

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 1C, 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 1C.
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 1C на сервер Mobile SMARTS.

Не нашли что искали?



Задать вопрос в техническую поддержку