

Получение данных статей

В данном разделе описаны методы класса [DBConnector](#) для получения данных статей (как обычных полей, так и связей).

GetContentData

```
public DataTable GetContentData(string siteName, string contentName, string fields, string whereExpression, string orderExpression, long startRow, long pageSize, ref long totalRecords, byte useSchedule, string statusName, byte showSplittedArticle, byte includeArchive, bool cacheResult, double cacheInterval, bool useClientSelection, bool withReset)
public DataTable GetContentData(string siteName, string contentName, string fields, string whereExpression, string orderExpression, long startRow, long pageSize, ref long totalRecords, byte useSchedule, string statusName, byte showSplittedArticle, byte includeArchive, bool cacheResult, double cacheInterval)
public DataTable GetContentData(string siteName, string contentName, string whereExpression, string orderExpression, long startRow, long pageSize, ref long totalRecords, byte useSchedule, string statusName, byte showSplittedArticle, byte includeArchive)
public DataTable GetContentData(string siteName, string contentName, string fields, string whereExpression, string orderExpression, long startRow, long pageSize, ref long totalRecords, byte useSchedule, string statusName, byte showSplittedArticle, byte includeArchive)
```

Данный метод был создан как упрощенный вариант получения данных из контента вместо более тяжелого [Publishing Container](#) для сайтов, построенных на объектах QP7. Но он может использоваться и для сайтов/приложений, созданных по другой технологии.

Метод возвращает не саму таблицу из кэша, а ее копию, так что ее можно изменять.

Метод не выполняет автоматическую [замену плейсхолдеров на URL](#), поэтому при использовании текстовых полей, которые могут содержать URL, нужно вызывать метод [FormatField](#).

Для связей Many-to-Many и Many-to-One метод возвращает данные, которые реально хранятся в таблице контента (это идентификатор связи для Many-to-Many и идентификатор базового поля для Many-to-One). Используя эти данные, можно получить реальные данные связей статьи вызовом метода [GetContentItemLinkIDs](#).

Обязательные параметры

- *siteName* – имя сайта
- *contentName* – имя контента
- *whereExpression* – пользовательская часть предложения WHERE SQL-запроса (аналог поля

Фильтр (Filter) для Publishing Container)

- *orderExpression* – предложение ORDER SQL-запроса (аналог поля **Выражение динамической сортировки (Dynamic order expression)** для Publishing Container)
- *startRow* – параметр постраничного вывода (аналог поля **Номер первой отображаемой записи (First article number)** для Publishing Container)
- *pageSize* – параметр постраничного вывода (аналог поля **Количество (Count)** для Publishing Container)
- *totalRecords* – параметр возвращает количество записей с учетом фильтрации, но без учета постраничного вывода
- *useSchedule* – дополнительная фильтрация статей, для которых выставлено расписание (аналог поля **Использовать расписание статей (Use Articles Schedule)** для Publishing Container)
- *statusName* – имя статуса (аналог поля **Статус (Status)** для Publishing Container)
- *showSplittedArticle* – указывает, какую версию статьи возвращать: текущую или опубликованную. Если передан 0, возвращается опубликованная версия статьи (live), иначе текущая (stage).
- *includeArchive* – определяет, включать ли архивные статьи в результат (аналог поля **Показать архивированные (Show archived)** для Publishing Container)

Необязательные параметры

- *fields* – набор полей, которые необходимо вернуть. Служебные поля автоматически не попадают в результат, то есть их тоже нужно указывать (при необходимости). При отсутствии параметра возвращаются все поля.
- *cacheResult* – кэшировать или нет результат. Значение по умолчанию - *false*.
- *cacheInterval* – продолжительность кэширования. По умолчанию используется значение из конфигурации - [InternalExpirationTime](#)
- *useClientSelection* – при передаче *true* фильтрация, сортировка и постраничный вывод осуществляется на клиентской стороне, что позволяет уменьшить объем кэшируемых данных. Следует иметь в виду, что клиентская фильтрация поддерживает только часть возможностей серверной. [Подробнее о синтаксисе](#). Значение по умолчанию - *false*.
- *withReset* – при передаче *true* осуществляется принудительный сброс кэша для данного вызова и повторное его заполнение. Значение по умолчанию - *false*.

Пример:

```
long total = ;
DataTable dt = cnn.GetContentData("Sandbox NET", "Events",
"[content_item_id], [Title]", "[Location] = 1662", "[Modified] DESC", 1, 20,
ref total, 1, "Published", , );
```

GetCachedContentData

```
public DataTable GetCachedContentData(string siteName, string contentName,
string whereExpression, string orderExpression, long startRow, long
pageSize, ref long totalRecords, byte useSchedule, string statusName, byte
showSplittedArticle, byte includeArchive)
public DataTable GetCachedContentData(string siteName, string contentName,
```

```
string whereExpression, string orderExpression, long startRow, long
pageSize, ref long totalRecords, byte useSchedule, string statusName, byte
showSplittedArticle, byte includeArchive, bool useClientSelection)
public DataTable GetCachedContentData(string siteName, string contentName,
string whereExpression, string orderExpression, long startRow, long
pageSize, ref long totalRecords, byte useSchedule, string statusName, byte
showSplittedArticle, byte includeArchive, double cacheInterval)
public DataTable GetCachedContentData(string siteName, string contentName,
string whereExpression, string orderExpression, long startRow, long
pageSize, ref long totalRecords, byte useSchedule, string statusName, byte
showSplittedArticle, byte includeArchive, double cacheInterval, bool
useClientSelection)
public DataTable GetCachedContentData(string siteName, string contentName,
string fields, string whereExpression, string orderExpression, long
startRow, long pageSize, ref long totalRecords, byte useSchedule, string
statusName, byte showSplittedArticle, byte includeArchive)
public DataTable GetCachedContentData(string siteName, string contentName,
string fields, string whereExpression, string orderExpression, long
startRow, long pageSize, ref long totalRecords, byte useSchedule, string
statusName, byte showSplittedArticle, byte includeArchive, bool
useClientSelection)
public DataTable GetCachedContentData(string siteName, string contentName,
string fields, string whereExpression, string orderExpression, long
startRow, long pageSize, ref long totalRecords, byte useSchedule, string
statusName, byte showSplittedArticle, byte includeArchive, double
cacheInterval)
public DataTable GetCachedContentData(string siteName, string contentName,
string fields, string whereExpression, string orderExpression, long
startRow, long pageSize, ref long totalRecords, byte useSchedule, string
statusName, byte showSplittedArticle, byte includeArchive, double
cacheInterval, bool useClientSelection)
public DataTable GetCachedContentData(string siteName, string contentName,
string fields, string whereExpression, string orderExpression, long
startRow, long pageSize, ref long totalRecords, byte useSchedule, string
statusName, byte showSplittedArticle, byte includeArchive, bool
useClientSelection, bool withReset)
```

Аналог метода [GetContentData](#), кэширующий результат во всех случаях (если кэширование [не запрещено](#) на уровне экземпляра класса DBConnector).

GetContentItemLinkIds

```
public string GetContentItemLinkIds(string linkFieldName, long itemId)
public string GetContentItemLinkIds(string linkFieldName, string itemIds)
public string GetContentItemLinkIds(int linkId, long itemId)
public string GetContentItemLinkIds(int linkId, string itemIds)
public string GetContentItemLinkIds(int linkId, long itemId, bool
isManyToMany)
public string GetContentItemLinkIds(int linkId, string itemIds, bool
```

isManyToMany)

Получение данных для полей типа Many-to-Many и Many-to-One. Тип связи задается параметром *isManyToMany* (значение по умолчанию - *true*). Связь задается либо именем поля в контенте (параметр *linkFieldName*), либо идентификатором связи (параметр *linkId*).

Параметр *linkId* имеет разный смысл для полей типа Many-to-Many и Many-to-One: для Many-to-Many - это идентификатор связи (записи в таблице CONTENT_TO_CONTENT), для Many-to-One - это идентификатор базового поля One-to-Many (записи в таблице CONTENT_ATTRIBUTE). Эти идентификаторы хранятся в соответствующих полях контентной таблицы (CONTENT_NNN) и могут быть получены методом [GetContentData](#), что позволяет избежать дополнительных запросов к базе.

Идентификатор статьи, для которой ищутся связи, можно задать параметром *itemId*. Если нужно найти связи сразу для нескольких статей, можно воспользоваться одной из перегруженных версий метода, принимающей параметр *itemIds* - строку идентификаторов статей через запятую.

Результат возвращается в виде строки идентификаторов, связанных статей через запятую. К результирующей строке всегда добавляется 0, так что строка никогда не пустая, и ее можно использовать в SQL-конструкции IN без дополнительного кода.

На поведение данного метода влияют свойства [IsStage](#) и [CacheData](#). Если *IsStage* == *true*, то возвращаются текущие версии полей Many-to-Many и Many-to-One, иначе только опубликованные. Если *CacheData* == *true*, то результат выполнения кэшируется на интервал времени [InternalShortExpirationTime](#).

GetRealContentItemLinkIds

```
public string GetRealContentItemLinkIds(string linkFieldName, long itemId)
public string GetRealContentItemLinkIds(string linkFieldName, string
itemId)
public string GetRealContentItemLinkIds(int linkId, long itemId)
public string GetRealContentItemLinkIds(int linkId, string itemIds)
public string GetRealContentItemLinkIds(int linkId, long itemId, bool
isManyToMany)
public string GetRealContentItemLinkIds(int linkId, string itemIds, bool
isManyToMany)
```

Поведение данного метода полностью совпадает с методом [GetContentItemLinkIds](#), за исключением того, что результат его выполнения не кэшируется вне зависимости от свойства [CacheData](#).

GetContentItemLinkQuery

```
public string GetContentItemLinkQuery(string linkFieldName, long itemId)
public string GetContentItemLinkQuery(string linkFieldName, string itemIds)
public string GetContentItemLinkQuery(int linkId, long itemId)
```

```
public string GetContentItemLinkQuery(int linkId, string itemIds)
public string GetContentItemLinkQuery(int linkId, long itemId, bool
isManyToMany)
public string GetContentItemLinkQuery(int linkId, string itemIds, bool
isManyToMany)
```

Метод похож на [GetContentItemLinkIds](#), но он возвращает не результат запроса к базе, а сам SQL-запрос.

From:

<http://wiki.qpublishing.ru/> - **QP7.Framework Docs**

Permanent link:

http://wiki.qpublishing.ru/doku.php?id=api:dbconnector:get_article

Last update: **2012/06/20 17:44**

