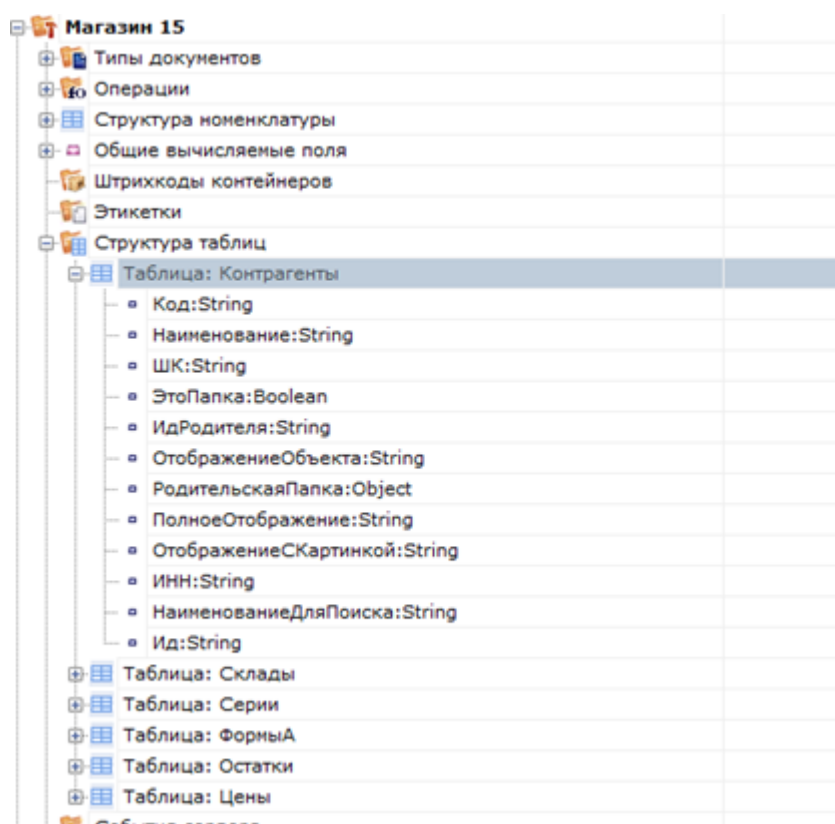


Дополнительные таблицы Mobile SMARTS

Последние изменения: 2024-03-26

Дополнительная таблица представляет собой “плоский” набор данных, строки таблицы содержат поля простых типов (числа, строки, булевы). Структура данных дополнительных таблиц создается и редактируется с помощью [Панели управления](#) Mobile SMARTS. Разработчик конфигурации может создавать произвольные дополнительные таблицы. В таблице кроме полей, которые содержат данные, могут быть заданы вычисляемые поля (значения вычисляемых полей определяется путем операций над значениями других полей), для вычисляемых полей задается Шаблон значения. Таблица может как загружаться на терминал, так и храниться на сервере (в случае серверной базы Mobile SMARTS).

Например, таблица Контрагенты используется для хранения списка сторонних организаций, на терминале при работе с документом Приёмки пользователь может выбирать организацию, от которой пришел товар. В таблице Остатки хранятся остатки товаров в разрезе складов, привязка к номенклатуре выполняется с помощью поля ИдНоменклатуры.



Дополнительные таблицы

Для работы с дополнительными таблицами используется объект [TableAccessor \[ДоступКТаблице\]](#), с помощью функций этого объекта можно выгружать данные в таблицу, читать данные, добавлять и удалять строки таблицы. Для получения объекта [TableAccessor \[ДоступКТаблице\]](#) следует использовать функцию [GetTableAccessor \[ПолучитьДоступКТаблице\]](#) из [StorageConnector \[Соединение\]](#). Перед вызовом [GetTableAccessor \[ПолучитьДоступКТаблице\]](#) должно быть установлено подключение к базе данных Mobile SMARTS с помощью [SelectCurrentApp \[УстановитьПодключениеСБазойСМАРТС\]](#).

Наименование
Параметры и возвращаемое значение
Описание

GetTableAccessor
[ПолучитьДоступКТаблице]
string tableName
Имя таблицы, для которой нужно получить объект доступа. Должно соответствовать имени одной из таблиц из конфигурации Mobile SMARTS.
Возвращаемое значение: объект TableAccessor [ДоступКТаблице] .
Возвращает объект TableAccessor [ДоступКТаблице] , позволяющий выгружать данные в таблицу, читать данные, добавлять и удалять строки таблицы.

Объект [TableAccessor \[ДоступКТаблице\]](#) имеет следующие функции:

Наименование
Параметры и возвращаемое значение
Описание
Выгрузка
BeginUpload [НачатьВыгрузку]
bool overwrite
Истина - полностью перезаписать таблицу при выгрузке, Ложь - добавлять строки
Возвращаемое значение: нет
Открывает выгрузку данных в таблицу.
Upload [ДобавитьВВыгрузку]
RowCollection rows
rows - коллекция выгружаемых строк
Возвращаемое значение: нет
Добавляет порцию строк в выгрузку.

EndUpload [ЗавершитьВыгрузку]
Параметры: нет Возвращаемое значение: нет
Завершает выгрузку. Данные сохраняются в базу Mobile SMARTS.
ResetUpload [СброситьВыгрузку]
Параметры: нет Возвращаемое значение: нет
Сбрасывает начатую выгрузку без сохранения данных.
Чтение данных Свойства:
Count [Количество]
Возвращаемое значение: Число
Количество строк в таблице.
Item [Элемент]
int index
index - индекс строки Возвращаемое значение: нет
Получает строку по индексу.
Поиск
FindByField
[НайтиПоЗначениюПоля]
string fieldName, object value
fieldName
Наименование поля для отбора.
value
Значение для отбора
Возвращаемое значение:
TableAccessor [ДоступКТаблице]
Выполняет поиск в таблице по значению заданного поля.
Возвращает объект
TableAccessor [ДоступКТаблице], из которого можно читать строки, удовлетворяющие отбору.

FindFirstByField

[НайтиПервуюЗаписьПоЗначениюПоля]

string fieldName, object value

fieldName

Наименование поля для отбора.

value

Значение для отбора

Возвращаемое значение: [Row](#) Найденная строка таблицы или null.

Выполняет поиск в таблице по значению заданного поля и возвращает первую найденную строку или null, если строка не найдена.

Редактирование

Add [Добавить]

[Row](#) item

item - добавляемая строка Возвращаемое значение: нет

Добавляет строку в таблицу. Чтобы зафиксировать сделанные изменения, следует вызвать [CommitChanges](#)

[\[СохранитьИзменения\]](#).

Remove [Удалить]

[Row](#) item

Удаляет строку из таблицы.

item - строка таблицы для удаления

Возвращаемое значение: bool

Чтобы зафиксировать сделанные изменения, следует вызвать [CommitChanges](#)

[\[СохранитьИзменения\]](#). .

CommitChanges [\[СохранитьИзменения\]](#)

Параметры: нет Возвращаемое значение: нет

Сохраняет сделанные изменения в таблицу в базе данных Mobile SMARTS.

UndoChanges [ОтменитьИзменения]

Параметры: нет Возвращаемое значение: нет

Отменяет сделанные изменения (добавленные, удаленные строки).

Пример выгрузки

C#:

```
var connector = new Cleverence.Warehouse.StorageConnector(); // Создаем экземпляр объекта
соединения

// Выполняем подключение к базе Mobile SMARTS

connector.SelectCurrentApp("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-c7c7003887c5");

// Получаем объект для доступа к таблице

var tableContractors = connector.GetTableAccessor("Контрагенты");

try

{

tableContractors.BeginUpload(true); // Начинаем выгрузку

var rows = new Cleverence.Warehouse.RowCollection(); // Коллекция строк таблицы, которую
будем

// использовать при выгрузке

using (SqlConnection connection = new SqlConnection(connectionString)) //Устанавливаем
соединение с //БД, из которой будем читать данные

{

connection.Open(); command.Connection = connection;

SqlDataReader reader = command.ExecuteReader(); while (reader.Read()) //Читаем данные

{

string id = reader.GetString(0); string name = reader.GetString(1); string inn =
reader.GetString(1);

if(rows.Count >= 1000)

{

//Если набрали достаточно строк, выгружаем порцию строк и создаем новый объект
```

коллекции,

```
//чтобы набирать строки дальше в пустую коллекцию. tableContractors.Upload(rows);

rows = new Cleverence.Warehouse.RowCollection();

}

var row = new Cleverence.Warehouse.Row(); //Создаем строку таблицы row.SetField("Ид", id);
//Проставляем поля row.SetField("Наименование", name);

row.SetField("ИНН", inn);

rows.Add(row); //Добавляем строку в коллекцию строк

}

//Если есть строки для выгрузки, выгружаем
} if(rows.Count > 0)

tableContractors.Upload(rows);

}

tableContractors.EndUpload(); //Завершаем выгрузку

}

catch

{

//Обработка ошибки

tableContractors.ResetUpload(); //Если возникла ошибка, сбрасываем выгрузку
```

«1С:Предприятие 8»:

```
connector = новый СОМОбъект("Cleverence.Warehouse.StorageConnector"); // Создаем
экземпляр объекта

//соединения

connector.УстановитьПодключениеСБазойСМАРТС("михаил-пк:10501/d7c15f54-fd05-4811-8f6d-
с7с7003887с5");

//Выполняем подключение

таблицаКонтрагенты = connector.ПолучитьДоступКТаблице("Контрагенты"); // Получаем
объект для доступа к // таблице
```

Попытка

таблицаКонтрагенты.НачатьВыгрузку(Истина); // Начинаем выгрузку

строкиДляВыгрузки = новый СОМОбъект("Cleverence.Warehouse.RowCollection"); //
Коллекция строк //таблицы, которую будем использовать при выгрузке

Запрос = Новый Запрос(ТекстЗапроса);

ТаблицаРезультата = Запрос.Выполнить().Выгрузить(); //Выполняем запрос

Для каждого СтрокаТаблицы из ТаблицаРезультата Цикл

Ид = XMLСтрока(СтрокаТаблицы.КонтрагентСсылка); //Получаем необходимые поля
Наименование = СтрокаТаблицы.Наименование;

ИНН = СтрокаТаблицы.ИНН;

Если строкиДляВыгрузки.Количество >= 1000 Тогда

//Если набрали достаточно строк, выгружаем порцию и очищаем коллекцию строк
таблицаКонтрагенты.ДобавитьВВыгрузку(строкиДляВыгрузки);
таблицаКонтрагенты.Очистить();

КонецЕсли;

строкаДляВыгрузки = новый СОМОбъект("Cleverence.Warehouse.Row"); //Создаем строку
таблицы строкаДляВыгрузки.УстановитьПоле("Ид", Ид); //Проставляем поля
строкаДляВыгрузки.УстановитьПоле("Наименование", Наименование);
строкаДляВыгрузки.УстановитьПоле("ИНН", ИНН);

строкиДляВыгрузки.Добавить(строкаДляВыгрузки); //Добавляем строку в коллекцию
строк КонецЦикла;

//Если есть строки для выгрузки, выгружаем Если строкиДляВыгрузки.Количество > 0
Тогда

таблицаКонтрагенты.ДобавитьВВыгрузку(строкиДляВыгрузки); КонецЕсли;

таблицаКонтрагенты.ЗавершитьВыгрузку(); //Завершаем выгрузку Исключение

// Обработка ошибки

таблицаКонтрагенты.СброситьВыгрузку(); //Если возникла ошибка, сбрасываем выгрузку
КонецПопытки;



Не нашли что искали?



Задать вопрос в техническую поддержку