

Пример разработки коннектора к внешней системе

Для примера разработаем коннектор, который будет выполнять получение данных из файла Excel онлайн по запросам с ТСД. В примере будет рассмотрена, в том числе, перегрузка функций-обработчиков событий сервера для получения номенклатуры, документов и обработки запросов к таблицам.

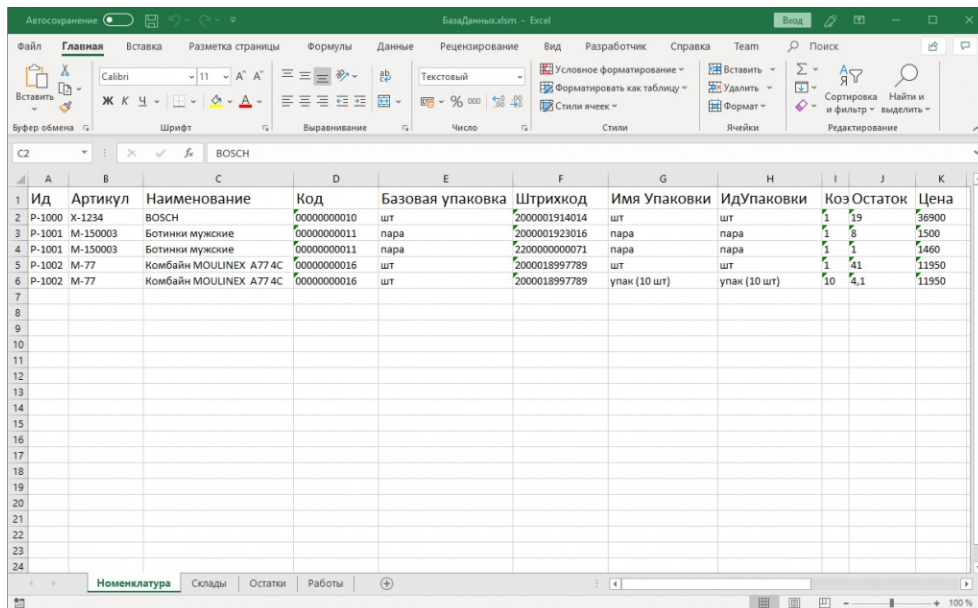
— **Готовые исходники с тестовыми данными.**

Постановка задачи

Требуется разработать коннектор, который будет по запросам с ТСД получать данные из файлов Excel. Используются следующие справочники: «Номенклатура», «Склады», «Остатки». Должна быть обеспечена работа с документами «Поступление», на ТСД должен отображаться список документов для работы, подготовленных на сервере. Пользователь ТСД выбирает документ, принимает товар, после завершения работы данные записываются в исходный документ. Используется база Mobile SMARTS «Магазин 15».

Подготовка данных

Справочники будут храниться в одном файле (книге) Excel на листах: «Номенклатура», «Склады», «Остатки». Файл со справочниками имеет фиксированное название «БазаДанных.xlsx» (книга Excel с поддержкой макросов) и располагается в заданной папке (путь к папке указывается в настройках коннектора). Файлы документов «Поступления» будут находиться в этой же папке и называться «Поступление №00001.xlsx», «Поступление №00002.xlsx» и т.д. В архиве с исходниками данные находятся в папке Xlsx, состав полей см. в файлах.



Ид	Артикул	Наименование	Код	Базовая упаковка	Штрихкод	Имя Упаковки	ИдУпаковки	Коэф	Остаток	Цена
P-1000	X-1234	BOSCH	00000000010	шт	2000001914014	шт	1	19	36900	
P-1001	M-150003	Ботинки мужские	00000000011	пара	2000001923016	пара	1	8	1500	
P-1001	M-150003	Ботинки мужские	00000000011	пара	2200000000071	пара	1	1	1460	
P-1002	M-77	Комбайн MOULINEX A77 4C	00000000016	шт	2000018997789	шт	1	41	11950	
P-1002	M-77	Комбайн MOULINEX A77 4C	00000000016	шт	2000018997789	упак (10 шт)	10	4,1	11950	

Разработка и тестирование коннектора

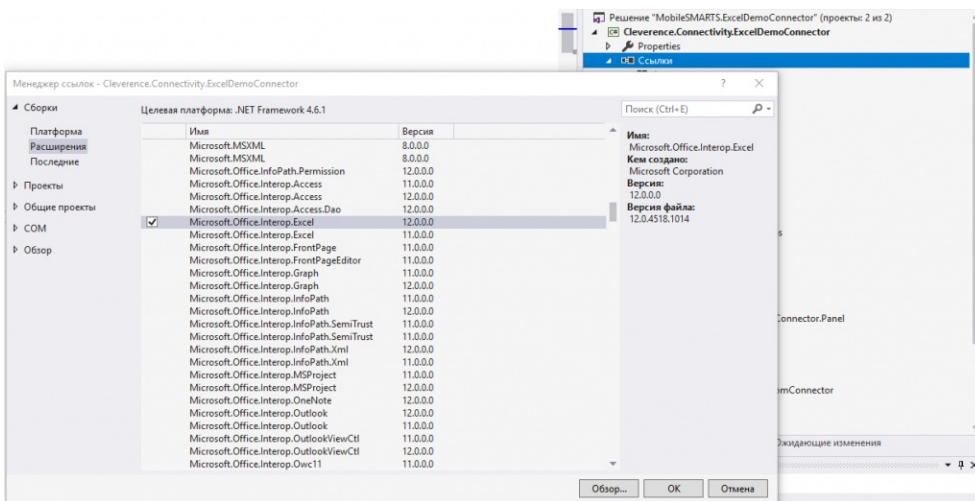
В Visual Studio создаем «Решение» (solution) и два проекта в нем:

Cleverence.Connectivity.ExcelDemoConnector (для сервера) и
Cleverence.Connectivity.ExcelDemoConnector.Panel (для панели управления).

Добавляем в ссылки (Reference assemblies) в серверный проект Cleverence.Connectivity.ExcelDemoConnector: Cleverence.Common.dll, Cleverence.Connectivity.dll, Cleverence.DataCollection.dll, Cleverence.MobileSMARTS.dll.

В проект для панели управления Cleverence.Connectivity.ExcelDemoConnector.Panel: Cleverence.DataCollection.dll, Cleverence.MobileSMARTS.ComConnector.dll.

Для работы с файлом Excel будем использовать COM-объект Excel.Application (на ПК должен быть установлен Microsoft Excel), добавим в ссылки в проект для сервера Microsoft.Office.Interop.Excel:



Добавим в каждый из проектов по файлу `ExcelDemoConnector.cs`, в котором будет находиться класс коннектора `ExcelDemoConnector`. Базовым классом для `ExcelDemoConnector` является `ConnectorTypical`.

Добавим свойство, с помощью которого можно задать путь к папке с файлами Excel, используемыми для обмена данными:

```
[C#]
```

```
public string DatabaseFolder
{
    get;
    set;
}
```

Свойство должно быть как в серверном варианте коннектора, так и в варианте для Панели управления. Других настроек для нашего коннектора не требуется, займемся разработкой серверной части, которая будет выполнять обмен данными с Excel.

Добавим приватное поле `Excel.Application excel` для хранения ссылки на COM-объект Excel. Теперь первым делом нам нужно реализовать, функцию `Initialize`, которая переводит коннектор в рабочее состояние:

```
[C#]
```

```
public override void Initialize()
{
    this.Dispose();
    if (!this.Enabled)
        throw new InvalidOperationException(this.GetType().Name + " is not enabled.");
    this.excel = new Excel.Application();
    this.excel.Visible = false;
}
```

Вызываем `Dispose`, чтобы выполнить деинициализацию, если ранее коннектор был инициализирован. Если вызовы коннектора запрещены через панель управления (`Enabled == false`), вызываем исключение. Если вызовы разрешены, создаем COM-объект.

Деинициализация:

[C#]

```
public override void Deinitialize()
{
    if (this.excel != null)
    {
        this.excel.Quit();
        this.excel = null;
    }
}
```

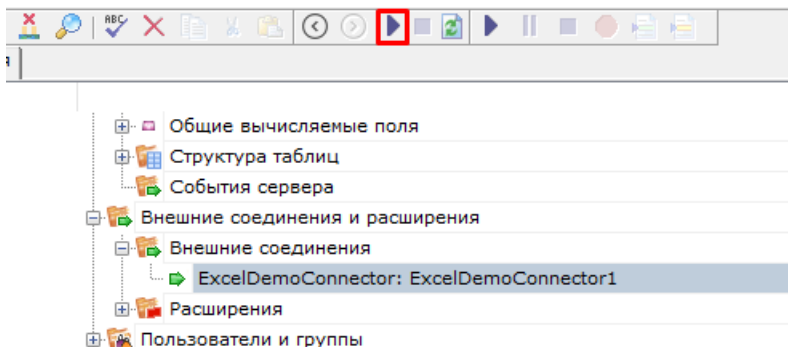
Признак того, что коннектор инициализирован:

```
public override bool Initialized
{
    get
    {
        return this.excel != null;
    }
}
```

Этого достаточно, чтобы проверить запуск коннектора через панель управления.

Соберем решение в Visual Studio. Развернем базу «Магазин 15 Расширенный» или «Полный». Скопируем dll для сервера в <Папка базы Mobile SMARTS>\Server\DataService\bin\, для панели управления в <Папка базы Mobile SMARTS>\Control panel\Addins. Запустим сервер в режиме отладки из командной строки: C:\Program Files (x86)\Cleverence Soft\Mobile SMARTS\Server\Cleverence.MobileSMARTS.Server.exe/debug.

Добавим коннектор в конфигурацию, сохраним и проверим, что выполняется запуск.



Реализуем получение номенклатуры онлайн из файла Excel. Когда товар запрашивается терминалом на сервере по каким-либо реквизитам (Ид, Штрихкод, Артикул) и товар не найден в выгруженном справочнике номенклатуры, выполняется вызов обработчика события «Получить товар»

В случае наследования от ConnectorTypical обработчик события может быть реализован как во внешней системе (вызов через InvokeMethod функции из внешней системы по имени), так непосредственно в коннекторе. Для реализации обработчика в самом коннекторе нужно перегрузить функцию GetProduct:

[C#]

```
[EventProcessor(EventType = EventType.ProductNotFound)]
public override object GetProduct(DeviceInfo di, string productId, string packingId, SearchProductMode searchMode)
{
    CheckInit();
    return GetProductInternal(productId, packingId, searchMode);
}
```

Не забываем указать атрибут EventProcessor(EventType = EventType.ProductNotFound). Реализацию поиска товаров по заданным реквизитам в зависимости от режима (searchMode) см. в GetProductInternal.

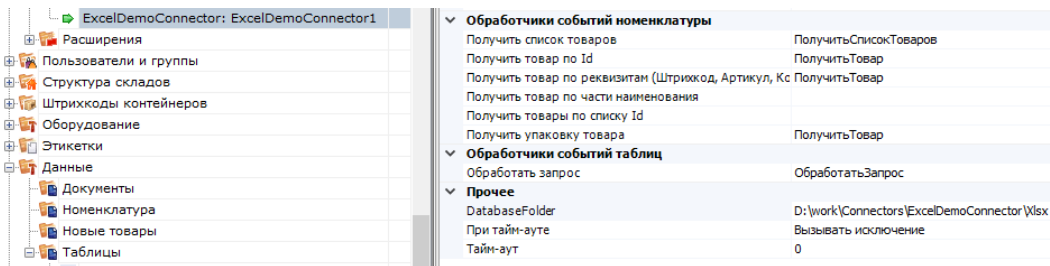
Событие «Получить список товаров» возникает, когда пользователь на ТСД заходит в список номенклатуры (выбор товара по 0 к действию «Выбор номенклатуры») и в базе нет выгруженного справочника. Перегрузим в коннекторе функцию GetProductsList:

[C#]

```
[EventProcessor(EventType = EventType.ListProducts)]
public override object GetProductsList(DeviceInfo di, string searchStr)
{
    CheckInit();
    return GetProductsListInternal(searchStr);
}
```

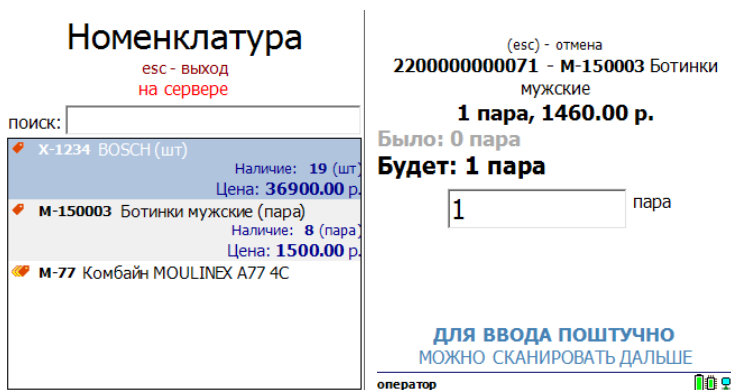
Функция возвращает PackedProductCollection со всем списком номенклатуры.

Проверим, как происходит получение номенклатуры. В настройках коннектора нужно указать путь к папке, в которой находится файл Excel со справочниками (DatabaseFolder). Нужно заполнить обработчики событий номенклатуры: «Получить список товаров», «Получить товар по Id», «Получить товар по реквизитам» (штрихкод, артикул, код), «Получить упаковку товара».



Задание имен обработчиков требуется для того, чтобы указать, что коннектор имеет подписку на определенные события. При вызове обработчика через InvokeMethod указанные имена обработчиков передаются в InvokeMethod.

Проверить получение номенклатуры можно с помощью клиента Mobile SMARTS для ПК. Для получения списка товаров заходим «Просмотр справочников -> Товары». Проверить получение по штрихкоду можно, например, в операции «Сбор штрихкодов»:



Реализуем получение списка документов. Обработчик события «Получить список документов» вызывается, когда пользователь на ТСД заходит в список выбора документов по кнопке типа документа.

[C#]

```
[DocumentEventProcessor(EventType = EventType.ListDocuments)]
public override object GetDocumentsList(DeviceInfo di, string documentTypeName)
{
    ...
}
```

Функция возвращает DocumentDescriptionCollection со списком описаний документов (DocumentDescription), готовых для отдачи на ТСД. Список получаем на основе имен файлов xlsx в папке с данными.

Для получения документа в работу на ТСД нужно реализовать обработчик события «Получить документ». Обработчик вызывается, когда пользователь на ТСД выбирает документ из списка, полученного с помощью события «Получить список документов», или сканирует штрихкод документа в окне выбора документов.

[C#]

```
[DocumentEventProcessor(EventType = EventType.GetDocument)]
public override object GetDocument(DeviceInfo di, string documentTypeName, string identity, GetDocumentMode mode)
{
    ...
}
```

Функция возвращает объект Document, если документ найден по переданным параметрам.

Проверим работу на примере документов «Поступление»:

Номенклатура	План	Факт
X-1234 BOSCH (шт)	3	0
M-150003 Ботинки мужские (пара)	6	0
M-77 Комбайн MOULINEX A77 4C (шт)	10	0

Для загрузки завершено на ТСД документа используем событие «Документ завершен».

[C#]

```
[DocumentEventProcessor(EventType = EventType.DocumentFinished)]
public override object DocumentFinished(DeviceInfo di, string documentTypeName, Document doc)
{
    CheckInit();
    string fileName = GetFileNameByDocId(doc.Id, documentTypeName);
    if (!string.IsNullOrEmpty(fileName))
    {
        if (LoadDocumentFromTerminal(fileName, doc))
        {
            MessageCenter.AddNewMessage(string.Format("Документ '{0}' загружен.", doc.Name), null, doc.UserId, false);
            BaseRuntimeContext.Current.DocumentsManager.Delete(doc);
            return doc;
        }
    }
    return base.DocumentFinished(di, documentTypeName, doc);
}
```

В данной функции после загрузки документа на ТСД отправляется сообщение с помощью MessageCenter.AddNewMessage. Завершенный документ удаляется из базы Mobile SMARTS с помощью BaseRuntimeContext.Current.DocumentsManager.Delete(doc).

Реализуем получение данных онлайн при запросах к таблицам Mobile SMARTS. Например, в конфигурации «Магазин 15» есть таблицы «Склады», «Остатки» и др. На ТСД может выполняться получение списка всех складов, запрос остатков определенного товара и др. Если таблица хранится на сервере, в настройках таблицы включен поиск во внешней системе и задан обработчик события «Обработать запрос», то сервер вызывает указанный обработчик. Перегрузим в нашем коннекторе функцию ProcessTableRequest.

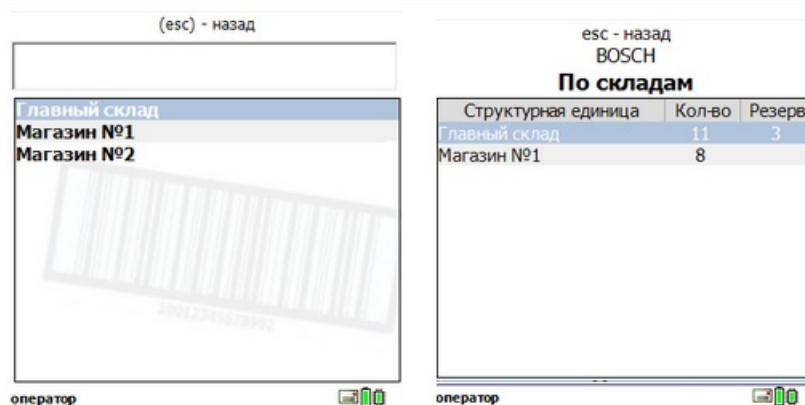
[C#]

```
[EventProcessor(EventType = EventType.TableRequest)]
public override object ProcessTableRequest(DeviceInfo di, DocumentQuery dq, DocumentTableInfo tableInfo)
{
    ...
}
```

В функцию передается: DocumentQuery dq - объект запроса, в свойстве WhereRootElement содержится корень синтаксического дерева запроса. Обойдя дерево, можно сформировать запрос в том виде, который ожидает внешняя система. DocumentTableInfo tableInfo - описание таблицы, в свойстве Name содержится имя таблицы из конфигурации Mobile SMARTS. В простейшем случае можно возвращать все строки запрошенной таблицы, не накладывая условие из DocumentQuery, сервер Mobile SMARTS сам выполнит выборку данных по заданным условиям. Однако, в случае реальной работы с большими объемами данных, так делать не рекомендуется.

Функция возвращает коллекцию строк RowCollection.

Проверим получение складов и остатков:



Кроме обработки событий сервера, есть возможность вызова произвольных функций внешней системы, которые будут возвращать некоторые данные на ТСД или выполнять какую-то работу во внешней системе. В конфигурации Mobile SMARTS для этого используется действие «**Вызов внешней системы**». В коннекторе должна быть реализована функция InvokeMethod.

В нашем случае InvokeMethod будет вызывать макрос Excel с указанным именем, передавая полученные с ТСД аргументы:

[C#]

```
public override object InvokeMethod(string methodName, object[] args)
{
    CheckInit();
    string dbPath = Path.Combine(DatabaseFolder, DbFileName);
    Excel.Workbook workbook = this.excel.Workbooks.Open(dbPath);
    try
    {
        return RunMacro(methodName, args);
    }
    finally
    {
        workbook.Close();
    }
}
```

Добавим в книгу Excel лист «Работы», на котором будет таблица с количеством собранных сотрудниками заказов за неделю:

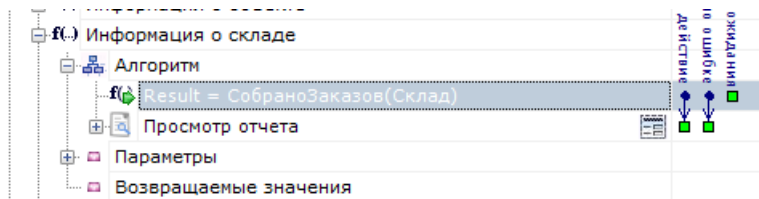
	A	B	C
1	Комплектовщик	Склад	Собрано заказов
2	Фахуртдинов	Главный склад	352
3	Петров	Главный склад	278
4	Коршунов	Магазин №1	94
5	Мухамедов	Магазин №2	65

На VB в Excel напишем функцию, которая будет возвращать таблицу с колонками «Комплектовщик», «Собрано заказов» по переданному складу.

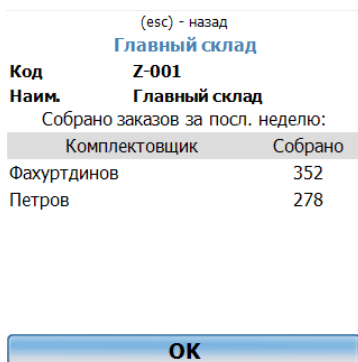
```
[C#]

Function СобраноЗаказов(Склад As String)
Dim sh_src As Worksheet
Dim storageConnector As Object, rowCollection As Object, row As Object
Dim picker As String
Dim ordCnt As Integer
Dim rng As Excel.Range
Set storageConnector = CreateObject("Cleverence.Warehouse.StorageConnector")
Set rowCollection = CreateObject("Cleverence.Warehouse.RowCollection")
Set sh_src = Worksheets("Работы")
.....
СобраноЗаказов = storageConnector.ToXml(rowCollection)
End Function
```

Функция возвращает объект Cleverence.Warehouse.RowCollection, сериализованный в xml. В конфигурации Mobile SMARTS сделаем вызов данной функции при просмотре информации о складе:



В действии «Просмотр отчета» выведем полученную таблицу:



Была ли статья полезна?

Нет

Да