Элементы.Добавить(Тип("ЭлементОтбораКомпоновкиДанных"));

ЭлементОтбора.ВидСравнения = ПолучитьВидСравнения(queryElement.ComparisonTypeStr);

ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных(queryElement.LeftValue);

ЭлементОтбора.ПравоеЗначение = ПолучитьЗначениеДляОтбора(ИмяОтбора,
queryElement.LeftValue, queryElement.RightValue);

Иначе

ЭлементОтбора = Элементы.Добавить(Тип("ГруппаЭлементовОтбораКомпоновкиДанных"));

ЭлементОтбора.ТипГруппы = ПолучитьТипГруппыОтбора(queryElement.GroupTypeStr);

Для Инд = 0 По queryElement.Elements.Count-1 Цикл

queryEl = queryElement.Elements.Item(Инд);

ЗаполнитьОтбор(ИмяОтбора, queryEl, ЭлементОтбора.Элементы);

КонецЦикла;

КонецЕсли;

КонецПроцедуры



события сервера, коннекторы

Не нашли что искали?



Задать вопрос в техническую поддержку

Ручная отправка событий для документов на сервере Mobile SMARTS

Последние изменения: 2024-03-26

Часто в процессе разработки или внедрения необходимо как-то добиться повторного возникновения **события на сервере** (например, «Документ завершен»).

Ранее разработчику приходилось повторно отправлять документ на ТСД и завершать его на мобильном устройстве, чтобы сервер заново вызвал обработку события.

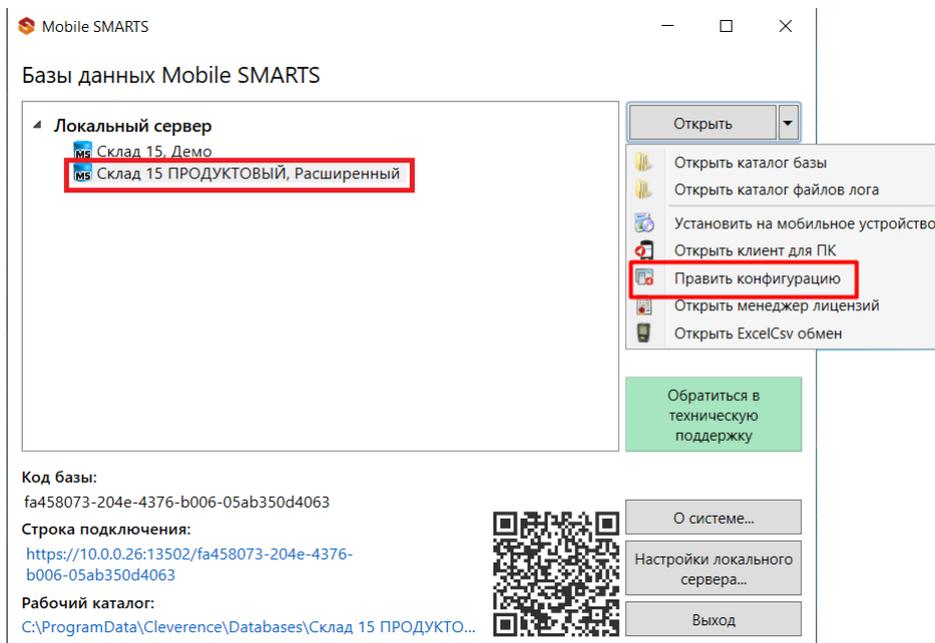
Начиная с версии 3.2 платформы Mobile SMARTS, в **панели управления** становится доступной функция ручной отправки основных событий для документов:

- **ДокументЗавершен.**
- **ДокументДобавлен.**
- **ДокументИзменен.**
- **ДокументНазначен.**

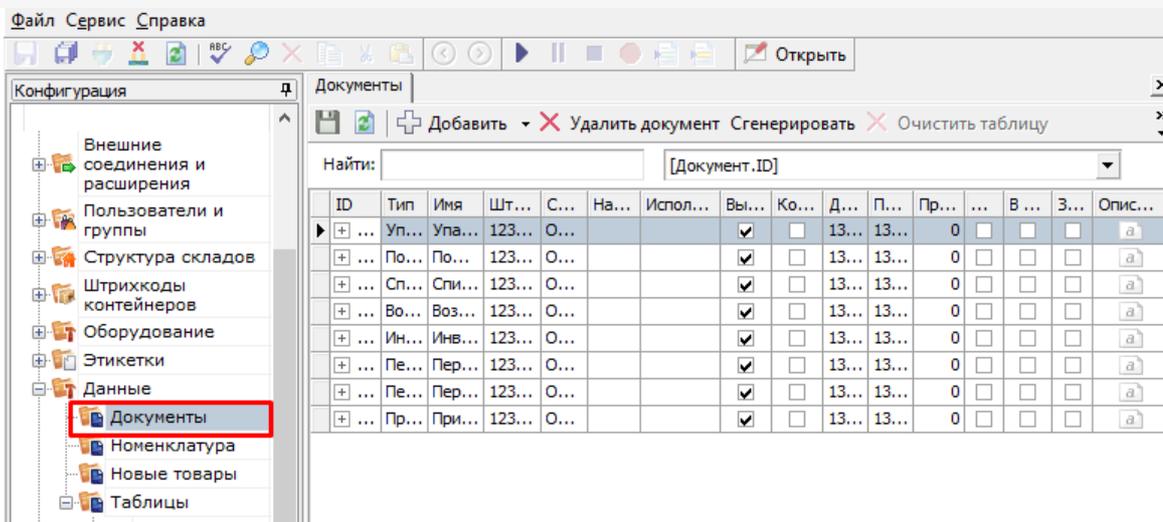
Это значит, что для вызова серверного события теперь не нужно производить какие-либо действия с документом на устройстве, достаточно просто нажать на одну кнопку в панели управления.

Для этого необходимо:

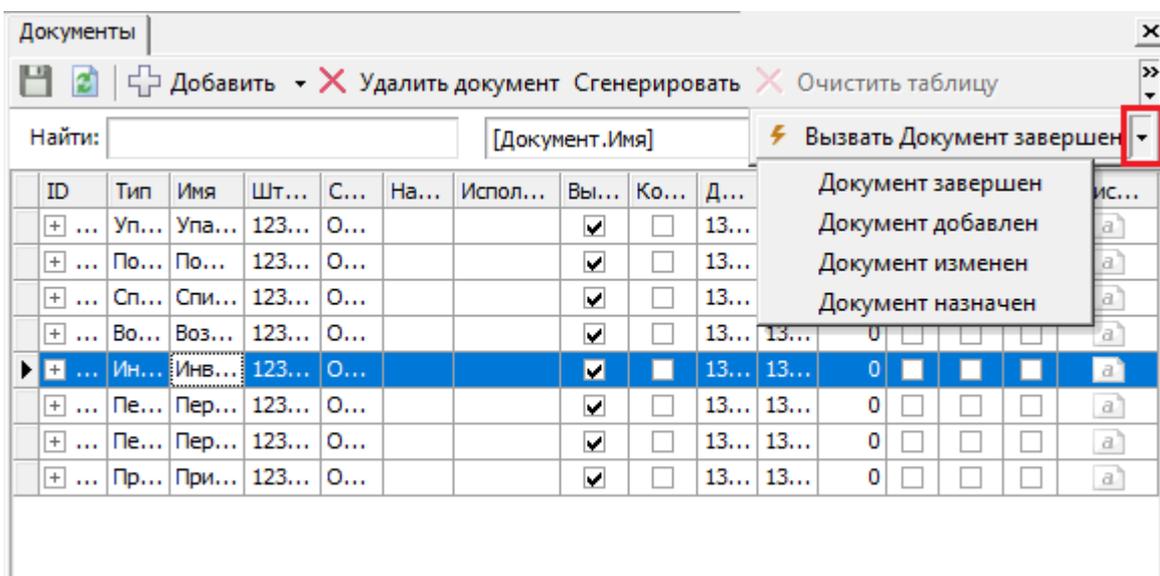
1. В **менеджере баз Mobile SMARTS** выбрать нужную базу и открыть для нее панель управления с помощью кнопки «Править конфигурацию».



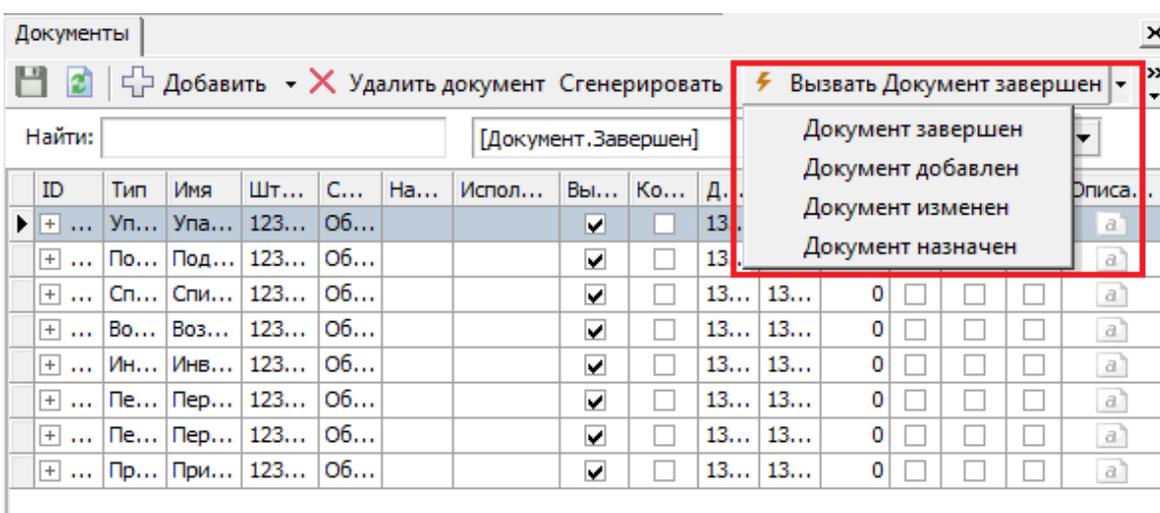
2. В панели управления открыть список документов данной базы.



3. Добавить кнопку вызова события на панель инструментов с помощью кнопки . После этого выбрать нужное событие из выпадающего списка.



4. Кнопка для вызова выбранного события появится на панели инструментов. Для того чтобы сменить вызываемое событие, выберите новое из выпадающего списка.



5. Далее для вызова события выберите нужный документ из списка, и нажмите на кнопку.

Документы

Добавить Удалить документ Сгенерировать Очистить таблицу **Вызвать Документ завершен**

Найти: [Документ.Тип]

ID	Тип	Имя	Штрихкод	Склад	Назначено	Испо...	В...	К...	...	П...	П...	...	В...	...	Описание
0a...	Упаковочн...	Упаковочный ...	123456789012346	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a
20...	Подбор зак...	Подбор заказа ...	123456789012341	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a
6a...	Списание	Списание № д...	123456789012315	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a
c04...	Возврат	Возврат № демо	123456789012314	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a
df3...	Инвента...	Инвентаризац...	123456789012341	Общий			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1...	1...	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	a
e3...	Перемещен...	Перемещение ...	123456789012342	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a
ef0...	Перемещен...	Перемещение ...	123456789012343	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a
f32...	Приход на с...	Приход на скл...	123456789012398	Общий			<input checked="" type="checkbox"/>	<input type="checkbox"/>	1...	1...	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	a

Не нашли что искали?



Задать вопрос в техническую поддержку