

Механизм 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

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

Свойства	22
Методы	22
Синхронизация почты с MS Exchange	23
Реализация интеграции	23
Синхронизируемые данные	24
Логика заполнения участников письма	25
Логика выбора данных для синхронизации	26
Синхронизация с MS Exchange	26
Синхронизация задач с MS Exchange	26
Синхронизация контактов с MS Exchange	29
Синхронизация встреч с MS Exchange	33

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.

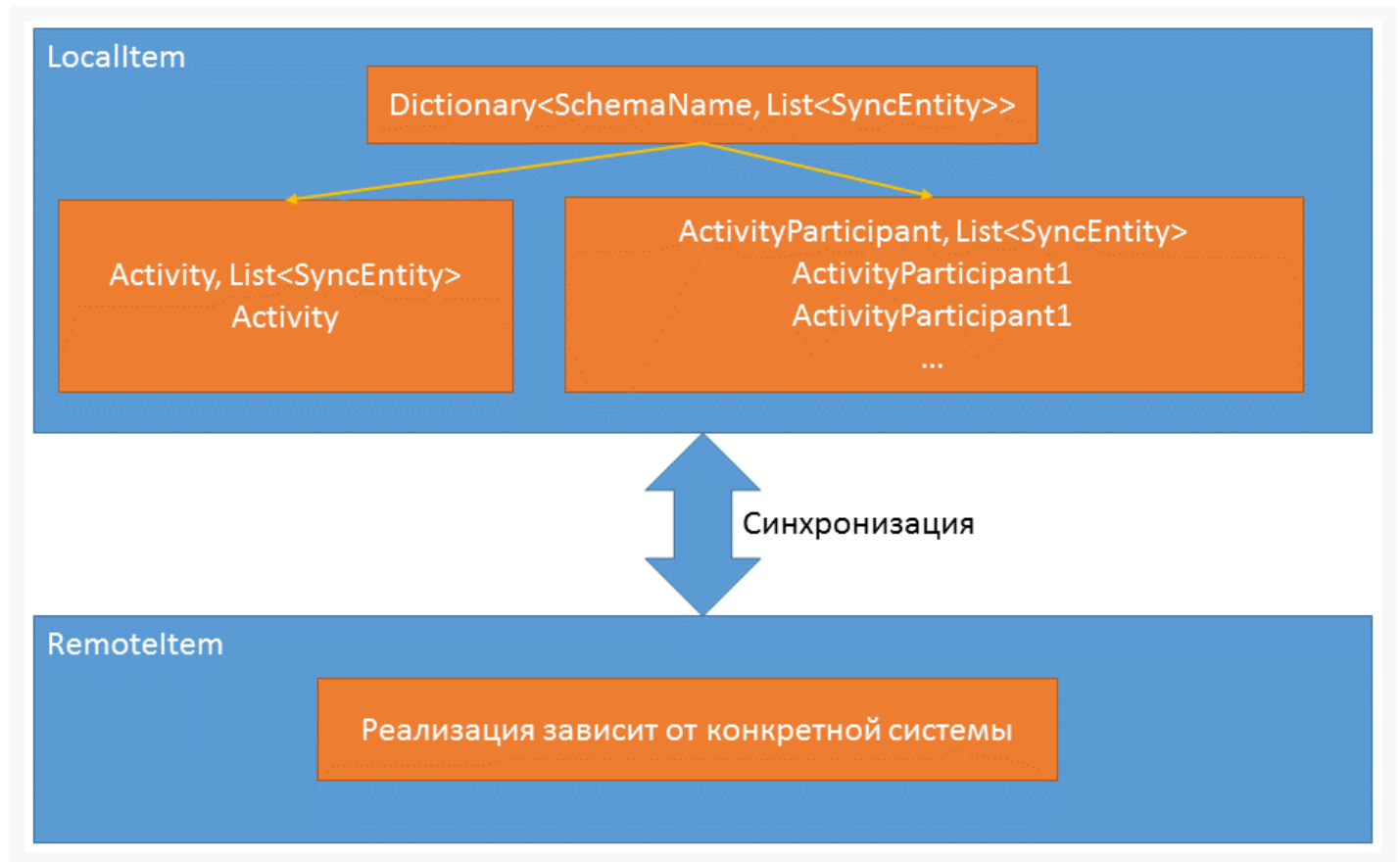
Описание синхронизации

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:



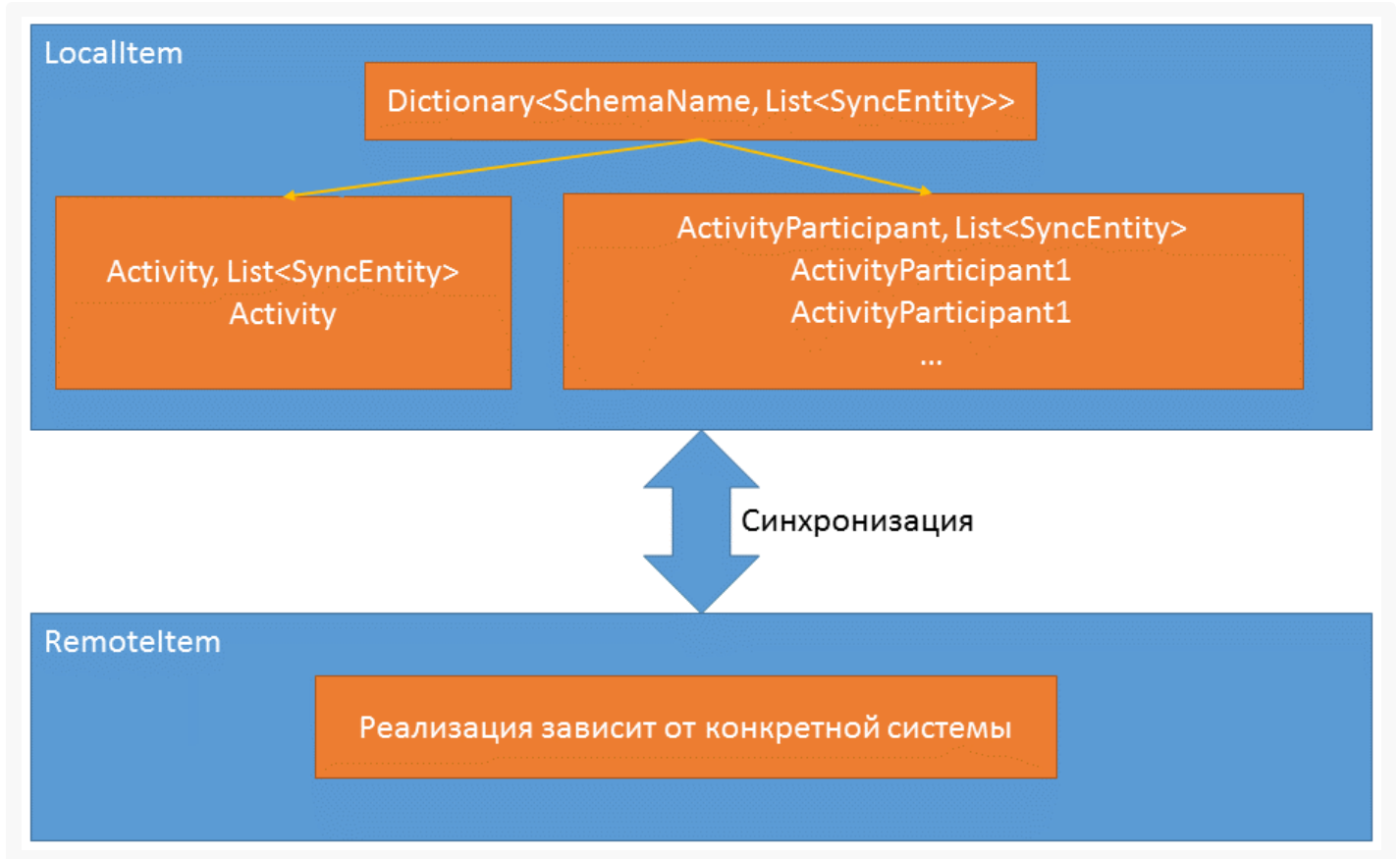
Работа с метаданными для синхронизации

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 C#



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 `Entity`, 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 `Entity`.

```
NeedMetadataActualization() 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#



Сложный

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`.

Синхронизация почты с MS Exchange

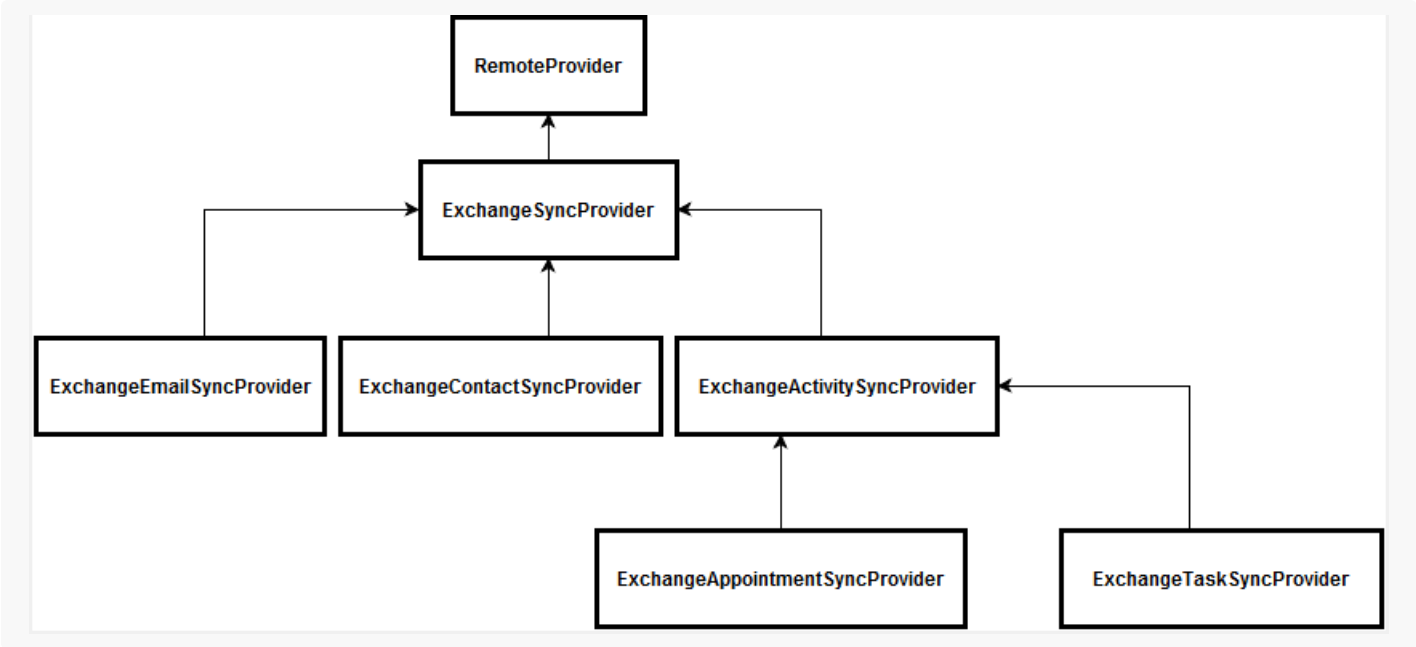


В отличие от общего механизма синхронизации, почта синхронизируется только из MS Exchange в Creatio. В силу того что письма после отправки уже не могут быть изменены, в синхронизации участвуют только ранее не синхронизированные письма. Основным отличием механизма синхронизации почты от других интеграций является механизм поиска письма в Creatio. Так как одно и то же письмо может быть синхронизировано от имени любого из получателей, и даже по протоколу IMAP, то для поиска ранее синхронизированных писем нельзя использовать метаданные синхронизации. Для поиска письма используется тема, дата отправки и тело письма. Из тела письма удаляется разметка и пробельные символы. Для ускорения поиска используется md5 хэш, который хранится в колонке `MailHash` объекта `Activity`.

Вторым отличием данной синхронизации является то, что деталь вложения синхронизируется отдельным процессом, после того как обработаны все письма. Это сделано для того, чтобы время на скачивание вложения не влияло на время сохранения писем.

Реализация интеграции

Как описано в [статье](#), для того чтобы реализовать интеграцию с использованием данного механизма, необходим класс, реализующий логику работы с внешним хранилищем (наследник `RemoteProvider`) и класс, реализующий интерфейс `IRemoteItem`, который представляет один экземпляр элемента синхронизации (в нашем случае — `email MS Exchange`).

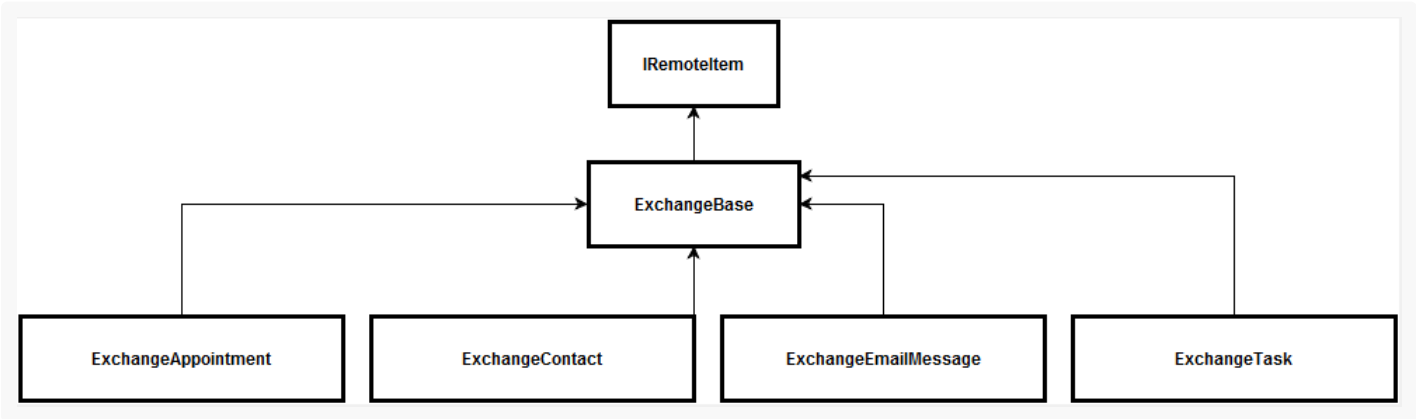


Класс `ExchangeEmailSyncProvider` является провайдером для работы с внешним хранилищем exchange, в нем реализована логика по выбору данных из MS Exchange.

Класс `ExchangeEmailMessage` реализует интерфейс `IRemoteItem` и в нем реализована логика заполнения данных в объектах Creatio.

Класс `ExchangeUtility` содержит утилитные методы для использования библиотеки EWS API и методы, используемые для загрузки вложений писем.

Класс `ExchangeEmailMessageUtility` содержит утилитные методы для преобразования справочных значений из полей email.



Синхронизируемые данные

Соответствие объектов Creatio и полей класса `EmailMessage` MS Exchange отображено в таблице.

Соответствие объектов Creatio и полей класса `EmailMessage` MS Exchange

Объект Creatio	Поле объекта	Соответствующее поле EmailMessage
Activity	Title	Subject
	Body	Body.Text
	Sender	Sender
	Receipient	ToReceipients
	CopyReceipient	CcReceipients
	BlindCopyReceipient	BccReceipients
	SendDate	DateTimeSpent
	Priority	Importance
	DueDate, StartDate	DateTimeReceived
ActivityFile	Name	Name
	Data	Content
	Size	Content.Length

Логика заполнения участников письма

Для того чтобы письмо правильно отображалось только у пользователей, которые его синхронизировали, реализован следующий механизм заполнения детали [*Участники активности*]. Условно эту логику можно разделить на две части:

- 1. Создание участников для нового письма.
- 2. Актуализация списка участников при его изменении (в том числе, повторной синхронизации).

Создание участников для нового письма

Основным значением, которое влияет на то, кто попадет в участники письма, является деталь [*Средства связи*] контакта. Если у контакта на детали [*Средства связи*] есть email, который указан в одном из адресных полей письма ([*от*], [*кому*], [*копия*], [*скрытая копия*]), то контакт может быть добавлен в участники. Дополнительно выполняется проверка, существует ли для этого контакта пользователь системы, не являющийся пользователем портала. Пользователь добавляется в участники только тогда, когда он синхронизирует это письмо. Это позволяет добавить в участники переписки всех внешних контактов, и только тех пользователей, которые синхронизировали это письмо.

Актуализация списка участников письма

Для того чтобы после синхронизации уже существующего письма пользователь попал в участники письма, при изменении письма происходит актуализация участников — все участники, которые не являются пользователями Creatio, удаляются с детали, и затем выполняется алгоритм заполнения детали для нового письма. Таким образом, пользователи, которые ранее синхронизировали письмо, остаются в участниках, новый пользователь добавляется, а список контактов актуализируется.

Логика выбора данных для синхронизации

При выборе писем для синхронизации из папок MS Exchange используется следующий набор фильтров: выбрать письма, которые изменены после даты последней синхронизации почты, и не являются черновиками. Для папок синхронизации существует ограничение: папка "Удаленные" и папка "Конфликтующие элементы" не участвуют в синхронизации. При выборе писем не учитывается наличие метаданных синхронизации. Направление изменений всегда "сохранить изменения в Creatio". При обработке каждого письма вначале проверяется наличие письма в Creatio. Если письмо уже существует в Creatio, выполняется обновление участников, если нет, то создается новое письмо. В конце сессии синхронизации в планировщик добавляется задание на синхронизацию вложений.

Синхронизация с MS Exchange



На базе механизма синхронизации Sync Engine в конфигурации реализована интеграция с различными сущностями MS Exchange по протоколу EWS (Exchange Web Services).

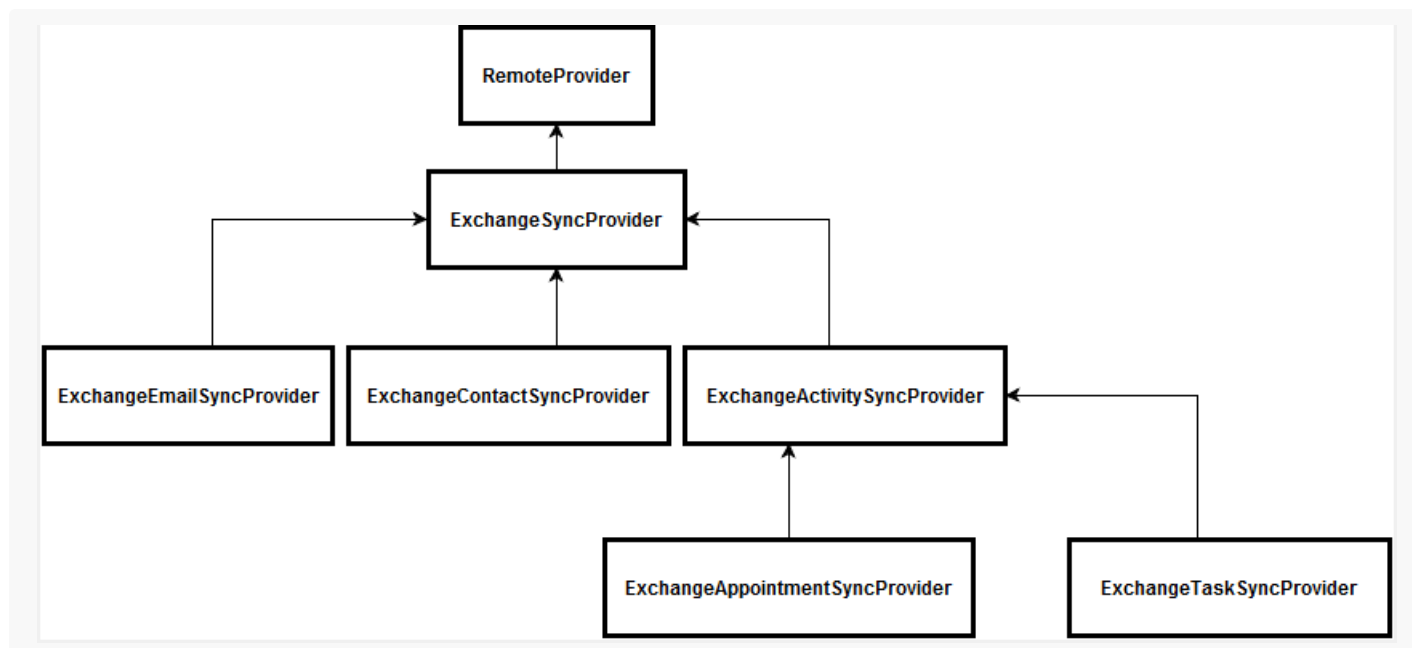
Синхронизация задач с MS Exchange

Алгоритм синхронизации задач происходит в три этапа:

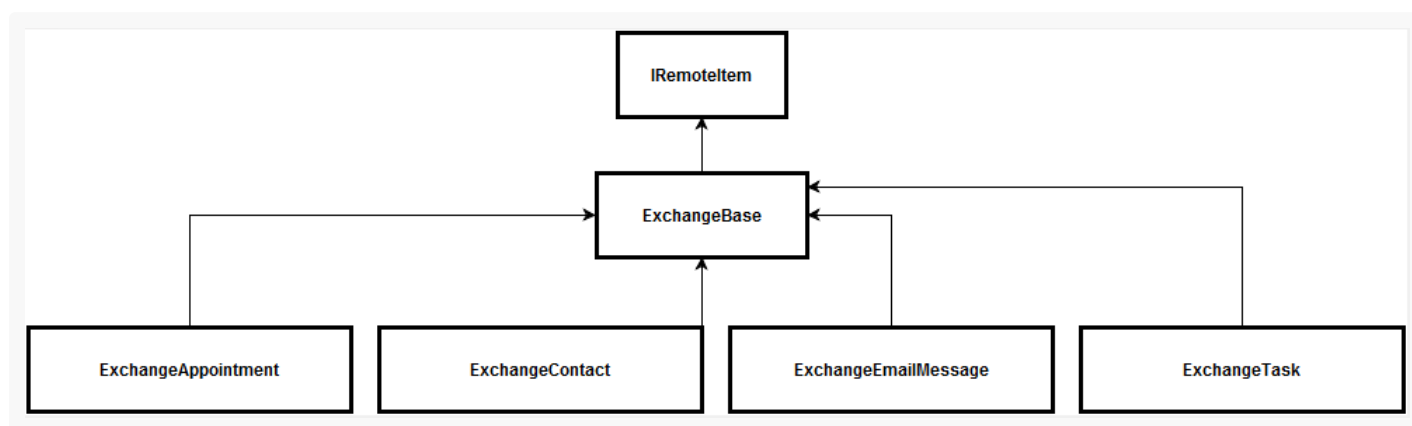
1. Получение изменений из MS Exchange, их применение;
2. Получение изменений из Creatio, их применение;
3. Создание новых задач из Creatio в MS Exchange.

Реализация интеграции

Как описано в [статье](#), для того, чтобы реализовать интеграцию с использованием данного механизма, необходим класс, реализующий логику работы с внешним хранилищем (наследник `RemoteProvider`). Вся иерархия классов провайдера изображена на рисунке. Также необходим класс, реализующий интерфейс `IRemoteItem`, который представляет один экземпляр элемента синхронизации (в данном случае — задачу MS Exchange). Вся иерархия `RemoteItem` изображена на рисунке.



Класс `ExchangeTaskSyncProvider` является провайдером для работы с внешним хранилищем MS Exchange, в нем реализована логика по выбору данных и сохранению изменений в Creatio и MS Exchange. Класс `ExchangeTask` реализует интерфейс `IRemoteItem`. В нем реализована логика заполнения данных в соответствующих системах.



Синхронизируемые данные

Соответствие объектов Creatio и полей класса `ExchangeTask` отображено в таблице.

Соответствие объектов Creatio и полей класса `ExchangeTask`

Объект Creatio	Поле объекта	Поле <code>ExchangeTask</code>
Activity	Title	Subject
	StartDate	StartDate
	DueDate	CompleteDate или DueDate в зависимости от того, завершена задача или нет.
	Priority	Importance
	Status	Status
	RemindToOwner	IsRemindSet
	RemindToOwnerDate	ReminderDueBy
	Notes	Body.Text

Логика выбора данных для синхронизации

Для выбора изменений списка задач из отобранных для синхронизации папок MS Exchange используются следующие условия: выбрать задачи MS Exchange, которые были изменены после даты последней синхронизации задач, или задачу MS Exchange, которая не была отмечена, как ранее синхронизированная. Для задач MS Exchange, которые были изменены, находятся соответствующие активности в Creatio. Актуальные изменения применяются в соответствующей системе.

При выборе измененных активностей Creatio выбираются активности:

- имеющие запись в метаданных синхронизации как задача MS Exchange;
- автором которых является текущий пользователь;
- дата последнего изменения которых больше чем дата последней синхронизации.

При выборе новых активностей Creatio, устанавливается набор общих фильтров и фильтры, настраиваемые пользователем. Основными фильтрами являются:

1. Тип активности не равен "email".
2. У активности не установлен признак [*Отображать в расписании*].

Пользователь может указать группы активностей, которые будут экспортироваться из Creatio.

Заполнение полей [*Дата начала*], [*Дата завершения*] (опционально)

В объекте `ExchangeTask` есть несколько особенностей при работе с датами начала и завершения:

- В этих полях хранятся значения без времени. При изменении задачи в MS Exchange после синхронизации в Creatio применится дата из задачи MS Exchange, а время — из активности Creatio.
- Для даты завершения в `ExchangeTask` есть два поля — дата завершения (due date) и дата выполнения (complete date). В зависимости от статуса задачи, актуальной является одна из них — для не завершенных используется дата завершения, для завершенных дата выполнения. Дата выполнения устанавливается текущей датой, пока задача не будет завершена.
- Дата начала и дата завершения — поля, необязательные для заполнения в MS Exchange. Если какое-то из них не установлено, используется текущая дата. Из-за этого могут возникать конфликты, так как в Creatio и дата начала, и дата завершения обязательны для заполнения, и дата начала должна быть меньше даты завершения.

Синхронизация контактов с MS Exchange

Алгоритм синхронизации контактов происходит в три этапа:

1. Получение изменений из MS Exchange, их применение;
2. Получение изменений из Creatio, их применение;
3. Создание новых контактов из Creatio в MS Exchange.

Реализация интеграции

Как описано в [статье](#), для того, чтобы реализовать интеграцию с использованием данного механизма, необходим класс, реализующий логику работы с внешним хранилищем, — наследник `RemoteProvider` и класс, реализующий интерфейс `IRemoteItem`, который представляет один экземпляр элемента синхронизации (в данном случае — контакт MS Exchange).

Схема иерархии RemoteProvider

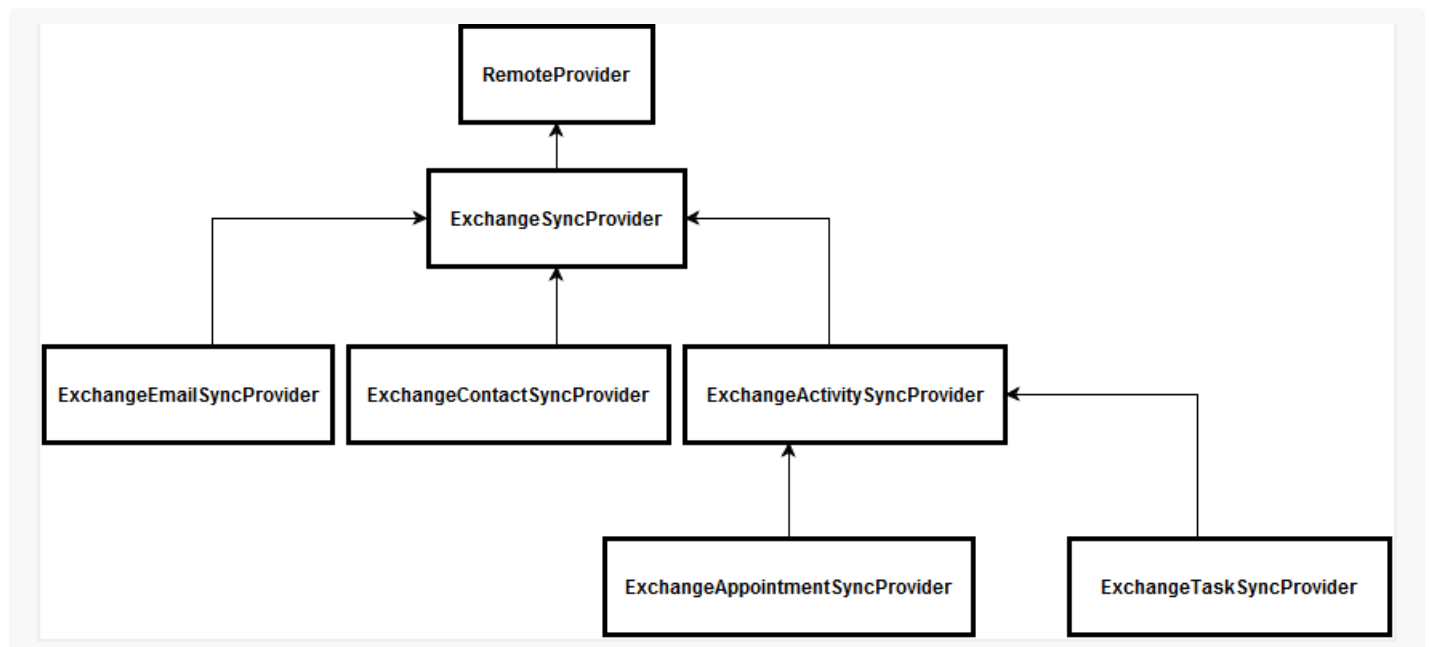
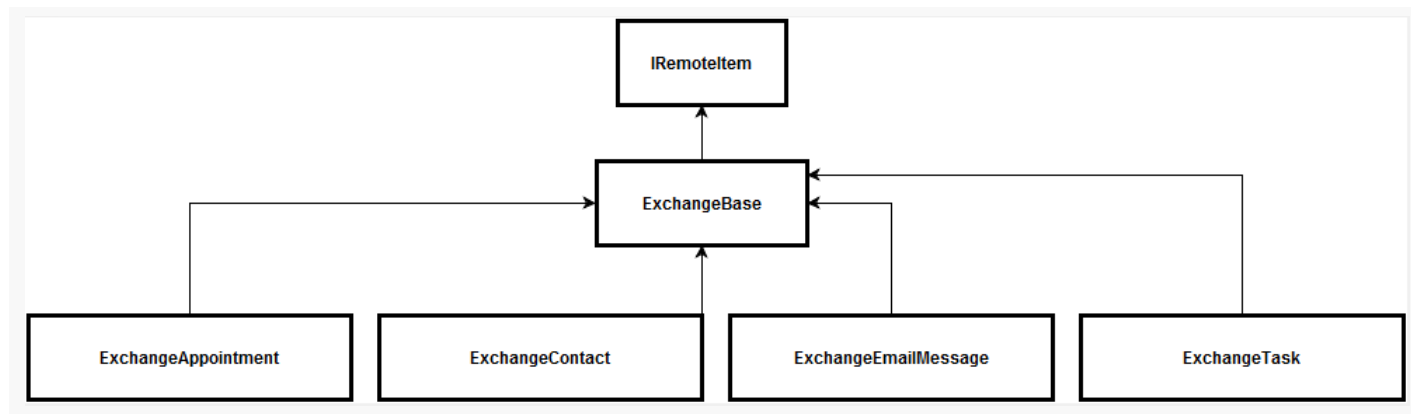


Схема иерархии RemoteItem



Для синхронизации контактов также используются следующие классы:

- Класс `ExchangeContactSyncProvider` является провайдером для работы с внешним хранилищем MS Exchange, в нем реализована логика по выбору данных и сохранению изменений в Creatio и MS Exchange.
- Класс `ExchangeContact` реализует интерфейс `IRemoteItem`. В нем реализована логика заполнения данных в соответствующих системах.
- Класс `ExchangeAddressDetailsSynchronizer` содержит утилитные методы для преобразования данных об адресах контактов.
- Класс `ExchangeEmailAddressDetailsSynchronizer` содержит утилитные методы для преобразования данных об email-адресах контактов.
- Класс `ExchangePhoneNumbersDetailsSynchronizer` содержит утилитные методы для преобразования данных о телефонах контактов.

Логика заполнения деталей вынесена в отдельные классы, так как существуют значительные отличия в форматах хранения данных в Creatio и MS Exchange, и требуется дополнительное преобразование.

Синхронизируемые данные

Соответствие объектов Creatio и полей класса Contact MS Exchange отображено в таблице.

Соответствие объектов Creatio и полей класса Contact MS Exchange

Объект Creatio	Поле объекта	Поле класса Contact MS Exchange
Contact	Name	DisplayName
	Surname	Surname
	GivenName	GivenName
	MiddleName	MiddleName
	Account	CompanyName
	JobTitle	JobTitle
	Department	Department
	BirthDate	Birthday
	SalutationType	TitleTag
	Gender	GenderTag
ContactCommunication	Number	Значение в коллекции PhoneNumbers
	CommunitactionType	Ключ для значения в коллекции PhoneNumbers
ContactAddresses	City	Поле City элемента коллекции PhysicalAddresses
	Country	Поле CountryOrRegion элемента коллекции PhysicalAddresses
	Region	Поле State элемента коллекции PhysicalAddresses
	Address	Поле Street элемента коллекции PhysicalAddresses
	Zip	Поле PostalCode элемента коллекции PhysicalAddresses
	AddressType	Ключ для значения в коллекции PhysicalAddresses

Соответствие типов средств связи представлено в таблице.

Соответствие типов средств связи Creatio и MS Exchange

Тип средства связи Creatio	Тип средства связи MS Exchange
Email	EmailAddress1, EmailAddress2, EmailAddress3
WorkPhone	BusinessPhone, BusinessPhone2
HomePhone	HomePhone
MobilePhone	MobilePhone

Соответствие типов адресов представлено в таблице.

Соответствие типов адресов Creatio и MS Exchange

Тип адреса Creatio	Тип адреса MS Exchange
HomeAddress	Home
BusinessAddress	Business

Логика выбора данных для синхронизации

Для выбора изменений в списке контактов из отобранных для синхронизации папок MS Exchange используется следующий набор фильтров: выбирается контакт MS Exchange, который был изменен после даты последней синхронизации контактов, или контакт MS Exchange, который не был отмечен, как ранее синхронизированный. Для контактов, которые были изменены, находится соответствующий контакт в Creatio. Самые актуальные изменения применяются в соответствующей системе.

При выборе измененных контактов Creatio выбираются контакты:

- автором которых является текущий пользователь;
- дата последнего изменения которых больше чем дата последней синхронизации;
- контакт не использовался в первом этапе синхронизации.

На правила выбора новых контактов Creatio, помимо стандартного фильтра по автору и фильтра отсутствия метаданных синхронизации, влияют пользовательские настройки. Доступно три варианта настройки:

1. Синхронизировать контакты сотрудников. При выборе этой настройки к запросу будет добавлен фильтр по колонке [Тип контакта], и синхронизированы будут те контакты, у которых указан тип "Сотрудник".
2. Синхронизировать контакты клиентов. При выборе этой настройки к запросу будет добавлен фильтр по колонке [Тип контакта], и синхронизированы будут те контакты, у которых указан тип "Клиент".
3. Синхронизировать контакты из определенных групп. При выборе этой настройки к запросу будут добавлены фильтры выбранных групп контактов.

Дополнительные возможности

Возможность использовать расширенные ключи для контактов во внешнем хранилище

Ввиду особенностей работы MS Exchange возможна ситуация, когда при большом количестве контактов MS Exchange часть из них получит одинаковые идентификаторы, и, как следствие, синхронизация может некорректно определить соответствующий контакт в Creatio. Для обхода подобной ситуации реализована функциональность расширенных внешних ключей, которую можно включить при помощи настройки [*Использовать комбинированные идентификаторы для синхронизируемых контактов MS Exchange*]. Код настройки — `UseComplexExchangeContactId`. После ее включения может понадобиться повторная синхронизация.

Синхронизация встреч с MS Exchange

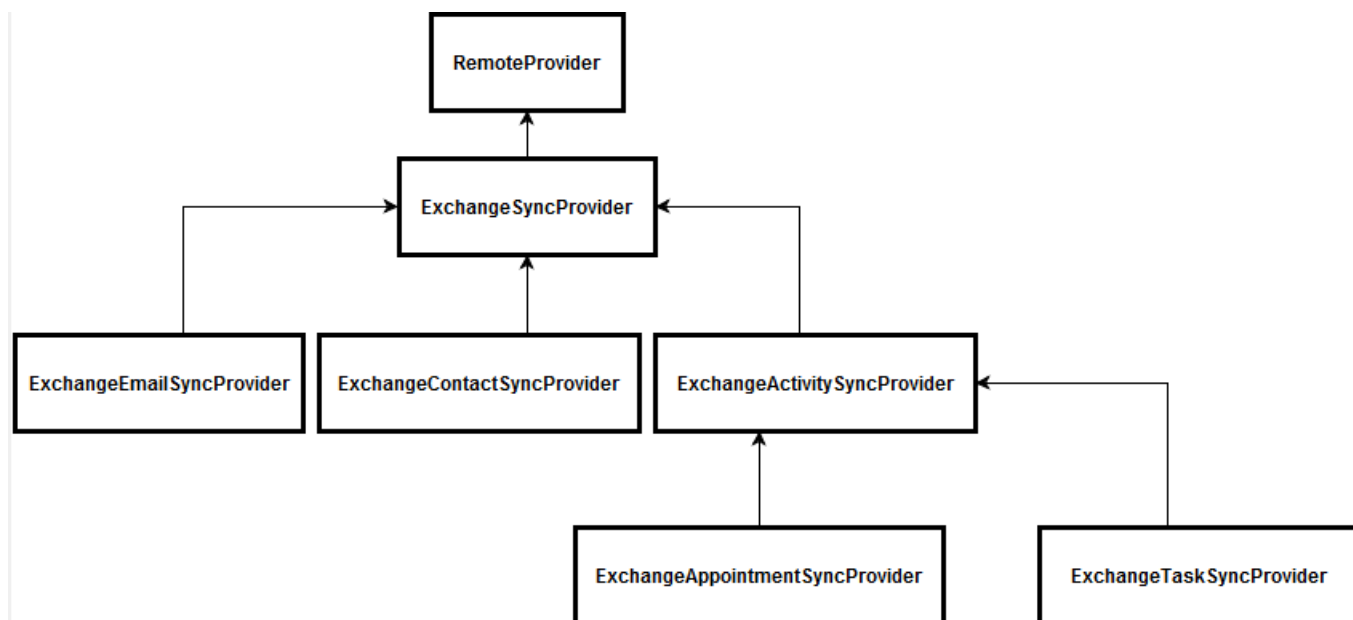
Синхронизация встреч Creatio происходит только для новых активностей или при изменении полей `Title`, `Location`, `StartDate`, `DueDate`, `Priority`, `Notes`. По этим полям формируется хэш, который хранится в дополнительных параметрах метаданных (в поле `ExtraParameters`). Если встреча была изменена в Creatio, и хэш из `ExtraParameters` не совпадает с новым хэшем, то эта встреча должна быть синхронизирована.

Алгоритм синхронизации встреч происходит в три этапа:

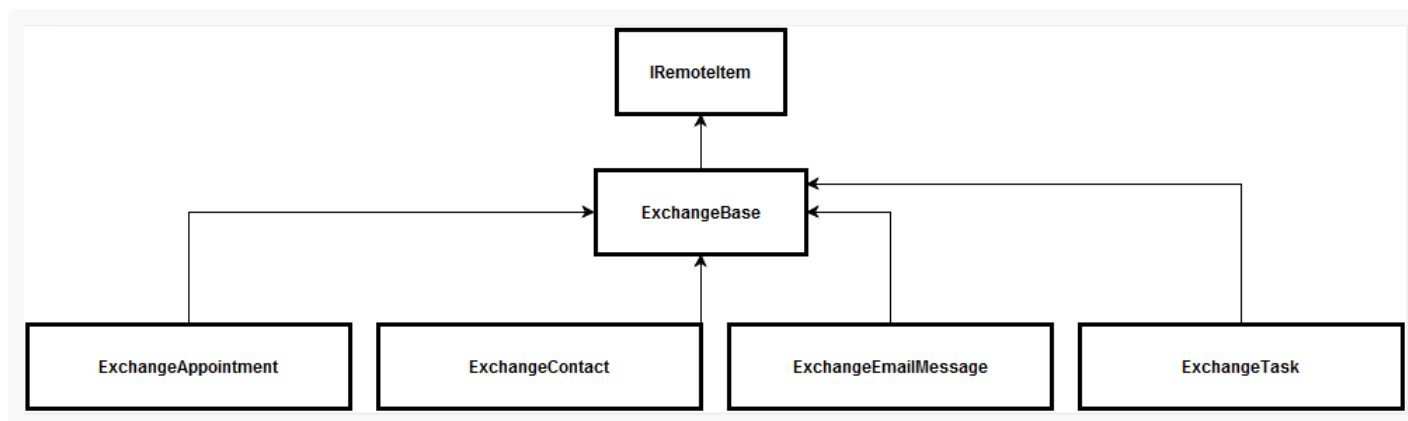
1. Получение изменений из MS Exchange, их применение;
2. Получение изменений из Creatio, их применение;
3. Создание новых встреч из Creatio в MS Exchange.

Реализация интеграции

Как описано в [статье](#), для того чтобы реализовать интеграцию с использованием данного механизма, необходим класс, реализующий логику работы с внешним хранилищем, — наследник `RemoteProvider`, и класс, реализующий интерфейс `IRemoteItem`, который представляет один экземпляр элемента синхронизации (в данном случае — встречу MS Exchange).



Класс `ExchangeAppointmentSyncProvider` является провайдером для работы с внешним хранилищем Exchange. В нем реализована логика по выбору данных и сохранению изменений в Creatio и Exchange. Класс `ExchangeAppointment` реализует интерфейс `IRemoteItem`, и в нем реализована логика заполнения данных в соответствующих системах.



Синхронизируемые данные

Соответствие объектов Creatio и полей класса `ExchangeAppointment` отображено в таблице.

Соответствие объектов Creatio и полей класса `ExchangeAppointment`

Объект Creatio	Поле объекта	Соответствующее поле MS Exchange Appointment
Activity	Title	Subject
	Location	Location
	StartDate	StartDate
	DueDate	CompleteDate или DueDate в зависимости от того, завершена встреча или нет.
	Priority	Importance
	Status	Заполняется по алгоритму: Если статус не указан и дата окончания больше, чем текущая, — статус в Creatio устанавливается как "Новая встреча". Если дата окончания меньше, чем текущая дата, — статус устанавливается как закрытая встреча со статусом "Информация получена".
	RemindToOwner	IsReminderSet
	RemindToOwnerDate	ReminderDueBy
	Notes	Body.Text
ActivityParticipant	InviteResponse	Если в MS Exchange установлен признак, что встреча принята или пользователь является организатором встречи, то в Creatio устанавливается признак "Встреча подтверждена". В противном случае устанавливается признак того, что встреча отменена.

Логика выбора данных для синхронизации

Для выбора изменений в списке встреч из отобранных для синхронизации папок MS Exchange используется следующий набор фильтров: выбираются встречи MS Exchange, которые были изменены после даты последней синхронизации встреч, или встречи MS Exchange, которые не были отмечены, как ранее синхронизированные. Для встреч, которые были изменены, находятся соответствующие активности в Creatio. Самые актуальные изменения применяются в соответствующей системе.

При выборе измененных активностей Creatio выбираются активности:

- которые имеют запись в метаданных синхронизации как встречи MS Exchange по RemoteId (определяется уникальным идентификатором встречи в календаре ICalId);
- дата последнего изменения которых больше, чем дата последней синхронизации;
- одна встреча в Creatio соответствует нескольким встречам в MS Exchange для каждого участника.

При выборе новых активностей Creatio устанавливается набор общих фильтров и фильтры, настраиваемые пользователем. Основными фильтрами являются:

1. Тип активности не равен "Email".
2. У активности установлен признак [*Отображать в расписании*].

Пользователь может указать группы активностей, которые будут экспортироваться из Creatio.

Логика выбора участников встречи

При синхронизации активности из MS Exchange в Creatio на деталь [*Участники*] добавляются те контакты, у которых на детали [*Средства связи*] есть email из списка учасников встречи в MS Exchange.

При синхронизации активностей из Creatio в MS Exchange участники встречи для MS Exchange заполняются основными email-адресами контактов.