



TMS FNC Cloud Pack
Services
DEVELOPERS GUIDE

Index

Availability	3
Online references	3
Terms of use	3
Limited warranty	4
Description	4
Registering your application.....	4
Getting Started	4
Demo	5
Services.....	7
TTMSFNCCloudCloudKit	7
TTMSFNCCloudDropBox.....	8
TTMSFNCCloudGoogleCalendar	9
TTMSFNCCloudGoogleDrive.....	10
TTMSFNCCloudGoogleFirebaseObjectDatabase	11
TTMSFNCCloudGoogleGMail.....	13
TTMSFNCCloudGoogleLookup.....	14
TTMSFNCCloudGoogleLocationLookup.....	14
TTMSFNCCloudGooglePeople	15
TTMSFNCCloudGooglePhotos	16
TTMSFNCCloudGoogleTasks.....	17
TTMSFNCCloudGoogleYouTube	18
TTMSFNCCloudHubic.....	19
TTMSFNCCloudMicrosoftOneDrive.....	20
TTMSFNCCloudMicrosoftOutlookCalendar.....	21
TTMSFNCCloudMicrosoftOutlookContacts	22
TTMSFNCCloudMicrosoftOutlookMail	23
TTMSFNCCloudMyCloudData.....	24
TTMSFNCCloudPayPal	36
TTMSFNCCloudSlack.....	37
Components	40
TMSFNCCloudDemoListBox.....	40
TMSFNCCloudImage	41

Availability

Supported frameworks and platforms

- VCL: Win32/Win64
- FMX: Win32/Win64, MacOS-X, iOS, Android
- LCL: Win32/Win64, Mac OS-X, iOS, Android, numerous Linux variants including Raspbian
- WEB: Chrome, Edge, Firefox, ...

Supported IDE's

- Delphi XE7 and C++ Builder XE7 or newer releases
- Lazarus 1.8.0 with FPC 3.0.4 or newer releases.

Important Notice: TMS FNC Cloud Pack requires TMS FNC Core (separately available at the [My Products page](#))

Online references

TMS software website:

<http://www.tmssoftware.com>

TMS FNC Cloud Pack page:

<http://www.tmssoftware.com/site/tmsfnccloudpack.asp>

Terms of use

With the purchase of TMS FNC Cloud Pack, you are entitled to our consulting and support services to integrate the supported cloud services in Delphi applications and with this consulting and support comes the full source code needed to do this integration. As TMS FNC Cloud Pack uses the supported services you are bound to the terms of these services that can be found at:

<https://developers.google.com/terms>

<https://docs.microsoft.com/en-us/legal/microsoft-apis/terms-of-use>

<https://www.dropbox.com/terms>

<https://www.youtube.com/static?template=terms>

https://www.paypal.com/webapps/mpp/ua/legalhub-full?country.x=US&locale.x=en_US

https://hubic.com/en/contracts/Contrat_hubiC_2014.pdf

<https://www.myclouddata.net/#/tos>

TMS software is not responsible for the use of TMS FNC Cloud Pack components. The purchase of TMS FNC Cloud Pack does not include any license fee that you might possibly be required to pay to the supported services. It will depend on your type of usage of these services whether a license fee needs to be paid.

It is the sole responsibility of the user or company providing the application that integrates the terms and conditions of the supported services. TMS software does not take any responsibility nor indemnifies any party violating the terms and conditions of the supported services.

We cannot guarantee that a 3rd party cloud service will approve or allow the use of the services used in TMS FNC Cloud Pack components in your application(s) now or at any time in the future.

This is up to the 3rd party cloud service provider to decide and not under our control. In case the 3rd party cloud service provider makes changes to the conditions for using the service, the API itself or

anything else that causes an incompatibility with our component implementations, tmssoftware.com will do its best to accommodate the changes but cannot be forced in any way or within any timeframe to do so and might be technically limited to do so.

Limited warranty

TMS software cannot guarantee the current or future operation and uptime of the supported services.

TMS software offers the consulting and support for TMS FNC Cloud Pack in good faith that the supported services are reliable and future-proof.

In no case, TMS software shall offer refunds or any other compensation in case the supported services terms/operation changes or stops.

Description

TMS FNC Cloud Pack is a set of components to access and use existing cloud services via REST and that offer this in a combination of 2 parts:

- Core
- Ready-to-use service implementation

This documentation focuses on the “Ready-to-use service implementation”. The core implementation part that can be used as a guide to implement any compatible custom service is explained in a separate documentation (TMS FNC Cloud Pack Services (PDF)).

Registering your application

Most services require authentication and authorization before use. A first step will be to register your application with the different cloud services so you can obtain an application key, secret and callback URL. Please refer to our [online documentation](#) for detailed instructions.

Getting Started

Once your application is registered and you have an application ID or client ID, an application secret or client secret and a callback URL, you can get started to get access to the cloud service(s). Most components work in a similar way:

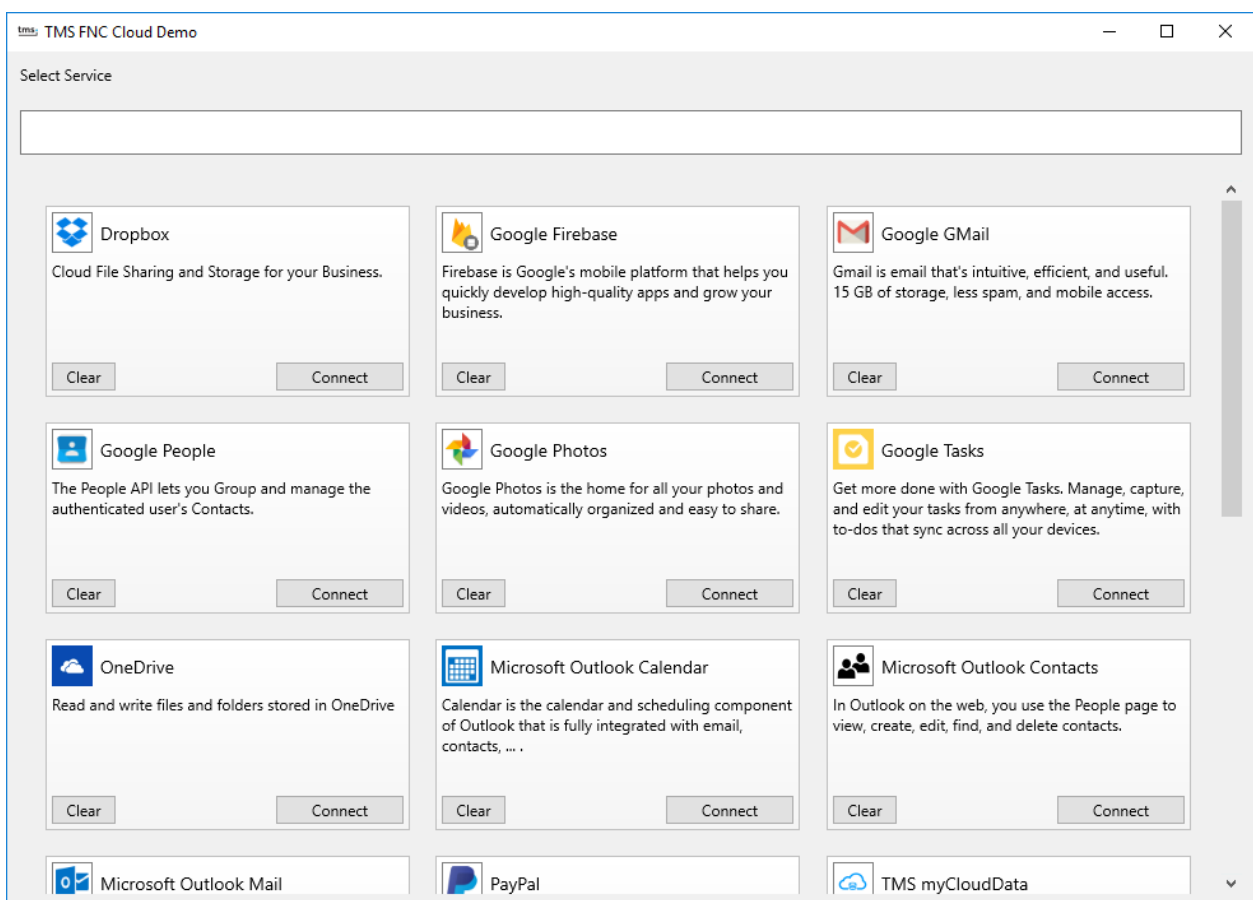
- 1) Drop the component on the form
- 2) Setup the required authorization information. For most services this is client ID, client secret and callback URL via the `.Authentication.ClientID`, `.Authentication.Secret` and `.Authentication.CallbackURL` properties. See detailed information under the specific service topic for services that require extra or different information.
- 3) Call the `.Connect` method
- 4) The login screen of the service is displayed in a separate browser window
- 5) After successful authentication, the `OnConnected` event is triggered and from that moment the component has access the cloud service API

Demo

A user friendly application to explore the supported services. Demonstrates how authentication, authorization and interacting with the API services works.

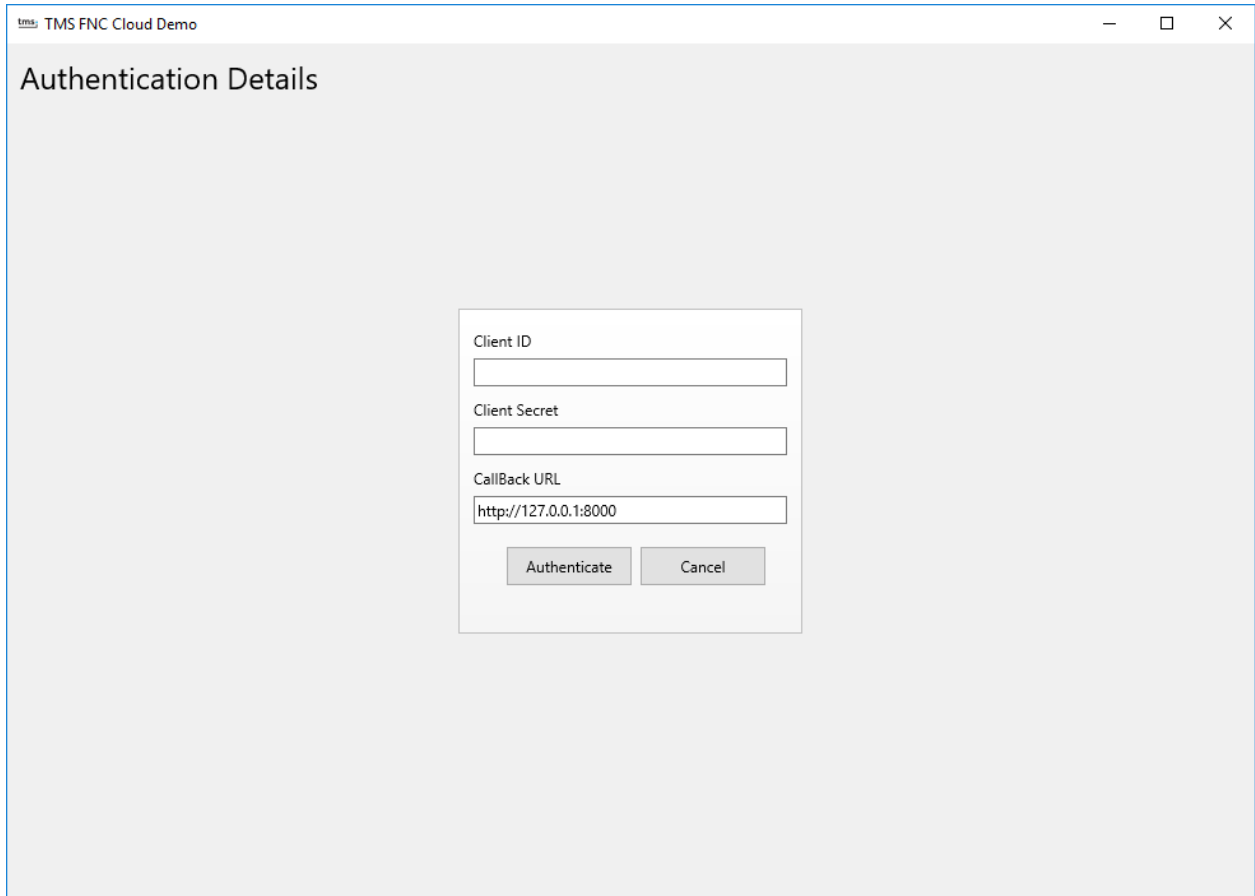
Usage

- Select the one of the services from the tiles menu

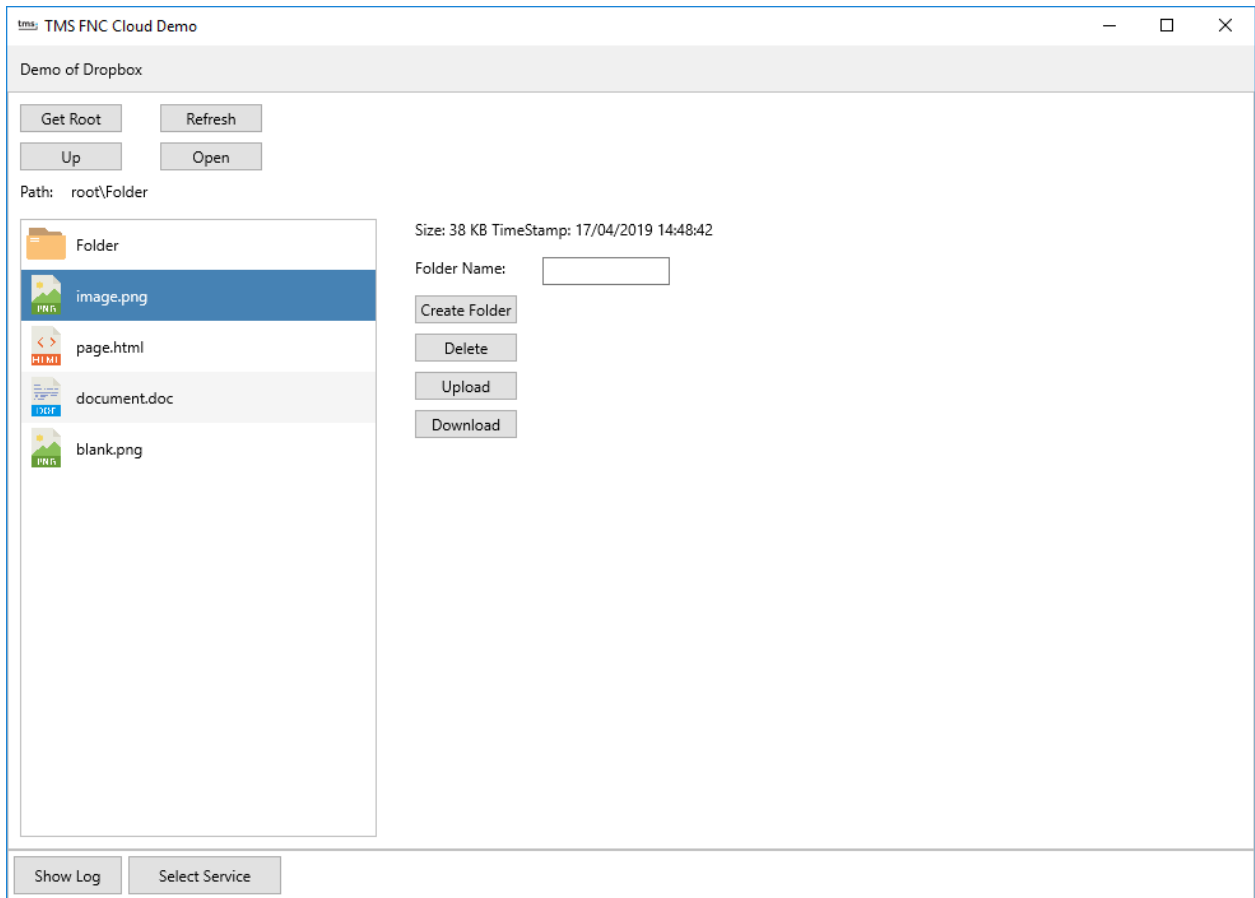


- If you're using the specific service for the first time, enter the required authentication details. Pressing the Authenticate button will bring up the login screen from the selected service. Once authenticated successfully, if the same service is selected again from the list, this screen is no longer displayed and the next screen is displayed directly.

Note: On a mobile platform, to avoid typing errors you can send an email to an account on the device and copy-paste the required values from there instead.



- The demo for the selected service is displayed and the API features can now be accessed



Services

An overview of all currently supported services:

TTMSFNCCloudCloudKit

Usage

A component that provides access to the Apple CloudKit service. It allows to read, create, update & delete data.

Authorization information

Client ID, Callback URL, Project

Organisation

Properties

Entities

A list of entities.

MetaData

A list of metadata.

Methods

Insert

Insert new data

Query

Retrieve existing data

Update

Update existing data

Delete

Delete existing data

TTMSFNCCloudDropBox

Usage

A component that provides access to the DropBox service. It allows to read, create and delete folders as well as read, upload, download and delete files.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

DDrive

A list of files and folders.

Info

Account info

Mode

Switch between Sandbox and Root mode

Methods

CreateFolder

Create a new folder

Delete

Delete a file or folder from the drive

Download
Download a file from the drive

GetAccountInfo
Retrieve the account info

GetFolderList
Retrieve the list of files and folders for the root folder or a specific folder

GetFolderListHierarchical
Retrieve the list of files and folders for specific folder and add them to the DDrive list

GetLink
Retrieve a direct link to a file

GetShare
Retrieve a share URL to a file or folder

Search
Search for files and folders based on a query

Upload
Upload a file to the drive

TTMSFNCCloudGoogleCalendar

Usage

A component that provides access to the Google Calendar service. It allows to read, create, update & delete Google Calendars and Google Calendar Events.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

Calendars
A list of Google Calendars.

Items
A list of Google Calendar Events.

Methods

GetCalendars
Fill the list of Calendars

AddCalendar

Add a new Calendar entry to the list of Google Calendars. The Summary, Description, Location and TimeZone properties are used to initialize the Calendar

DeleteCalendar

Delete an existing Google Calendar

UpdateCalendar

Update an existing Google Calendar. The Summary, Description, Location and TimeZone properties can be updated

GetCalendar

Fill the Items list with Calendar Events from a Google calendar for a certain timespan. If no ID is provided, the Events are retrieved from the default Google Calendar. If no FromDate/ToDate is specified the Events are retrieved for the default timespan. When ChangedSince is used, only Events that changed since the specified date/time are retrieved

GetColors

Fill the list of ItemColors and CalendarColors with the predefined values from the Google Calendar service

GetItemByID

Retrieve a single Event based on the ID of the Google Calendar and the Event ID

Add

Add a new Event to the Calendar as specified by Item.CalendarID

Update

Update an existing Event

Delete

Delete an existing Event

TTMSFNCCloudGoogleDrive

Usage

A component that provides access to the Google Drive service. It allows to read, create and delete folders as well as read, upload, download and delete files.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

GDrive

A list of files and folders.

Info
Account info

TTMSFNCCloudGoogleDriveItem.PublicShare
Indicate if the file is shared

TTMSFNCCloudGoogleDriveItem.WebContentLink
Retrieve a link to the file

Methods

CreateFolder
Create a new folder

Delete
Delete a file or folder from the drive

Download
Download a file from the drive

GetFolderList
Retrieve the list of files and folders for the root folder or a specific folder

GetFolderListHierarchical
Retrieve the list of files and folders for a specific folder and add them to the GDrive list

SearchFile
Search for files based on a query

SearchFolder
Search for folders based on a query

Upload
Upload a file to the drive

TTMSFNCCloudGoogleFirebaseObjectDatabase

Usage

A component that provides access to the Google Firebase Database service. It allows to read, create, update & delete data.

Authorization information

Client ID, Callback URL, Project

Organisation

Properties:

DatabaseName

Required to be assigned with the name of the Firebase database related to the authenticated user. Detailed information is available at: <http://www.tmssoftware.com/site/cloudkey.asp>

TableName

Assign the name of the table that is used to retrieve, add, update and remove with the methods indicated below.

GeneralRules

Contains the general rules specified for the Firebase Database. Use the GetRules method to fill the list with existing rules.

IndexRules

Contains the index rules specified for the Firebase Database. These are required when using one of the Query* methods. Use the GetRules method to fill the list with existing rules. Use the AddIndexRule method to add a rule.

Methods:

InsertObject

Insert a new object in the Table specified in the TableName property.

WriteObject

Update an existing object in the Table specified in the TableName property.

ReadObject

Retrieve an existing object based on it's ID from the Table specified in the TableName property.

DeleteObject

Delete an existing object from the Table specified in the TableName property.

InsertList

Insert a list of objects in the Table specified in the TableName property.

WriteList

Updates a list of objects in the Table specified in the TableName property.

ReadList

Retrieves a list of objects from the Table specified in the TableName property.

Deletelist

Deletes a list of objects from the Table specified in the TableName property.

QueryList(PropertyName, KeyWord: string)

QueryList(PropertyName, KeyWord: string; AList: TTMSFNCCloudGoogleFireBaseObjectList);

QueryList(PropertyName, KeyWord: boolean)

QueryList(PropertyName, KeyWord: boolean; AList: TTMSFNCCloudGoogleFireBaseObjectList);

QueryList(PropertyName, KeyWord: double)

QueryList(PropertyName, KeyWord: double; AList: TTMSFNCCloudGoogleFireBaseObjectList);

Retrieves a filtered list of objects from the Table specified in the TableName property. The list is filtered on the field specified in the PropertyName value based on the KeyWord value. The KeyWord can be a string, Boolean or integer/double value depending on the content of the field.

Note it is required that the PropertyName is present in the list of IndexRules. Use the AddIndexRule to add a new PropertyName.

QueryList(PropertyName, StartAt, EndAt: string)

QueryList(PropertyName, StartAt, EndAt: string; AList: TTMSFNCCloudGoogleFireBaseObjectList);

Retrieves a filtered list of objects from the Table specified in the TableName property. The list is filtered on the field specified in the PropertyName value based on the StartAt and EndAt values.

For example if the StartAt is "A" and the EndAt is "A" the result will contain a list of all objects for which the PropertyName field text starts with "A".

Note it is required that the PropertyName is present in the list of IndexRules. Use the AddIndexRule to add a new PropertyName.

QueryList(PropertyName, MinValue, MaxValue: string)

QueryList(PropertyName, MinValue, MaxValue: string; AList:

TTMSFNCCloudGoogleFireBaseObjectList);

Retrieves a filtered list of objects from the Table specified in the TableName property. The list is filtered on the field specified in the PropertyName value based on the MinValue and MaxValue values.

For example if the MinValue is "0" and the EndAt value is "100" the result will contain a list of all objects for which the PropertyName field data is higher or equal to "0" and lower or equal to "100".

Note it is required that the PropertyName is present in the list of IndexRules. Use the AddIndexRule to add a new PropertyName.

AddIndexRule

Add a new rule to the list of IndexRules and submit the new data to the Firebase Database rules.

If the rule based on the TableName and PropertyName already exists, no data is changed.

When using one of the Query* methods it is required that the PropertyName (= Fieldname) has been added to the list of index rules.

DeleteTable

Delete all objects and table object itself specified in the ATableName parameter value.

IDExists

Returns if an object with the specified ID parameter value exists in the Table specified with the TableName property value.

SetRules

Submits the data from the GeneralRules and IndexRules to the Firebase Database.

TTMSFNCCloudGoogleGMail

Usage

A component that provides access to the Google GMail service. It allows to retrieve and send email messages in HTML or plain text format. File attachments can be included when sending emails.

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

Files

A list of files to be included as attachment

Labes

A list of available labels.

Mails

A list of emails.

Methods

GetLabels

Retrieves all available labels

GetMails

Retrieves all emails with the specified Label

SendMessage

Sends an email

UpdateMessageLabels

Updates the labels for a specific email

TTMSFNCCloudGoogleLookup

Usage

A component that that can perform lookup on a partial string for the most frequently used Google search terms.

Note: This component is not supported in a TMS WEB Core framework running on localhost due to technical limitations of the API

Authorization information

Key

Organisation

Methods

Lookup

Returns a list of possible matches to the lookup query

TTMSFNCCloudGoogleLocationLookup

Usage

A component that that can perform can perform location lookup based on Google Maps location data.

Note: This component is not supported in a TMS WEB Core framework running on localhost due to technical limitations of the API

Authorization information

Key

Organisation

Methods

GetGeoLocation

Returns a list of possible geo coordinates matching the lookup query

LocationLookup

Returns a list of possible matches to the lookup query

TTMSFNCCloudGooglePeople

Usage

A component that provides access to the Google People service. It allows to read, create, update & delete Groups and Contacts.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

Contacts

A list of Google Contacts

Groups

A list of Google Groups

Methods

AddContactToGroup

Add an existing contact to a Group

GetContacts

Fill the list of Contacts

GetContactByID

Retrieve a specific Contact by ID

GetGroups
Fill the list of Groups

CreateContact
Create a new Contact

CreateGroup
Create a new Group

DeleteContact
Delete an existing Contact

DeleteGroup
Delete an existing Group

DeleteContactFromGroup
Remove an existing contact from a Group

UpdateContact
Update an existing Contact

UpdateGroup
Update an existing Group

TTMSFNCCloudGooglePhotos

Usage

A component that provides access to the Google Photos service. It allows to read, upload, download image files and read, create Albums.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

Albums
A list of Google Photos Albums

SearchResults
A list of Google Photos

Methods

AddFolderToAlbum
Upload all image files from a local folder to the specified Album

CreateAlbum
Create a new Album

DownloadPhoto
Download an existing Photo

FindAlbumByTitle
Find an Album by the specified Title

GetAlbums
Fills the list of Albums

SearchPhotos
Fill the list of SearchResults. Retrieve all available Photos or only Photos from the specified Album

UploadPhoto
Upload a new Photo to the specified Album

TTMSFNCCloudGoogleTasks

Usage

A component that provides access to the Google Tasks service. It allows to read, create, update & delete Google Tasks and Google TaskLists.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

TaskLists
A list of Google TaskLists

TaskLists.Tasks
A list of Google Tasks

HasNextTaskLists
Indicates if more TaskLists are available that can be retrieved with GetNextTaskLists

HasNextTasks
Indicates if more Taskss are available that can be retrieved with GetNextTasks

Methods

GetTaskLists
Fill the list of TaskLists. The request returns a limited number of TaskLists. The HasNextPage property indicates if more TaskLists are available, these can be retrieved with GetNextTaskLists

GetNextTaskLists

Retrieve the remaining TaskLists. The HasNextTaskLists property indicates if more TaskLists are available, these can be retrieved by calling GetNextTaskLists again

GetTasks

Fill the list of Tasks for the specified TaskList. The HasNextTasks property indicates if more Tasks are available, these can be retrieved with GetNextTasks

GetNextTasks

Retrieve the remaining Tasks for the specified TaskList. The HasNextTasks property indicates if more Tasks are available, these can be retrieved by calling GetNextTasks again

AddTaskList

Create a new TaskList

AddTaskToList

Add an existing Task the the specified TaskList

UpdateTaskList

Update an existing TaskList

UpdateTask

Update an existing Task

DeleteTaskList

Delete an existing TaskList

DeleteTask

Delete an existing Task

TTMSFNCCloudGoogleYouTube

Usage

A component that provides access to the Google YouTube service. It allows to read and upload videos.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

Videos

A list of YouTube Videos

CommentThreads

A list of YouTube comments

Methods

DeleteVideo

Delete an existing video

GetVideoDetails

Retrieve details for the specified Video

GetVideoRating

Retrieve the rating for the specified Video

SetVideoRating

Set the rating for the specified Video

GetAllVideos

Fill the list of Videos

GetFirstVideos

Fill the list of Videos. If more Videos are available than the specified maximum number, these can be retrieved with GetNextVideos

GetNextVideos

Fill the list of Videos with the remaining Videos after using GetFirstVideos

GetAllCommentThreads

Fill the list of CommentThreads for the specified Video

GetLatestCommentThreads

Fill the list of CommentThreads with the newest comments

GetNextCommentThreads

Fill the list of CommentThreads with the remaining comments after using GetLatestCommentThreads

UploadVideo

Upload a new Video

UpdateVideo

Update the Title, Description and CategoryID of an existing Video

TTMSFNCCloudHubic

Usage

A component that provides access to the Google Drive service. It allows to read, create and delete folders as well as read, upload, download and delete files.

Note: This component is not supported in a TMS WEB Core framework running on localhost due to technical limitations of the API

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

HubicDrive

A list of files and folders.

Info

Account info

Methods

CreateFolder

Create a new folder

Delete

Delete a file or folder from the drive

Download

Download a file from the drive

GetAccountInfo

Retrieve the account info

GetFolderList

Retrieve the full list of files and folders

GetLink

Retrieve a direct link to a file

GetShare

Retrieve a share URL to a file or folder

Upload

Upload a file to the drive

TTMSFNCCloudMicrosoftOneDrive

Usage

A component that provides access to the Microsoft OneDrive / Microsoft OneDrive for Business service. It allows to read, create and delete folders as well as read, upload, download and delete files.

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

OneDrive

A list of files and folders.

Info

Account info

Methods

CreateFolder

Create a new folder

Delete

Delete a file or folder from the drive

Download

Download a file from the drive

GetAccountInfo

Retrieve the account info

GetFolderList

Retrieve the list of files and folders for the root folder or a specific folder

GetFolderListHierarchical

Retrieve the list of files and folders for specific folder and add them to the OneDrive list

GetShare

Retrieve a share URL to a file or folder

Search

Search for files based on a query

Upload

Upload a file to the drive

TTMSFNCCloudMicrosoftOutlookCalendar

Usage

A component that provides access to the Microsoft Outlook Calendar service. It allows to read, create, update & delete Outlook Calendars and Outlook Calendar Events.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

Calendars

A list of Outlook Calendars.

Items

A list of Outlook Events.

Methods

GetCalendars

Fill the list of Calendars

AddCalendar

Add a new Calendar entry to the list of Google Calendars

DeleteCalendar

Delete an existing Calendar

UpdateCalendar

Update an existing Calendar

GetCalendar

Fill the Items list with calendar events from a Calendar for a certain timespan. If no ID is provided, the events are retrieved from the default Calendar. If no FromDate/ToDate is specified the events are retrieved for the default timespan

GetItemByID

Retrieve a single Event based on the ID of the Calendar and the Event ID

Add

Add a new Event to the calendar as specified by Item.CalendarID

UpdateItem

Update an existing Event

Delete

Delete an existing Event

TTMSFNCCloudMicrosoftOutlookContacts

Usage

A component that provides access to the Microsoft Outlook Contacts service. It allows to read, create, update & delete Contacts.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

Properties

Contacts

A list of Google Contacts

Methods

GetContacts

Fill the list of Contacts

GetContactByID

Retrieve a specific Contact by ID

CreateContact

Create a new Contact

DeleteContact

Delete an existing Contact

UpdateContact

Update an existing Contact

TTMSFNCCloudMicrosoftOutlookMail

Usage

A component that provides access to the Outlook Mail service. It allows to retrieve and send email messages. File attachments can also be included when sending emails.

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

Folders

A list of Outlook Folders

Mails

A list of Outlook Mail messages

Methods

GetFolders

Fill the list of Folders

GetMails

Fill the list of Mails. Optionally a FolderID can be specified to only retrieve items from a specific folder. By default only the items from the 'Inbox' folder are retrieved. If no FolderID is specified the items for all folders are returned. For specific system folders the name of the folder can be used as FolderID, this includes 'Inbox', 'Drafts', 'SentItems' and 'DeletedItems'. Optionally a PageSize and PageIndex can be provided, if not, the first 100 items are returned

SendMessage

Send an email message. At least one Recipient email address is required, the other parameters are optional

TTMSFNCCloudMyCloudData

Usage

A component that provides seamless access to the myCloudData.net service that allows to create tables with meta data of choice to store data in the cloud. A user can have access to one or more tables. After login, the collection of tables available to the user is returned with GetTables and accessible via the Tables collection. A table on myCloudData.net is represented by the class TTMSFNCCloudMyCloudDataTable. A table has metadata, entities, filters, sort order and shares. The table entities are represented by the class TTMSFNCCloudMyCloudDataEntity. The shares are represented by the class TTMSFNCCloudMyCloudDataShare. The metadata for a table is retrieved via Table.GetMetaData. The entities are retrieved via the Table.Query function and the list of shares is retrieved with Table.Shares. The filter is accessible via the collection Table.Filters and the sort ordering can be setup via the collection Table.SortOrder.

Authorization information

Client ID, Client Secret, Callback URL

Organisation

TTMSFNCCloudMyCloudDataEntity

The TTMSFNCCloudMyCloudDataEntity class is the class that wraps an entity (in database terminology also often referred to as record). The TTMSFNCCloudMyCloudDataEntity class has following methods & properties:

Properties

Value[AName: string]: TValue;
Blob[AName: string]: TTMSFNCCloudMyCloudDataBlob;

Methods

Update;
Insert;
Delete;

To get or set a value for a field within the entity, you can use Value[AName: string].

TTMSFNCCloudMyCloudDataEntities

This is the collection of entities retrieved via Table.Query.
Typical operations on entities are as such:

1. Create a new entity in the cloud storage:

```
var
  ent: TTMSFNCCloudMyCloudDataEntity;
begin
  ent := Table.Entities.Add;
  ent.Value['NAME'] := 'Elon Musk';
  ent.Value['STATE'] := 'California';
  ent.Value['COMPANY'] := 'Tesla';
  ent.Insert;
end;
```

2. Update an existing entity in the cloud storage:

```
var
  ent: TTMSFNCCloudMyCloudDataEntity;
begin
  ent := Table.Entities[x];
  ent.Value['COMPANY'] := 'SpaceX';
  ent.Update;
end;
```

3. Delete an entity permanently from the cloud storage:

```
var
  ent: TTMSFNCCloudMyCloudDataEntity;
begin
  ent := Table.Entities[y];
  ent.Delete;
end;
```

TTMSFNCCloudMyCloudDataBlob

The TTMSFNCCloudMyCloudDataBlob class is a wrapper class for binary data stored in a blob in the cloud service. Note that the blob storage capability is not available for a free account in myCloudData.net but requires a subscription.

The TTMSFNCCloudMyCloudDataBlob class has following properties and methods:

Properties

Table: TTMSFNCCloudMyCloudDataTable
Table to which the blob belongs

Entity: TTMSFNCCloudMyCloudDataEntity
Entity to which the blob belongs

Field: string
Name of the field holding the blob

Methods

LoadFromFile(AFileName: string);
Load data from AFileName into the blob field.

SaveToFile(AFileName: string);
Get data from the blob field and save it to a file.

LoadFromStream(AStream: TStream);
Load data from the stream into the blob field.

SaveToStream(AStream: TStream);
Get data from the blob field and save it to a stream.

A typical operation to store some binary data into a blob field in a new entity would be:

Example:

```
var
  ent: TTMSFNCCloudMyCloudDataEntity;
begin
  blob: TTMSFNCCloudMyCloudDataBlob;
  ent := Table.Entities.Add;
  blob := ent.Blob['BIN'];
  blob.LoadFromFile('mybinfile.bin');
end;
```

Note that for performance reasons, blobs are returned via the entity only and retrieved from the cloud storage at the time SaveToStream() or SaveToFile() is executed.

TTMSFNCCloudMyCloudDataLookupFieldValue

The TTMSFNCCloudMyCloudDataLookupFieldValue class contains the key and lookup values associated with a LookupField.

Properties

KeyValue: Variant
The key value of the lookupdata

LookupValue: Variant
The lookup value of the lookupdata

Tag: integer
Defines a custom related value

TTMSFNCCloudMyCloudDataLookupFieldValues

This is the collection of lookup field values from a LookupField.

Properties

LookupValues[AKey: Variant]: Variant;

TTMSFNCCloudMyCloudDataLookupField

The TTMSFNCCloudMyCloudDataLookupField class contains the lookupdata for a LookupField.

Properties

LookupField: string

The name of the field in the parent table that the lookup data is associated with

TableID: Int64

The ID of the table that contains the lookup data

LookupFieldValues: TMyCloudDataLookupFieldValues

The list of key and lookup values

TTMSFNCCloudMyCloudDataLookupFieldList

This is the collection of lookup fields retrieved via Table.GetLookupData.

Properties

Field[AFieldName: string]: TTMSFNCCloudMyCloudDataLookupFieldValues;
property List[AFieldName: string]: TStringList;

TTMSFNCCloudMyCloudDataTable

This class represents the table in the cloud storage and is part of the set of tables in the collection Tables retrieved with GetTables.

A TTMSFNCCloudMyCloudDataTable class has following properties and methods:

Properties

ID: int64

Read-only property returning the unique identifier of the table

OwnerID: int64

Read-only property returning the unique owner identifier of the table

IsOwner: boolean

Read-only property returns true when the logged in user owns the table

Name: string

Gets or sets the name of the table

Permissions: TTMSFNCCloudMyCloudDataPermissions

Permissions the user has on the table. Permissions are: CRUD, i.e. create/read/update/delete

MetaData: TMyCloudDataMetaData

Access to the metadata of the table via a collection after calling GetMetaData

Entities: TMyCloudDataEntities

Access to the entities of the table via a collection after calling Query

Filters: TMyCloudDataFilters

Access to the filter conditions for a query via a collection

SortOrder: TMyCloudDataSortOrderList

Access to the sort order settings for a query via a collection

LookupFieldList: TMyCloudDataLookupFieldList

Access to the list of lookup fields via a collection after calling GetLookupData.

Methods

GetMetaData: boolean

Retrieves the metadata from a table, stored in MetaData

SetMetaData: boolean

Updates the metadata of the table on the cloud storage with Table.MetaData

Query

Simply query for all entities of the table, stored in Entities

Query(Fields: TStringArray)

Query with specifier of selection of fields to return, stored in Entities

Query(AFields: TStringList)

Query with specifier of selection of fields to return, stored in Entities

Share(Email: string; Permissions: TMyCloudDataPermissions)

Share a table with another user defined by email

RemoveShare(Email: string)

Remove an existing share with another user via email

Delete

Delete the table from the cloud storage

GetShares: TTMSFNCCloudMyCloudDataShares

Retrieve the list of email addresses & permissions with who the table was shared

GetLookupData: Boolean

Retrieves the lookup data from a table, stored in LookupFieldList. It is required that the Table metadata contains valid values for the LookupTable, LookupField and LookupKeyField properties in order to be able to retrieve lookup data.

To perform a simple query, use:

```
Table.Query;
```

This will fill the Entities collection with the entities retrieved from the cloud storage.

```
Table.Query(['NAME','COMPANY','BIRTHDATE'];
```

This will fill the Entities collection but the entities will only hold the fields NAME, COMPANY, BIRTHDATE

To filter data, following code can be used:

```
var
  filter: TMSFNCCloudMyCloudDataFilter;
begin
  Table.Filters.Clear; // removes all filter conditions
  filter := Table.Filters.Add('NAME', coLike, 'Musk', loNone);
  Table.Query;
end;
```

Or alternatively:

```
var
  filter: TMSFNCCloudMyCloudDataFilter;
begin
  Table.Filters.Clear; // removes all filter conditions
  filter := Table.Filters.Add;
  filter.FieldName := 'NAME';
  filter.ComparisonOperator := coLike;
  filter.Value := 'Musk';
  filter.LogicalOperator := loNone;
  Table.Query;
end;
```

Note that the ComparisonOperator can be any of the following values:

coEqual, coNotEqual, coLike, coGreater, coGreaterOrEqual, coLess, coLessOrEqual, coStartsWith, coEndsWith, coNull, coNotNull;

The LogicalOperator that sets the logical operation between two sequential filter conditions can be: loAND, loOR, loNone

To specify the sort order for a query, the Table.SortOrder collection can be used:

```
Table.SortOrder.Clear;
Table.SortOrder.Add('NAME', soAscending);
Table.SortOrder.Add('COMPANY', soDescending);
Table.Query;
```

Or alternatively:

```
var
  sortorder: TTMSFNCCloudMyCloudDataSortOrderItem;
begin
  Table.SortOrder.Clear;
  sortorder := Table.SortOrder.Add;
  sortorder.FieldName := 'NAME';
  sortorder.SortOrder := soAscending;
  sortorder := Table.SortOrder.Add;
  sortorder.FieldName := 'COMPANY';
  sortorder.SortOrder := soDescending;
  Table.Query;
end;
```

TTMSFNCCloudMyCloudDataTables

This is the collection of all the tables a user has access to, either because the user owns the table or the table was shared with the user.

TTMSFNCCloudMyCloudDataMetaDataItem

This class holds the information about a single meta data item in the meta data collection of a table. The meta data item class has following properties:

Properties

General Properties

These properties define the behavior of the MetaDataItem on the myCloudData server as well as in a client application.

PropertyName: string
Gets or sets the field name

DataType: TFieldType
Gets or sets the field type. The field type can be any of following value: ftString, ftWideString, ftInt, ftBigInt, ftFloat, ftBlob, ftSmallInt, ftWord, ftBoolean, ftDate, ftDateTime, ftTime

Data: Boolean
Returns true when the meta data item pertains actual data

Size: integer
Optionally gets or sets the size of a field (only available for fields of type ftString, ftWideString)

Client Properties

These properties can define the appearance of the MetaDataItem and how it behaves in a client application. These properties have no influence on the behavior of the MetaDataItem on the myCloudData server.

LabelText: string
Gets or sets the label text associated with the field

DefaultValue: string
Gets or sets the default value associated with the field

Width: integer
Gets or sets the width associated with the field

Order: integer
Gets or sets the order index associated with the field (relative to the other fields in the table)

Mask: string
Gets or sets the field content mask

Minimum: double

Gets or sets the minimum allowed value associated with the field

Maximum: double

Gets or sets the maximum allowed value associated with the field

MinimumDate: tdatetime

Gets or sets the minimum allowed date and/or time associated with the field

MaximumDate: tdatetime

Gets or sets the maximum allowed date and/or time associated with the field

Visible: Boolean

Gets or sets the visibility associated with the field. Default is true

Description: string

Gets or sets the description value associated with the field

Enabled: Boolean

Gets or sets the enabled status associated with the field. Default is true

Required: Boolean

Gets or sets the required status associated with the field. Default is false

LookupTable: int64

Gets or sets the ID of the lookup table associated with the field

LookupField: string

Gets or sets the lookup value field from the LookupTable associated with the field. The value should be identical to one of the field names of the table defined in LookupTable and should be different from the LookupKeyField value.

LookupKeyField: string

Gets or sets the lookup key field from the LookupTable associated with the field. The value should be identical to one of the field names of the table defined in LookupTable and value should be different from the LookupField value.

The meta data item is part of the meta data collection TTMSFNCCloudMyCloudDataMetaData accessible via Table.MetaData.

Typical operations on the meta data are:

1. Retrieval of meta data & list all fields in a listbox

```
var
    i: integer;
begin
    Table.GetMetaData;
    for i := 0 to Table.MetaData.Count - 1 do
        begin
            listbox.Items.Add(Table.MetaData[i].PropertyName);
        end;
    end;
```

2. Creating meta data for a new table

```
Table.Metadata.Clear;
Table.Metadata.Add('NAME', ftWideString, 50);
Table.Metadata.Add('COMPANY', ftWideString, 50);
Table.Metadata.Add('BIN', ftBlob);
Table.SetMetaData;
```

Or alternatively

```
var
  metadata: TTMSFNCCloudMyCloudDataMetaDataItem;
begin
  Table.Metadata.Clear;
  metadata := Table.Metadata.Add;
  metadata.PropertyName := 'NAME';
  metadata.DataType := ftWideString;
  metadata.Size := 50;
  metadata := Table.Metadata.Add;
  metadata.PropertyName := 'COMPANY';
  metadata.DataType := ftWideString;
  metadata.Size := 50;
  metadata := Table.Metadata.Add;
  metadata.PropertyName := 'BIN';
  metadata.DataType := ftBlob;
  Table.SetMetaData;
end;
```

TTMSFNCCloudMyCloudDataShares

To share a table with another myCloudData.net user, call:

```
Table.Share('myfriend@company.com', [pCreate, pRead, pUpdate]);
```

This adds a share with user myfriend@company.com. Note that it is required that myfriend@company.com is recognized as a valid myCloudData.net user. The myCloudData.net API will not send a notification of the share itself. It is the responsibility of the user to do so. Here a share is created with all permissions except the permission to delete entities in the table.

To remove the share at a later time, call:

```
Table.RemoveShare('myfriend@company.com');
```

When the share existed for the user it will be removed.
To see with who a table is shared, use:

```
var
  shares: TTMSFNCCloudMyCloudDataShares;
begin
  shares := Table.GetShares;

  for i := 0 to shares.Count - 1 do
  begin
    listbox.Items.Add(shares[i].Email);
  end;
end;
```

TTMSFNCCloudMyCloudData

TTMSFNCCloudMyCloudData is the class that wraps the entire access to the myCloudData.net service.

Methods & properties available in TTMSFNCCloudMyCloudData:

Properties

TableId: int64

Gets or sets the unique ID of the table TTMSFNCCloudAdvMyCloudData can work on

Methods

Table related methods:

GetTables

Retrieves the list of tables and makes these accessible via TAdvMyCloudData.Tables

TableByName(AName: string)

Retrieves on table based on its name

TableList

Retrieves the list of available tables as stringlist

AddTable(ATable: TTMSFNCCloudMyCloudDataTable)

Creates a new table from an existing TTMSFNCCloudMyCloudDataTable class

CreateTable(ATableName: string)

Creates a new table with name ATableName and returns an instance to the table

DeleteTable(AID: int64)

Deletes a table based on its unique ID

DeleteTable(ATable: TTMSFNCCloudMyCloudDataTable)

Deletes a table based on an existing TMyCloudDataTable class

UpdateTable(ATable: TTMSFNCCloudMyCloudDataTable)

Updates table info, such as name, permissions based on an existing TMyCloudDataTable class

ShareTable(ATable: TTMSFNCCloudMyCloudDataTable; AEmail: string; APermissions: string)

Share a table with specific permissions with another myCloudData.net user

GetTableShares(ATable: TMyCloudDataTable)

Fills the Table.Shares collection with shares found

Metadata related methods:

GetMetaData

Retrieves the metadata for the table specified by TTMSFNCCloudMyCloudData.TableId

AddMetaData(AMetaData: TTMSFNCCloudMyCloudDataMetaDataType)

Sets the metadata for table specified by TTMSFNCCloudAdvMyCloudData.TableId

UpdateMetaData(AOldFieldName, ANewFieldName: string; ADataType: TFieldType = ftUnknown; ASize: integer = -1)

Modifies the meta data of a single field (specified by ANewFieldName and limited to to the field name, field datatype and field size) for a table specified by TTMSFNCCloudAdvMyCloudData.TableId

UpdateMetaData(AMetaData: TTMSFNCCloudMyCloudDataMetaDataType)

Modifies all the meta data of a single field (specified by AMetaData) for a table specified by TTMSFNCCloudAdvMyCloudData.TableId.

DeleteMetaData(APropertyName: string)

Delete a field from the meta data for a table specified by TTMSFNCCloudAdvMyCloudData.TableId

Entity related methods:

Insert(AValues: TStringList): TTMSFNCCloudDataStoreEntity

Inserts entity values via a stringlist

Query

Retrieves entities for a table specified by TTMSFNCCloudAdvMyCloudData.TableId

Query(AFields: TStringList)

Retrieves entities with fields limited to the specified list for a table specified by TTMSFNCCloudAdvMyCloudData.TableId

Query(AFields: TStringList; AFilters: TTMSFNCCloudMyCloudDataFilters)

Retrieves entities with filter conditions

Query(AFields: TStringList; ASortOrder: TTMSFNCCloudMyCloudDataSortOrderList)

Retrieves entities with sort order specified

Query(AFields: TStringList; AFilters: TTMSFNCCloudMyCloudDataFilters; ASortOrder: TTMSFNCCloudMyCloudDataSortOrderList)

Retrieves entities with filter conditions and sort order specified

Delete(AID: string)

Delete an entity with ID from a table specified by TTMSFNCCloudAdvMyCloudData.TableId

Delete(AIDList: TStringList)

Delete multiple entities from a table specified by TTMSFNCCloudAdvMyCloudData.TableId. Only entities with an ID specified in AIDList will be deleted.

DeleteAll

Delete all entities from a table specified by TTMSFNCCloudAdvMyCloudData.TableId

Update(AEntity: TDataStoreEntity)

Update the entity in a table specified by TTMSFNCCloudAdvMyCloudData.TableId

Update (AIDList, AFieldValues: TStringList)

Update multiple entities in a table specified by TTMSFNCCloudAdvMyCloudData.TableId. Only entities with an ID specified in AIDList will be updated. Only fields specified in AFieldValues will be updated.

Example:

```
var
  slIDs, slFields: TStringList;
begin
  slIDs := TStringList.Create;
  slIDs.Add('1');
  slIDs.Add('2');

  slFields := TStringList.Create;
  slFields.CommaText := 'FieldName=FieldValue';

  AdvMyCloudData1.Update(slIDs, slFields);
  slIDs.Free;
  slFields.Free;
```

UpdateAll(AFieldValues: TStringList)

Update all entities in a table specified by TTMSFNCCloudMyCloudData.TableId. Only fields specified in AFieldValues will be updated.

Download(ATableID, AEntityId: Int64; AFieldName: string; const TargetFile: string)

Download a blob to a file value from a specific entity in a specific table and fieldname

Download(ATableID, AEntityId: Int64; AFieldName: string; AStream: TStream)

Download a blob value in a stream from a specific entity in a specific table and fieldname

Upload(ATableID, AEntityId: Int64; AFieldName: string; FileName: string)

Upload a file to a blob field from a specific entity in a specific table and fieldname

Upload(ATableID, AEntityId: Int64; AFieldName: string; AStream: TStream)

Upload a stream to a blob field from a specific entity in a specific table and fieldname

Users related methods:

GetUser

Gets information about the currently authenticated user. Fills the User properties.

GetUsers

Get list of users, for non-admin users, this retrieves the logged in user. Fills the Users collection property

AddUser(AUser: TMyCloudDataUser)

Available for admin level users only

DeleteUser(AID: int64)

Available for admin level users only

Additional properties:

User: TTMSFNCCloudMyCloudDataUser

Contains information about the currently authenticated user

Users: TTMSFNCCloudMyCloudDataUsers

Collection of users, filled by the GetUsers method

Tables: TTMSFNCCloudMyCloudDataTables
Collection of users, filled by the GetTables method

PageIndex: integer

When > 0, this specifies the page of entities to retrieve for the Query() methods
property PageSize: integer;

When > 0, specifies the maximum number of entities to return for the Query() methods.
The last page is retrieved when number of entities is smaller than PageSize.

TTMSFNCCloudPayPal

Usage

A component that provides access to the PayPal payment service. It allows users to make payments using an existing PayPal account and for an easy way to integrate a PayPal based payments service in applications.

The component supports using the Sandbox (development) environment as well as the Live environment.

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

APIEnvironment

Switch between the Sandbox (development) and the Live API environment

Payment

Contains the payment information

PaymentError

Contains the payment error information

Transaction

Contains the transaction information

Methods

DoPayment

Execute a PayPal payment based on the Transaction configuration. During payment execution the user will be asked to authorize the payment using PayPal credentials.

If the user cancels the authorization the OnPaymentCancelled event is triggered.

If the payment succeeds the Payment class properties are assigned with the payment results and the OnPaymentAccepted event is triggered.

If an error occurred during the payment execution, the PaymentError class properties are assigned and the OnPaymentFailed is triggered.

Events

OnPaymentAccepted

Event fired when a Payment was successfully authorized and executed. Payment details are available in the Payment class properties.

OnPaymentCancelled

Event fired when a Payment was cancelled by the user.

OnPaymentFailed

Event fired when an error occurred during execution of the Payment. Error details are available in the PaymentError class properties.

TTMSFNCCloudSlack

Usage

A component that provides access to the Slack service. It allows users to extend and expand the capabilities of your Slack workspace. Cultivate conversation, customize functionality, or integrate ingenious services by building an app.

Authorization information

Client ID, Client Secret, CallBack URL

(The CallBack URL must also be set in the settings on the website.)

Organisation

TTMSFNCCloudSlack

Properties

CurrentUserName: *String*

The username of the user that has authenticated.

CurrentUserID: *String*

The ID of the user that has authenticated.

Users: *TTMSFNCCloudSlackUsers [TTMSFNCCloudSlackUser]*

Contains the members of your Slack team.

Every user has an Profile object of the type TTMSFNCCloudSlackProfile

Conversations: *TTMSFNCCloudSlackConversations [TTMSFNCCloudSlackConversation]*

Contains the different public and private channels, direct messages and group direct messages.

Every conversation has an object list Messages of the type TTMSFNCCloudSlackMessages that contains different TTMSFNCCloudSlackMessage objects these are the messages posted in the different channels. These in turn have the possibility to also contain 'Replies' and 'Files'.

ProfileImageSize: *TTMSFNCCloudSlackProfileImageSizes default sizOriginal*

You can select on which size the profile pictures get downloaded. By default this is set to sizOriginal which gets the original size. Other possible values are: siz24, siz32, siz48, siz72, siz192, siz512.

Methods

GetConversations/GetAllConversations

Retrieve (all) the conversations in the team. The procedure fills the Conversations (*TTMSFNCCloudSlackConversations*) property.

GetUsers/GetAllUsers

Retrieve (all) the users in the team. The procedure fills the Users (*TTMSFNCCloudSlackUsers*) property.

GetUserInformation

Retrieve the information for the requested user in the team. The procedure fills the Users (*TTMSFNCCloudSlackUsers*) property with the chosen user if it doesn't exist. The required parameter for the method is the UserID (*String*).

GetConversationHistory/GetAllConversationHistory

Retrieve the messages for the requested user in the team. The procedure fills the Messages (*TTMSFNCCloudSlackMessages*) property of the chosen conversation. The required parameter for the method is the Conversation (*TTMSFNCCloudSlackConversation*) or the ConversationID (*String*).

UploadFile

Uploads a file to your slack. The procedure creates a slack file object (*TTMSFNCCloudSlackFile*). The required parameter for the method is the FNC file (*TTMSFNCCloudSlackFile*).

PostMessage

Adds a message to the selected conversation. The procedure adds the create message to the Messages (*TTMSFNCCloudSlackMessages*) property of the chosen conversation. The required parameter for the method is the Conversation (*TTMSFNCCloudSlackConversation*) or the ConversationID (*String*) and the text of the message (*String*).

JoinConversation

Tries to add the authenticated user to a conversation. The required parameter for the method is the Conversation (*TTMSFNCCloudSlackConversation*) or the ConversationID (*String*).

GetUserConversations/GetAllUserConversations

Retrieve (all) the conversations for a specific user. The procedure fills the Conversations (*TTMSFNCCloudSlackConversations*) property. The required parameter for the method is the User (*TTMSFNCCloudSlackUser*) UserID (*String*).

DeleteMessage

Removes a message from the selected conversation. The procedure removes the message from the Messages (*TTMSFNCCloudSlackMessages*) property of the chosen conversation. The required parameter for the method is the message (*TTMSFNCCloudSlackMessage*).

GetMessageReplies/ GetAllMessageReplies

Retrieve (all) the replies from a specific message. The procedure fills the Replies (*TTMSFNCCloudSlackReplies*) property from a message (*TTMSFNCCloudSlackMessage*). The required parameter for the method is the Message (*TTMSFNCCloudSlackMessage*).

DownloadFile

Downloads the content of a slack file. The procedure fills the stream (*TMemoryStream*) property from a file (*TTMSFNCCloudSlackFile*). The required parameter for the method is the file (*TTMSFNCCloudSlackFile*).

PostReply

Adds a reply to the selected message thread. The procedure adds the created reply to the Replies (*TTMSFNCCloudSlackReplies*) property of the chosen message. The required parameter for the method is the Message (*TTMSFNCCloudSlackMessage*) and the text of the reply (*String*).

DownloadProfilePicture

Downloads the content of a slack profile picture. The procedure fills the profilepicturestream (*TMemoryStream*) property from a file (*TTMSFNCCloudSlackFile*). The required parameter for the method is the User (*TTMSFNCCloudSlackUser*), the UserID (*String*) or the user profile (*TTMSFNCCloudSlackUserProfile*).

Events

OnGetConversations/OnGetAllConversations

Event fired when the/all requested conversations are retrieved. This event returns additional to the sender and the request result also the list of conversations (*TTMSFNCCloudSlackConversations*) and the possible ErrorMessage (*String*).

OnGetConversationHistory/ OnGetAllConversationHistory

Event fired when the/all requested messages of the chosen conversation are retrieved. This event returns additional to the sender and the request result also the chosen conversations (*TTMSFNCCloudSlackConversation*), the requested messages (*TTMSFNCCloudSlackMessages*) and the possible ErrorMessage (*String*).

OnGetUsers/OnGetAllUsers

Event fired when the/all requested users are retrieved. This event returns additional to the sender and the request result also the list of users (*TTMSFNCCloudSlackUsers*) and the possible ErrorMessage (*String*).

OnGetFile

Event fired when the requested file is retrieved. This event returns additional to the sender and the request result also the message of the file (*TTMSFNCCloudSlackMessage*), the requested file (*TTMSFNCCloudSlackFile*), the stream of the file (*TMemoryStream*) and the possible ErrorMessage (*String*).

OnPostMessage

Event fired when the created message is added to the conversation. This event returns additional to the sender and the request result also the conversation in which the message was added (*TTMSFNCCloudSlackConversation*), the message itself (*TTMSFNCCloudSlackMessage*) and the possible ErrorMessage (*String*).

OnUploadFile

Event fired when the file is uploaded to the chosen conversations. This event returns additional to the sender and the request result also the file itself (*TTMSFNCCloudSlackFile*) and the possible ErrorMessage (*String*).

OnJoinConversation

Event fired when a user is added as a member of a channel. This event returns additional to the sender and the request result also a Boolean if the user was added or not, the requested conversation (*TTMSFNCCloudSlackConversation*), a possible WarningMessage (*String*) and the possible ErrorMessage (*String*).

OnGetProfilePicture

Event fired when the picture of the requested profile is retrieved. This event returns additional to the sender and the request result also the user (*TTMSFNCCloudSlackUser*), the profile

(*TTMSFNCCloudSlackProfile*), the stream of the picture (*TMemoryStream*) and the possible *ErrorMessage (String)*.

OnGetUserConversations/OnGetAllUserConversations

Event fired when the/all requested conversations of which the user is a member of are retrieved. This event returns additional to the sender and the request result also the list of conversations (*TTMSFNCCloudSlackConversations*), the user that is a member of the conversations (*TTMSFNCCloudSlackUser*) and the possible *ErrorMessage (String)*.

OnDeleteMessage

Event fired when the message was deleted in a conversation. This event returns additional to the sender and the request result also a Boolean that indicates if the message was deleted or not, the conversation in which the message was deleted (*TTMSFNCCloudSlackConversation*) and the possible *ErrorMessage (String)*.

OnGetMessageReplies/OnGetAllMessageReplies

Event fired when the/all requested replies of the chosen message are retrieved. This event returns additional to the sender and the request result also the chosen message (*TTMSFNCCloudSlackMessage*) and the possible *ErrorMessage (String)*.

OnPostReply

Event fired when the created reply is added to the message thread. This event returns additional to the sender and the request result also the message in which the reply was added (*TTMSFNCCloudSlackMessage*), the reply itself (*TTMSFNCCloudSlackReply*) and the possible *ErrorMessage (String)*.

OnGetUserInformation

Event fired when the requested user is retrieved. This event returns additional to the sender and the request result also the user (*TTMSFNCCloudSlackUser*) and the possible *ErrorMessage (String)*.

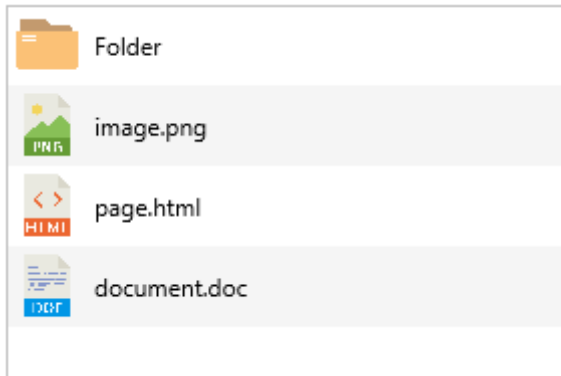
Components

An overview of the helper components included with TMS FNC Cloud Pack:

TMSFNCCloudDemoListBox

Usage

A *ListBox* style component with multiline support that can display a file type bitmap for each item. This component is intended as a helper component for demonstration purposes only.



Organisation

Properties

ItemIndex
The index of the currently selected item

Items
List of items

ItemHeight
Set the height of the displayed items

Methods

AddItem
Add a new item to the list

InsertItem
Inserts a new item in the list

SupportsFileType
Returns if a bitmap is available for the specified file type

Clear
Remove all items from the list

TMSFNCCloudImage

Usage

Image component that can automatically retrieve an image from an URL and display it on the form. It supports BMP, PNG, GIF, ICO, JPEG formats.

Organisation

Properties

LoadFromURL

Sets the URL where to retrieve the image from