

# Общие принципы работы механизма синхронизации Sync Engine

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

<b>Общие принципы работы механизма синхронизации Sync Engine</b>	<b>5</b>
Синхронизация с внешними хранилищами данных	5
Работа с метаданными для синхронизации	7
<b>Класс SyncContext</b>	<b>8</b>
Конструкторы	8
Свойства	9
Методы	9
<b>Класс RemoteProvider</b>	<b>10</b>
Свойства	11
Методы	11
<b>Интерфейс IRemoteItem</b>	<b>12</b>
Методы	12
Атрибут Map	13
<b>Класс LocalProvider</b>	<b>14</b>
Конструкторы	15
Свойства	15
Методы	15
<b>Класс SyncEntity</b>	<b>15</b>
Свойства	16
<b>Класс SyncItemSchema</b>	<b>16</b>
Свойства	16
Методы	17
<b>Класс EntityConfig</b>	<b>17</b>
Свойства	17
<b>Класс DetailEntityConfig</b>	<b>18</b>
Свойства	18
<b>Класс LocalItem</b>	<b>18</b>
Свойства	19
Методы	19
<b>Класс SysSyncMetaData</b>	<b>19</b>
Свойства	19
<b>Класс MetaDataStore</b>	<b>21</b>
Методы	21
<b>Класс ItemMetaData</b>	<b>21</b>
Свойства	21
<b>Интерфейс IReplicaMetadata</b>	<b>21</b>

Свойства	22
Методы	22

# Общие принципы работы механизма синхронизации Sync Engine



## Синхронизация с внешними хранилищами данных

В Creatio реализован механизм синхронизации с внешними хранилищами данных (Sync Engine) который позволяет создавать, изменять и удалять `Entity` в системе на основании данных из внешних систем и экспортировать данные во внешние системы.

Процесс синхронизации осуществляется с помощью класса `SyncAgent`, который реализован в пространстве имен `Terrasoft.Sync` ядра приложения.

## Классы, задействованные в механизме синхронизации

- Агент синхронизации (`SyncAgent`) — класс с одним публичным методом `Synchronize`, который запускает синхронизацию между переданными ему хранилищами данных.
- Контекст синхронизации (`SyncContext`) — класс, представляющий собой агрегацию провайдеров и метаданных для работы `SyncAgent`.
- Хранилище — конкретный репозиторий синхронизируемых данных.
  - Локальное хранилище (`LocalProvider`) — позволяет работать с `LocalItem` в Creatio.
  - Внешнее хранилище (`RemoteProvider`) — внешний сервис или приложение, данные из которого синхронизируются с Creatio.
- Элемент синхронизации (`SyncItem`) — множество объектов из внешнего и локального хранилища, между которыми устанавливается однозначное соответствие.
  - Элемент синхронизации внешнего хранилища (`RemoteItem`) — представляет набор данных из внешнего хранилища, который синхронизируется атомарно. Может состоять из одной или нескольких сущностей (записей) внешнего хранилища.
  - Элемент синхронизации (`SyncEntity`) — является оберткой над конкретными `Entity`. `SyncEntity` необходим для работы с `Entity` как с синхронизируемым объектом, так и с состоянием и действием, которое с этим `Entity` необходимо произвести (добавить, удалить, изменить).
  - Элемент синхронизации (`LocalItem`) — один или несколько объектов из Creatio, которые синхронизируются с внешним хранилищем как одно целое. Элемент синхронизации из внешнего хранилища, конвертированный в сущности `LocalItem`, в свою очередь, содержит набор экземпляров класса `SyncEntity`.
- Таблица метаданных [`SysSyncMetadata`] — содержит служебную информацию синхронизируемых элементов, по сути является таблицей развязки `RemoteItem` - `LocalItem`. Описание метаданных можно посмотреть в [статье](#).

## Общий алгоритм синхронизации

Перед началом синхронизации необходимо создать экземпляр `SyncAgent` и контекст синхронизации `SyncContext`, после этого актуализировать записи в таблице метаданных данными из Creatio. Для этого нужно вызвать метод `CollectChangesInSyncedEntities` класса, реализующего интерфейс `IReplicaMetadata`.

Общий алгоритм актуализации записей метаданных следующий:

1. Если какая-либо ранее синхронизированная сущность в Creatio была изменена с момента последней синхронизации, значит у соответствующей записи в метаданных изменяется дата модификации, свойству `LocalState` устанавливается значение "Изменен", а в качестве источника модификации устанавливается идентификатор `LocalStore`.
2. Если синхронизированная сущность в Creatio была удалена после последней синхронизации — у соответствующей записи в метаданных устанавливается `LocalState` "Удален".
3. Если для сущности в Creatio нет соответствующей записи в метаданных — она игнорируется.

Далее начинается процесс синхронизации хранилищ:

1. Поочередно запрашиваются все изменения из внешнего хранилища.
2. Для каждого элемента внешнего хранилища необходимо получить метаданные.
  - a. Если метаданные получить не удалось — это новый элемент, который конвертируется в элемент Creatio и сохраняется. Для заполнения объекта синхронизации, в приложении вызывается метод `FillLocalItem` у конкретного экземпляра `RemoteItem`. Также сохраняется запись в метаданных (`Id` внешнего хранилища, `Id` элемента во внешнем хранилище, дата создания и модификации устанавливается текущей, источник создания и модификации — внешнее хранилище).
  - b. Если получены метаданные, значит, этот элемент уже был синхронизирован с Creatio. Необходимо перейти к решению конфликта версий. По умолчанию приоритет имеют последние изменения в приложении или внешнем хранилище (реализация в `RemoteProvider`).
  - c. Актуализируются метаданные для текущей пары элементов синхронизации.

После перебора всех измененных элементов из внешнего хранилища, в метаданных (в интервале между прошлой и текущей синхронизациями) остались элементы, которые были изменены в Creatio, но не были изменены во внешнем хранилище.

1. Необходимо получить измененные в Creatio элементы в промежутке между прошлой синхронизацией и началом текущей синхронизации.
2. Применяются изменения во внешнем хранилище.
3. Необходимо обновить дату модификации элементов в метаданных (источник изменения Creatio).

После этого необходимо новые, еще не синхронизированные записи из Creatio добавить во внешнее хранилище, а также добавить метаданные для новых элементов.

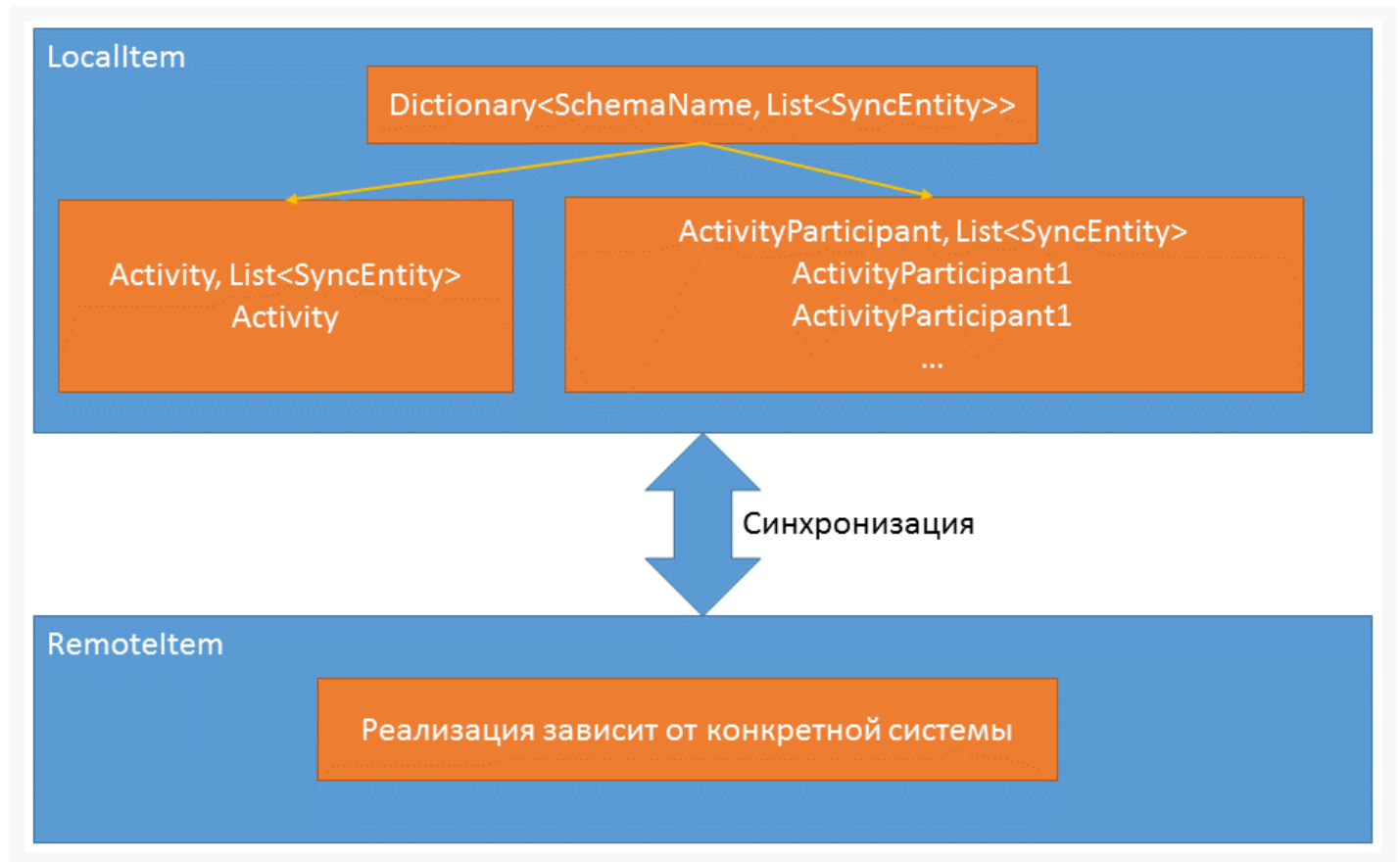
## Описание синхронизации

Активность и ее участники синхронизируются в одну задачу Google-календаря. Активность (`Activity`) и ее участники (`SyncEntity`) являются одним элементом синхронизации — `LocalItem`.

`RemoteItem` — это задача Google, которую получили извне Creatio. `LocalItem` — это набор объектов (

SyncEntity ), в которые в итоге конвертируется задача Google.

Схема синхронизации:



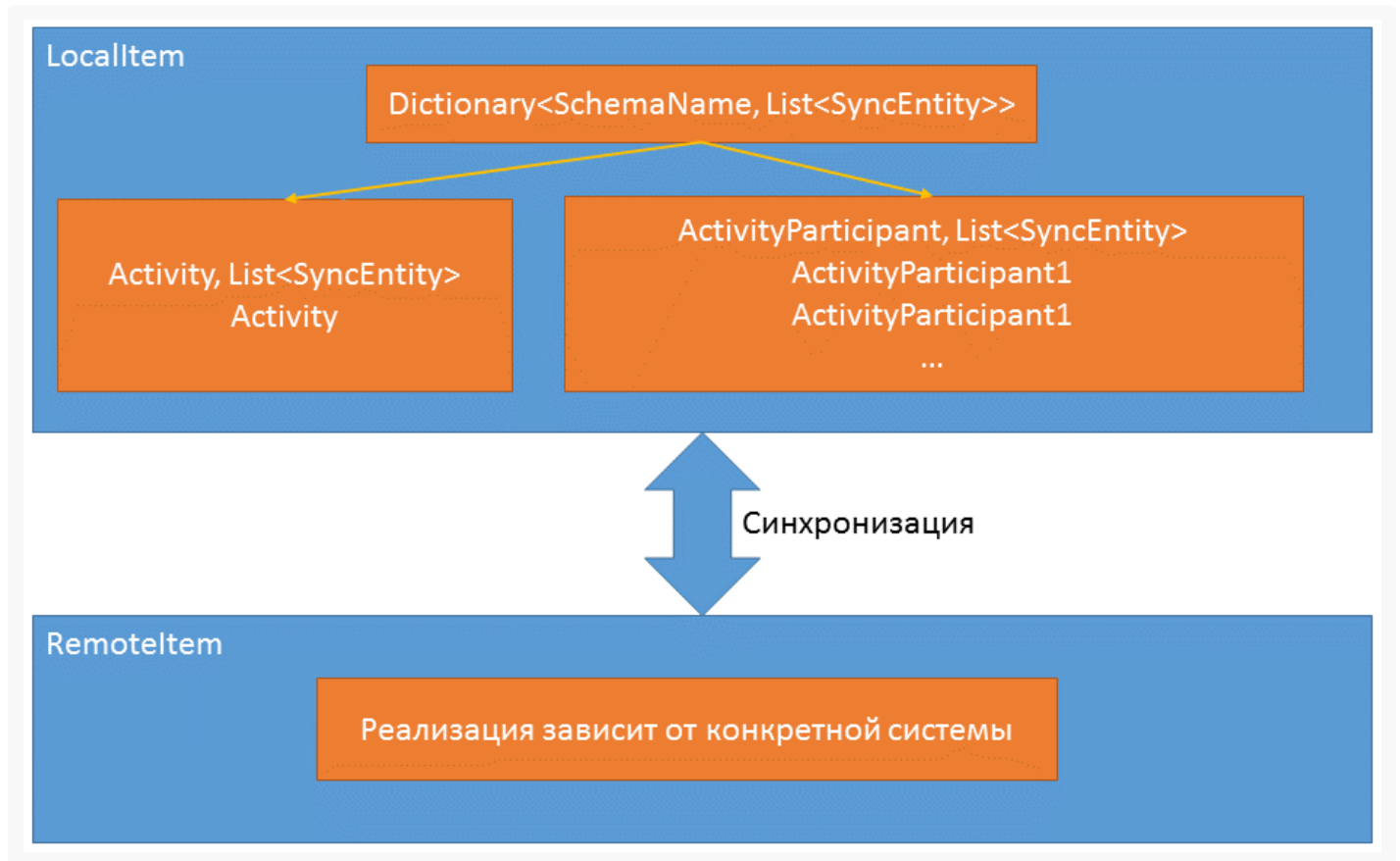
## Работа с метаданными для синхронизации

В метаданных хранится информация только об уже синхронизированных элементах.

Для одного элемента синхронизации может быть несколько записей в таблице метаданных — по одной для каждого объекта приложения, включенного в элемент синхронизации.

Активность и участники — это один элемент синхронизации, но в метаданных будет содержаться одна запись для активности и по одной записи для каждого участника.

На текущий момент только один объект из внешнего хранилища преобразуется в несколько объектов Creatio, как показано на рисунке. Вариант, когда допустимо соответствие многие ко многим (несколько внешних объектов соответствуют нескольким локальным объектам, и эта совокупность является одним элементом синхронизации), не поддерживается.



В системе метаданные для одного элемента синхронизации представлены объектом класса `ItemMetadata` (коллекция элементов `[SysSyncMetaData]`). А управление метаданными осуществляется через класс, реализующий интерфейс `IReplicaMetadata`. Экземпляр класса, реализующий интерфейс `IReplicaMetadata`, создается через класс-фабрику `MetadataStore` для конкретного хранилища.

## Класс SyncContext C#



Сложный

Пространство имен `Terrasoft.Sync`.

Класс, представляющий собой агрегацию провайдеров и метаданных для работы SyncAgent.

**Note.** Полный перечень методов и свойств класса `SyncContext`, его родительских классов, а также реализуемых им интерфейсов можно найти в документации "[.NET библиотеки классов ядра платформы](#)".

## Конструкторы

`SyncContext()`

Создает экземпляр класса.



## Свойства

`Logger` `ISyncLogger`

Объект, который позволяет сохранять сообщения в лог интеграции.

`LocalProvider` `LocalProvider`

Позволяет работать с `LocalItem`.

`RemoteProvider` `RemoteProvider`

Внешний сервис или приложение, данные из которого синхронизируются с Creatio.

`ReplicaMetadata` `IReplicaMetadata`

Реализует работу с метаданными.

`LastSyncVersion` `DateTime`

Дата и время последней синхронизации в UTC.

`CurrentSyncStartVersion` `DateTime`

Текущие дата и время синхронизации в UTC. Устанавливается после обновления метаданных.

## Методы

`void LogError(SyncAction operation, SyncDirection direction, System.string format, params System`

`void LogError(SyncAction operation, SyncDirection direction, System.string format, System.Exception`

Записывает в журнал сообщение об ошибке.

[Параметры](#)

operation	Записывает действие для синхронизации объекта.
direction	Направление синхронизации.
format	Формат.
exception	Исключение, вызвавшее ошибку.
args	Формат параметров.

```
void LogInfo(SyncAction operation, SyncDirection direction, System.string format, params System.
```

Записывает в журнал информационное сообщение.

#### Параметры

operation	Записывать действие синхронизации объекта.
direction	Направление синхронизации.
format	Формат.
args	Формат параметров.

```
void SetLookupColumnValue(Entity entity, System.string lookupColumnName, System.string lookupDis
```

```
void SetLookupColumnValue(Entity entity, System.string lookupColumnName, System.string lookupDis
```

Устанавливает значение столбца поиска объекта.

#### Параметры

entity	Экземпляр класса <code>Terrasoft.Core.Entities.Entity</code> .
lookupColumnName	Имя справочного столбца.
lookupDisplayValue	Отображаемое значение справочного столбца.
lookupDisplayColumnName	Отображаемое имя справочного столбца.

## Класс RemoteProvider



Сложный

Внешнее хранилище (`RemoteProvider`) — инкапсулирует работу с данными из внешнего хранилища.

`RemoteProvider` — базовый класс, реализующий работу с внешним хранилищем. По сути, единая точка работы с внешней системой. На данный момент его реализация дает много свободы в реализации внешнего провайдера.

## Свойства

---

`StoreId` `Guid`

Идентификатор внешнего хранилища, с которым будет происходить синхронизация.

---

`Version` `DateTime`

Дата и время последней синхронизации.

---

`SyncItemSchemaCollection` `List<SyncItemSchema>`

Схемы элементов внешнего хранилища.

---

`RemoteChangesCount` `Int`

Количество элементов, обработанных из внешнего хранилища.

---

`LocalChangesCount` `Int`

Количество элементов, обработанных из локального хранилища.

---

## Методы

---

`KnownTypes()` `IEnumerable<Type>`

Возвращает коллекцию всех типов, реализующих интерфейс `IRemoteItem`. `SyncAgent` строит на их основе экземпляры `SyncItemSchema`, которые описывают сущности, участвующие в синхронизации.

---

`ApplyChanges(SyncContext context, IRemoteItem synItem)` `Void`

Применяет изменения в элементе внешнего хранилища.

---

`CommitChanges(SyncContext context)` `Void`

Вызывается после обработки изменений во внешнем и локальном хранилище. Предназначен для реализации необходимых дополнительных действий для конкретной реализации интеграции.

---

```
EnumerateChanges(SyncContext context) IEnumerable<IRemoteItem>
```

Возвращает перечисление новых и измененных элементов внешнего хранилища.

```
LoadSyncItem(SyncItemSchema schema, string id) IRemoteItem
```

Заполняет экземпляр `IRemoteItem` данными из внешнего хранилища.

```
CreateNewSyncItem(SyncItemSchema schema) IRemoteItem
```

Создает новый экземпляр `IRemoteItem`.

```
CollectNewItems(SyncContext context) IEnumerable<LocalItem>
```

Возвращает перечисление новых сущностей Creatio, которые будут синхронизированы во внешнее хранилище.

```
ResolveConflict(IRemoveItem syncItem, ItemMetadata itemMetaData, Guid localStoreId) SyncConflict
```

Разрешает конфликт между измененными элементами локального и внешнего хранилища. По умолчанию (реализация в `RemoteProvider`) приоритет имеют изменения в Creatio.

```
NeedMetaDataAdapterlization() Boolean
```

Возвращает признак необходимости актуализации метаданных до начала синхронизации.

```
GetLocallyModifiedItemsMetadata(SyncContext context, EntitySchemaQuery modifiedItemsEsq) IEnumer
```

Возвращает элементы синхронизации, измененные в локальном хранилище после последней синхронизации.

## Интерфейс IRemoteItem C#



Класс, реализующий интерфейс `IRemoteItem`, является неделимой единицей синхронизации и представляет собой один элемент синхронизации из внешнего хранилища данных. Класс одновременно является контейнером для данных, приходящих из внешней системы, и сам знает, как преобразовывать эти данные в сущности `Entity`, и наоборот — из `Entity` наполнить себя. Интерфейс содержит два метода, `FillLocalItem` и `FillRemoteItem` для конвертации внешнего элемента синхронизации (`RemoteItem`, то есть себя) в `LocalItem` и наоборот.

## Методы

```
FillLocalItem(SyncContext context, ref LocalItem localItem) Void
```

Заполняет свойства элемента локального хранилища `LocalItem` значениями элемента внешнего хранилища. Используется для применения изменений в локальном хранилище.

```
FillRemoteItem(SyncContext context, ref LocalItem localItem) Void
```

Заполняет свойства элемента внешнего хранилища значениями из элемента локального хранилища `LocalItem`. Используется для применения изменений во внешнем хранилище.

## Атрибут Map C#

Пространство имен `Terrasoft.Sync`.

Атрибутом `Map` декорируются реализации интерфейса `IRemoteItem`.

Основным параметром атрибута является `SchemaName`. Это название той `EntitySchema`, которая участвует в текущем элементе синхронизации.

### Использование атрибута `Map`

```
[Map("Activity", 0)]
[Map("ActivityParticipant", 1)]
public class GoogleTask: IRemoteItem {
    ...
}
```

В таком объявлении класса задача из календаря Google будет синхронизироваться с активностью и участниками активности из Creatio.

Второй параметр `Order` определяет в каком порядке `Entity` будут сохранены в локальном хранилище. `Activity` указывается первой, т.к. `ActivityParticipant` хранит ссылку на создаваемую активность и при другом порядке будет вызвано исключение.

В большинстве случаев `SyncAgent` может автоматически сформировать запрос по выборке новых элементов синхронизации из Creatio. Для этого необходимо указать основную сущность и способ связи с деталями:

### Пример указания основной сущности и способа связи с деталями

```
[Map("Activity", 0, IsPrimarySchema = true)]
[Map("ActivityParticipant", 1, PrimarySchemaName = "Activity", ForeignKeyName = "Activity")
public class GoogleTask: IRemoteItem {
    ...
}
```

В данном случае в БД отправится один запрос на получение новых активностей и по одному запросу на каждую выбранную активность для получения связанных с ней участников. Список свойств атрибута

представлен в таблице.

## Свойства

SchemaName `String`

Название схемы объекта.

Order `Int`

Порядок обработки сущности для элемента синхронизации.

IsPrimarySchema `Boolean`

Флаг, указывающий, что данная схема является основной в данном элементе синхронизации. Может быть установлен только у одной схемы.

PrimarySchemaName `String`

Имя схемы главного объекта. Не может указываться в паре с `IsPrimarySchema`.

ForeignKeyColumnName `String`

Имя колонки для связи детали с главным объектом. Не может указываться в паре с `IsPrimarySchema`.

Direction `SyncDirection`

Определяет направление синхронизации для объектов текущего типа. Значение по умолчанию — `DownloadAndUpload`.

Если не содержит флаг `Download` — изменения не будут применяться в Creatio.

Если не содержит флаг `Upload` — не будут выбираться новые сущности из Creatio.

FetchColumnNames `String[]`

Названия колонок, которые будут загружаться из локального хранилища.

## Класс LocalProvider C#

 Сложный

Пространство имен `Terrasoft.Sync`.

Локальное хранилище (`LocalProvider`) — инкапсулирует работу с данными во внутреннем хранилище

(Creatio).

`LocalProvider` — базовый класс, реализующий работу с локальным хранилищем. Позволяет работать с `LocalItem` в Creatio. Методы данного класса являются неизменными, а их перечень представлен в таблице.

## Конструкторы

```
public LocalProvider(UserConnection userConnection)
```

Создает экземпляр класса с помощью `UserConnection`.

## Свойства

```
StoreId Guid
```

ID локального хранилища, которое будет синхронизировано.

```
MaxItemsPerSelect int
```

Максимальное количество элементов в выбранном количестве.

```
UserConnection UserConnection
```

Экземпляр `UserConnection`.

## Методы

```
AddItemSchemaColumns(EntitySchemaQuery esqForFetching, EntityConfig entityConfig) Void
```

Добавляет в `EntitySchemaQuery` колонки, указанные в `EntityConfig`.

```
ApplyChanges(SyncContext context, LocalItem entities) Void
```

Применяет изменения к каждому `SyncEntity` в `LocalItem`.

```
FetchItem(ItemMetadata itemMetaData, SyncItemSchema itemSchema, bool loadAllColumns = false) Loc
```

Загружает коллекцию сущностей, связанных с конкретным элементом синхронизации.

## Класс SyncEntity



Сложный

Пространство имен `Terrasoft.Sync`.

Класс `SyncEntity` инкапсулирует экземпляр `Entity` и все необходимые для выполнения действий синхронизации этого экземпляра свойства.

## Свойства

`EntitySchemaName` `String`

Название схемы, для которой создается обертка.

`Entity` `Entity`

`Entity`, для которой создается обертка.

`State` `SyncState`

Последнее действие, произведенное над `entity` (0 — не изменено, 1 — новое, 2 — изменено, 3 — удалено).

## Класс SyncItemSchema C#

 Сложный

Пространство имен `Terrasoft.Sync`.

Схема настройки сущности элемента синхронизации.

## Свойства

`SyncValueName` `String`

Название типа сущности.

`SyncValueType` `Type`

Тип сущности.

`PrimaryEntityConfig` `EntityConfig`

Настройка основной сущности элемента синхронизации.

`Configs` `List<EntityConfig>`



Список настроек сущностей элемента синхронизации.

`DetailConfigs` `List<DetailEntityConfig>`

Список настроек сущностей деталей элемента синхронизации.

`Direction` `SyncDirection`

Определяет направление синхронизации для объектов текущего типа. Значение по умолчанию — `DownloadAndUpload`.

Если не содержит флаг `Download` — изменения не будут применяться в Creatio.

Если не содержит флаг `Upload` — не будут выбираться новые сущности из Creatio.

## Методы

`CreateSyncItemSchema(Type syncValueType)` `SyncItemSchema`

Создает настройки сущности элемента синхронизации со всеми настройками связанных сущностей.

`Validate(UserConnection userConnection)` `Void`

Метод проверяет правильность сформированного `EntityConfig`.

Если проверка не прошла — выдается исключение (в случае, если в `EntityConfig` имя схемы указано два раза — генерируется `DuplicateDataException`, если указано имя несуществующей схемы — `InvalidSyncItemSchemaException`).

`FetchItem(ItemMetadata itemMetaData, SyncItemSchema itemSchema, bool loadAllColumns = false)` `Loc`

Загружает коллекцию сущностей, связанных с конкретным элементом синхронизации.

## Класс EntityConfig C#



Пространство имен `Terrasoft.Sync`.

Настройка сущностей элемента синхронизации.

## Свойства

`SchemaName` `String`

Название схемы объекта.

Order `Int`

Порядок обработки сущности для элемента синхронизации. Чем меньше значение, тем раньше сущность будет обработана при обработке элемента синхронизации.

Direction `SyncDirection`

Определяет направление синхронизации для объектов текущего типа. Значение по умолчанию — `DownloadAndUpload`.

Если не содержит флаг `Download` — изменения не будут применяться в Creatio.

Если не содержит флаг `Upload` — не будут выбираться новые сущности из Creatio.

FetchColumnNames `String[]`

Названия колонок, которые будут загружаться из локального хранилища. Если при создании экземпляра значение не указано, то будут загружаться все колонки объекта.

## Класс DetailEntityConfig C#



Пространство имен `Terrasoft.Sync`.

Настройка сущностей деталей элемента синхронизации.

### Свойства

PrimarySchemaName `String`

Имя схемы основной синхронизируемой сущности Creatio.

ForeignKeyColumnName `String`

Имя колонки для связи детали с главным объектом.

## Класс LocalItem C#



Пространство имен `Terrasoft.Sync`.

Представляет собой один или несколько объектов из Creatio, которые синхронизируются с внешним хранилищем как одно целое. Содержит набор экземпляров классов `SyncEntity`.

## Свойства

`Entities Dictionary<string, List<SyncEntity>>`

Коллекция `SyncEntity`, которая ставится в соответствии с одним `SyncItem`. Содержит коллекцию пар "ключ-значение", где ключ — имя схемы, а значение — коллекция `SyncEntity` этой схемы.

`Version DateTime`

Наибольшее значение даты и времени модификации из всех `Entities` в `LocalItem`.

`Schema SyncItemSchema`

Схема настройки сущности элемента синхронизации.

## Методы

`AddOrReplace(string schemaName, SyncEntity syncEntity) Void`

Добавляет новый `SyncEntity` в коллекцию. В случае если `SyncEntity` с таким `EntityId` уже существует, заменяет его.

`Add(UserConnection userConnection, string schemaName) SyncEntity`

Создает и добавляет новый `SyncEntity` в коллекцию.

# Класс SysSyncMetaData C#



Для синхронизации используется вспомогательная таблица метаданных `SysSyncMetaData`, которая, по сути, является таблицей развязки между внешним `RemoteItem` (элемент синхронизации во внешнем хранилище) и `LocalItem` (элемент синхронизации в Creatio). Каждая строка таблицы представляется в системе экземпляром класса `SysSyncMetaData`.

## Свойства

`RemoteId String`

Идентификатор элемента во внешнем хранилище.

`SyncSchemaName String`

Название схемы синхронизируемого элемента.

---

LocalId Guid

Идентификатор элемента в локальном хранилище.

---

IsLocalDeleted Boolean

Указывает, удален ли элемент из локального хранилища со времени последней синхронизации. Параметр актуализируется перед синхронизацией и при применении изменений в локальном хранилище. На основе его значения при выборке измененных элементов из локального хранилища устанавливается состояние SyncEntity. Устарел, оставлен для совместимости. На текущий момент используется LocalState.

---

IsRemoteDeleted Boolean

Указывает, удален ли элемент из внешнего хранилища со времени последней синхронизации. Устарел, оставлен для совместимости. На текущий момент используется RemoteState.

---

Version Date

Дата последнего выполненного изменения элемента (дата модификации).

---

ModifiedInStoreId Guid

Id хранилища, в котором выполнено последнее изменение.

---

CreatedInStoreId Guid

Id хранилища, в котором создан элемент синхронизации.

---

RemoteStoreId Guid

Идентификатор внешнего хранилища, с которым был синхронизирован элемент.

---

ExtraParameters String

Дополнительные параметры для элемента.

---

LocalState Int

Состояние элемента в локальном хранилище (0 — не изменен, 1 — новый, 2 — изменен, 3 — удален).

RemoteState Int

Состояние элемента во внешнем хранилище (0 — не изменен, 1 — новый, 2 — изменен, 3 — удален).

## Класс MetaDataStore C#

 Сложный

Пространство имен `Terrasoft.Sync`.

Класс-фабрика создает конкретный экземпляр класса, который реализует интерфейс `IReplicaMetadata` для хранилища.

### Методы

`GetReplicaMetadata(Guid localStoreId, Guid remoteStoreId) IreplicaMetadata`

Создает экземпляр класса, реализующий интерфейс `IReplicaMetadata` для конкретного хранилища.

## Класс ItemMetadata C#

 Сложный

Пространство имен `Terrasoft.Sync`.

Этот класс является неделимым объектом метаданных синхронизации. Содержит набор метаданных для каждой сущности синхронизации (коллекция элементов `SysSyncMetadata`).

### Свойства

RemoteId String

Идентификатор элемента во внешнем хранилище.

RemoteItemName String

Наименование элемента во внешнем хранилище.

## Интерфейс IReplicaMetadata C#

 Сложный

Пространство имен `Terrasoft.Sync`.

Класс, реализующий интерфейс `IReplicaMetadata`, инкапсулирует метаданные синхронизации. Реализует

работу с объектами `ItemMetadata`.

## Свойства

`RemoteStoreId` `Guid`

Идентификатор внешнего хранилища.

`LocalStoreId` `Guid`

Идентификатор локального хранилища.

## Методы

`FindItemStore (string remoteItemId) ItemMetadata`

Находит и возвращает объект метаданных синхронизации `ItemMetadata` по идентификатору во внешнем хранилище `remoteItemId`.

`UpdateItemMetadata (ItemMetadata oldItemMetaDatas, IRemoteItem remoteItem, LocalItem localItem,`

Обновляет метаданные после синхронизации.

`EnumerateItemsWithChangesInBpm (SyncContext context) IEnumerable<ItemMetadata>`

Возвращает коллекцию объектов `ItemMetadata`, измененных после последней синхронизации в Creatio и не обработанных в процессе текущего сеанса синхронизации.

`CollectChangesInSyncedEntities (UserConnection userConnection, string schemaName, DateTime lastS`

Обновляет метаданные для элементов синхронизации, измененных в Creatio. Если элемент был изменен в приложении после последней синхронизации `SysMetadata`, колонка `Version` будет заполнена датой модификации элемента. Для актуализации метаданных используется хранимая процедура `ActualizeSysSyncMetadata`.

`CollectNewDetailsForSyncedEntities (UserConnection userConnection, DetailEntityConfig detailEnti`

Создает новые записи в таблице `SysSyncMetadata` для деталей элемента синхронизации.

`TryResolveRemoteId (Guid localId, out string remoteId) Boolean`

Возвращает идентификатор элемента во внешнем хранилище `remoteId` из метаданных по уникальному идентификатору элемента синхронизации в Creatio `localId`. Если элемент помечен как

удаленный, то `remoteId` возвращен не будет и метод возвратит `false`.

---

`TryResolveExtraParameters (Guid localId, out string extraParameters) Boolean`

Возвращает дополнительные параметры `extraParameters` для элемента синхронизации из метаданных по `localId`. Если `extraParameters` не будут найдены, метод вернет `false`.