FNC

TMS FNC Cloud Pack Services

DEVELOPERS GUIDE

March 2022

Web: https://www.tmssoftware.com
Email: info@tmssoftware.com

Index

Availability 4
Online references 4
Terms of use 4
Limited warranty 5
Description 5
Registering your application 5
Getting Started 5
Demo 6
Services 8
TTMSFNCCloudCloudKit 8
TTMSFNCCloudStorageServices 9
TTMSFNCCloudBox 10
TTMSFNCCloudDropBox 12
TTMSFNCCloudGoogleCalendar 13
TTMSFNCCloudGoogleDrive 14
TTMSFNCCloudGoogleFirebaseObjectDatabase 15
TTMSFNCCloudGoogleAnalytics 17
TTMSFNCCloudSumUp 18
TTMSFNCCloudCloudinary 19
TTMSFNCCloudTranslation 20
TTMSFNCCloudGoogleGMail 21
TTMSFNCCloudGoogleLookup 22
TTMSFNCCloudGoogleLocationLookup 22
TTMSFNCCloudGooglePeople 22
TTMSFNCCloudGooglePhotos 24
TTMSFNCCloudGoogleTasks 24
TTMSFNCCloudGoogleYouTube 26
TTMSFNCCloudHubic 27
TTMSFNCCloudMicrosoftOneDrive 28
TTMSFNCCloudMicrosoftOutlookCalendar 29
TTMSFNCCloudMicrosoftOutlookContacts 30
TTMSFNCCloudMicrosoftOutlookMail 31
TTMSFNCCloudMyCloudData 31
TTMSFNCCloudPayPal 43
TTMSFNCCloudSlack 44
TTMSFNCCloudExceptionless 48
TTMSFNCCloudSentry 49
TTMSFNCCloudRollbar 51
TTMSFNCCloudLogger 53
TTMSFNCCloudIPLocalization 55
Persisting Tokens 57

Components 57
TMSFNCCloudDemoListBox.. 57
TMSFNCCloudImage 58

Availability

Supported frameworks and platforms
- VCL Win32/Win64
- FMX Win32/Win64, macOS, iOS, Android, Linux
- LCL Win32/Win64, macOS, iOS, Android, numerous Linux variants including Raspbian
- WEB: Chrome, Edge, Firefox, …

Supported IDE's
- Delphi XE7 and C++ Builder XE7 or newer releases
- Lazarus 1.4.4 with FPC 2.6.4 or newer official releases
- TMS WEB Core for Visual Studio Code 1.3 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

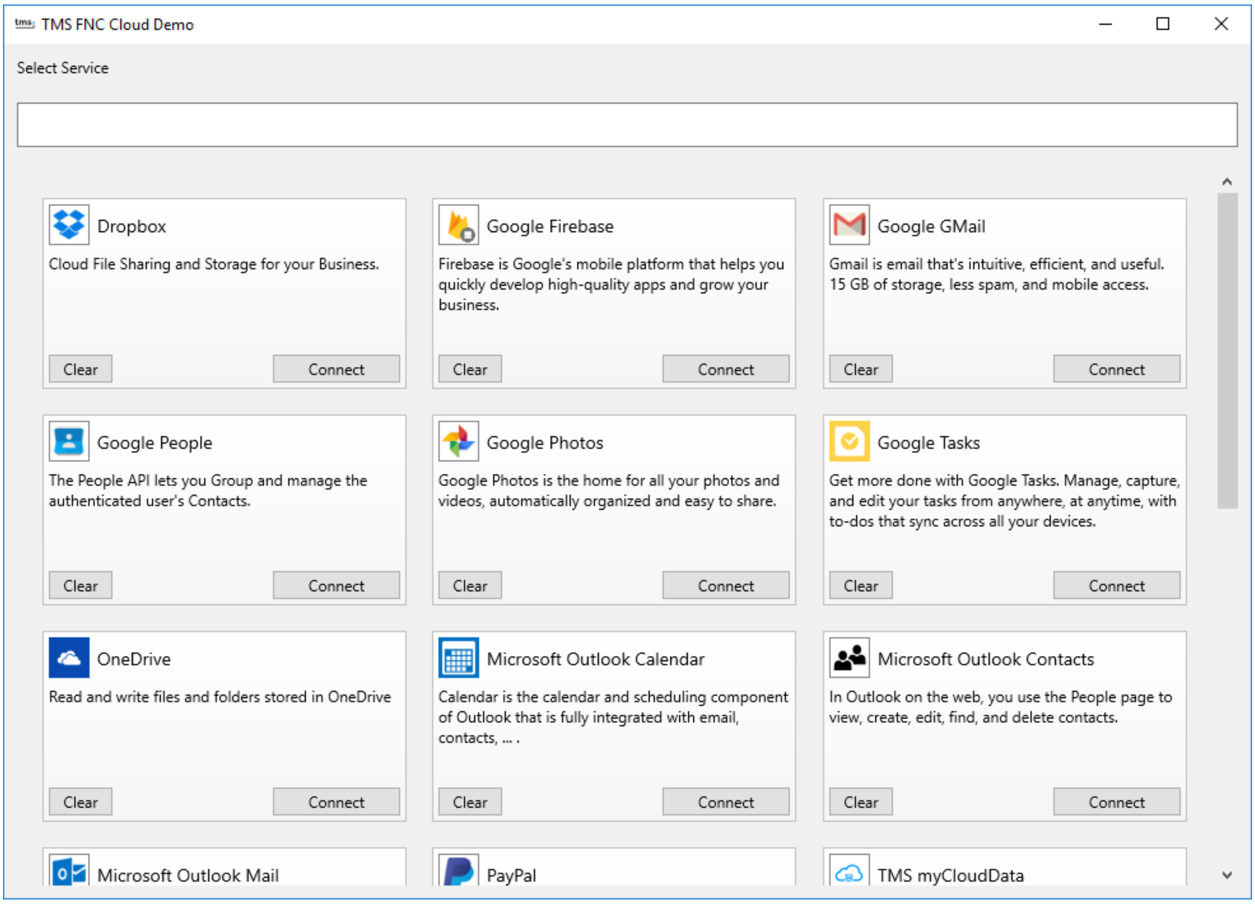
5) After successful authentication, the OnConnected event is triggered and from that moment the component has access the cloud service API

Demo

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

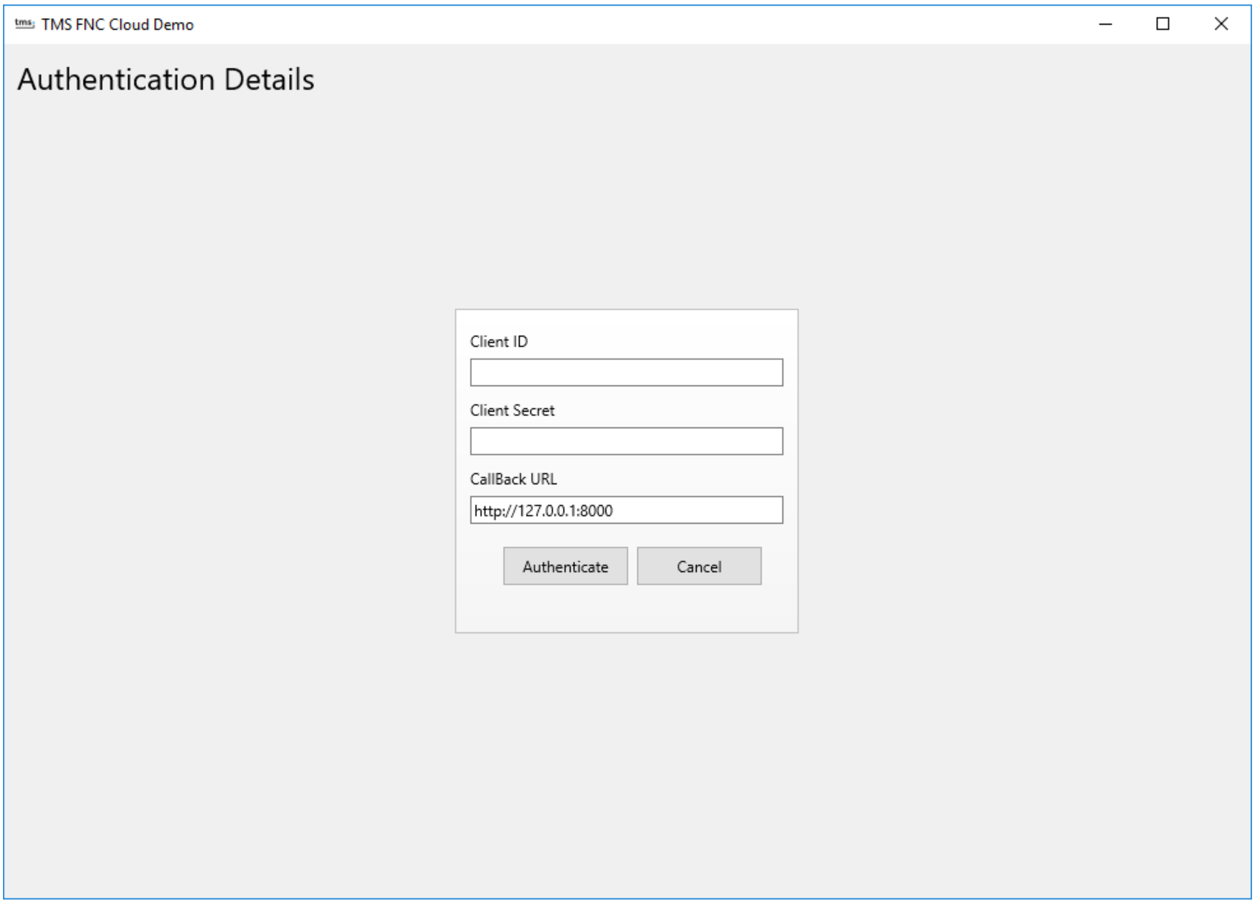
Usage

- Select the one of the services from the tiles menu


TMS FNC Cloud Demo
Select Service
Dropbox
Cloud File Sharing and Storage for your Business.
Clear
Connect
Google Firebase
Firebase is Google's mobile platform that helps you quickly develop high-quality apps and grow your business.
Clear
Connect
Google GMail
Gmail is email that's intuitive, efficient, and useful. 15 GB of storage, less spam, and mobile access.
Clear
Connect
Google People
The People API lets you Group and manage the authenticated user's Contacts.
Clear
Connect
Google Photos
Google Photos is the home for all your photos and videos, automatically organized and easy to share.
Clear
Connect
Google Tasks
Get more done with Google Tasks. Manage, capture, and edit your tasks from anywhere, at anytime, with to-dos that sync across all your devices.
Clear
Connect
OneDrive
Read and write files and folders stored in OneDrive
Clear
Connect
Microsoft Outlook Calendar
Calendar is the calendar and scheduling component of Outlook that is fully integrated with email, contacts, ...
Clear
Connect
Microsoft Outlook Contacts
In Outlook on the web, you use the People page to view, create, edit, find, and delete contacts.
Clear
Connect
Microsoft Outlook Mail
PayPal
TMS myCloudData

- 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.



Authentication Details

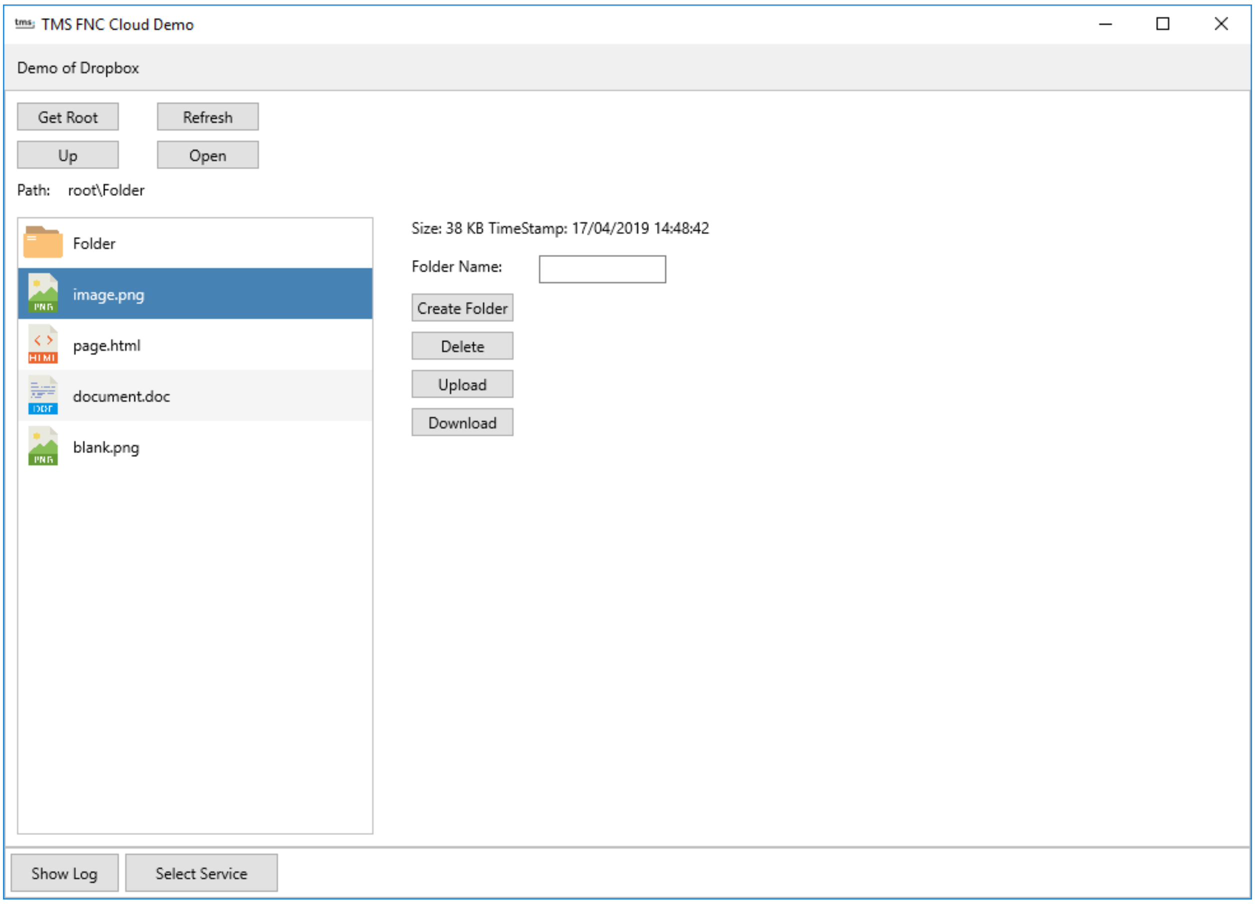
Client ID

Client Secret

CallBack URL

http://127.0.0.1:8000

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



Path: root\Folder

Folder
image.png
page.html
document.doc
blank.png

Size: 38 KB TimeStamp: 17/04/2019 14:48:42

Folder Name:

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

TTMSFNCCloudStorageServices

TTMSFNCCloudStorageServices is an abstract layer on top of the cloud file storage services described elsewhere in this document. It serves as a way to use the selected service in an abstract way, which means that whenever you are switching to another service, the code that was written will be 100% compatible and will behave exactly the same as the service you were originally using, ofcourse under the disclaimer that the service does not change the underlying APIs.

Usage

- Set the Service property to the required cloud file storage service.
- Set the Authentication.ClientID, Authentication.Secret and Authentication.CallBackURL values for the selected service.
- To change to a difference service, simply change the Service and Authentication values.

Supported services

- Box
- DropBox
- Google Drive
- OneDrive
- Hubic *

*Hubic does not support Search or Move. This is a limitation of the Hubic API service.

Organisation

Properties

Drive
A list of files and folders

Service
Set the cloud file storage service to use

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 specific folder and add them to the Drive list

Search
Search for files and folders based on a query

Upload
Upload a file to the drive

MoveFile
Move a file to a different folder

MoveFileToRoot
Move a file to the root folder

TTMSFNCCloudBox

Usage

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

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

Drive
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 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.
Options:

- ExactMatch: indicate if the query should match the filename exactly
- FileNameOnly: indicate if the search applies to the filename only or also to the contents of the file
- FileExtensions: only return files that have one of the given extensions
- Folder: only return results for files in this folder

Upload
Upload a file to the drive

MoveFile
Move a file to a different folder

MoveFileToRoot
Move a file to the root folder

TTMSFNCCloudDropBox

Usage

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

Authorization information

Client ID, Client Secret, CallBack URL

Organisation

Properties

Drive
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

DownloadFolderAsZip
Download all files in a given folder as a single zip file (some limitations to the number of files and total file size apply)

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.
Options:

- ExactMatch: indicate if the query should match the filename exactly
- FileNameOnly: indicate if the search applies to the filename only or also to the contents of the file
- FileExtensions: only return files that have one of the given extensions
- Folder: only return results for files in this folder

Upload
Upload a file to the drive

MoveFile
Move a file to a different folder

MoveFileToRoot
Move a file to the root folder

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, move, 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

Search
Search for files and folders based on a query.
Options:

- ExactMatch: indicate if the query should match the filename exactly
- FileNameOnly: indicate if the search applies to the filename only or also to the contents of the file
- FileExtension: only return files that have the given extension
- Folder: only return results for files in this folder

SearchFile
Search for files based on a query

SearchFolder
Search for folders based on a query

Upload
Upload a file to the drive

MoveFile
Move a file to a different folder

MoveFileToRoot
Move a file to the root folder

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.

TTMSFNCCloudGoogleAnalytics

Usage

TTMSFNCCloudGoogleAnalytics is a component that provides access to the Google Analytics service. It enables to get insights into how visitors find and use your site, and how to keep them coming back.

TTMSFNCCloudCustomGoogleAnalytics

Methods

GetData(AStartDate: string = 'today'; AEndDate: string = 'today'; AMaxResults: Integer = 1000): string;
Gets all the data that was requested by adding dimensions & metrics to the *RequestData* property.

Properties

Data: TTMSFNCCloudGoogleAnalyticsData
The returned data.

RequestData: TTMSFNCCloudGoogleAnalyticsRequestData
The place to distinguish the needed dimensions and metrics.

ErrorMessages: TTMSFNCCloudGoogleAnalyticsErrorMessages
Can set the error message for metrics and dimensions, when that error occurs.

Events

OnRetrieveData
Event called when successfully retrieving data from the service.

OnRetrieveDataError
Event called when an error occurs during data retrieval.

For more information about metrics and dimensions, please see the analytics reporting api and analytics realtime reporting api

TTMSFNCCloudGoogleAnalyticsData

Property

Data: TTMSFNCCloudGoogleAnalyticsStringArrayList
A list of all requested data (in the same order as requested)

TTMSFNCCloudSumUp

Usage

A component that provides access to the SumUp payment service. It allows users to make payments using a credit card.

Properties

MerchantID: string;
The ID tied to your merchant account necessary to initiate payments.

Checkout: TTMSFNCCloudSumUpCheckout;
Checkout information such as the amount, and description about the product/service you are selling.

Card: TTMSFNCCloudSumUpCard;
The user credit card information to authorize the payment

CheckoutError: string;

Error description when a payment failed.

CheckoutID: string;
ID of the checkout after making a payment that is succesfull or still pending

TransactionID: string;
ID of the transaction after making a succesfull payment

Methods

procedure DoCheckout;
Method to start a payment checkout after filling in the MerchantID, Checkout and Card information.
Events

OnCheckoutFailed
Event called when a checkout failed.

OnCheckoutAccepted
Event called when a checkout is accepted.

OnCheckoutPending
Event called when a checkout is pending.

OnCreateCheckout
Event called when a checkout is created.

OnProcessCheckout
Event called when a checkout is processed.

OnGetCheckoutAccess
Event called when the component is request access to being creating the checkout.

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

TTMSFNCCloudCloudinary

Usage

TTMSFNCCloudCloudinary is a component that provides access to the Cloudinary API. It provides support to upload an image file to a publically accessible URL.

Organisation

Properties

CloudName: string;

Extra property for authentication. Required to be assigned with the CloudName value of the Cloudinary app.

UploadPreset: string;

Extra property for authentication. Required to be assigned with the preset value configured in the

Cloudinary app.

Detailed information for both properties is available at:
http://www.tmssoftware.com/site/cloudkey.asp

Methods

UploadFile, UploadFileFromResource, UploadFileFromURL

Uploads a file. Returns the URL for the uploaded file in the OnUploadFile event

Events

OnUploadFile
Event called when a file is successfully uploaded.

TTMSFNCCloudTranslation

Usage

The TTMSFNCCloudTranslation component is a wrapper around various translation services:

- Microsoft Translator
- IBM Watson Translate
- Google Translate

Properties

SupportedLanguages
List of name / value paired language and language short code supported by the service to translate and / or detect text. For example: English-en

Translations
List of translations based on the original text and the translation language after calling the Translate method.

Detections
List of language detections based on the original text and detection language after calling the Detect method.

Service
Property to switch between the supported translation services.

Methods

GetSupportedLanguages
Method to get the supported languages used by the service

Translate
Method to translate single or multiple lines of text.

Detect
Method to detect the language of a single line of text.

Events

OnGetSupportedLanguages
Event called when GetSupportedLanguages method is called and the list of supported languages is successfully retrieved from the service.

OnTranslate
Event called when the Translate method is called and the translated text is successfully retrieved from the service.

OnDetect
Event called when the Detect method is called and the detected language is successfully retrieved from the service.

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 TasksLists 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 rateing for the specified Video

GetAllVideos
Fill the list of Videos

GetFirstVideos
Fill the list of Videos. If more Videos are available then 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, move, 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. Optionally indicate if the Share is ReadOnly or not

Search
Search for files based on a query

Upload
Upload a file to the drive

MoveFile
Move a file to a different folder

MoveFileToRoot
Move a file to the root folder

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: TTMSFNCCloudMyCloudDataMetaDataItem)
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: TTMSFNCCloudMyCloudDataMetaDataitem)
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 limitied 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

<u>TTMSFNCCloudSlack</u>

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’ 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 *(TTMSFNCUtilsFile)*.

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 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 messega. 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)*.

OnGetUserConverasations/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)*.

TTMSFNCCloudExceptionless

Usage

TTMSFNCCloudExceptionless is a component that provides access to the Exceptionless API which is capable of managing exceptions and log messages.

Authorization information

Email address and password

Organisation

TTMSFNCCloudExceptionless

Properties

EmailAddress: *String*
The email address of the user to authenticate.

Password: *String*
The password of the user to authenticate.

Projects: *TTMSFNCCloudExceptionlessProjects [TTMSFNCCloudExceptionlessProject]*
Contains the different projects that you have set up in Exceptionless.
These projects each have different stacks (*TTMSFNCCloudExceptionlessStacks)* in the property Stacks. This is a group of exceptions and messages with the same parameters and can be requested in the same way as the Projects on *TTMSFNCCloudExceptionlessProject* level. These stacks in turn have the property Events (*TTMSFNCCloudExceptionlessEvents)*, which is a collection of the exceptions and messages and can be retrieved in the same way on *TTMSFNCCloudExceptionlessStack* level.

ProjectPage: *Integer*
The page number to add to the GetProjects method to retrieve the following projects.

Methods

Connect
Retrieve the necessary authentication tokens to use the API. (Has the possibility to add email address and password as parameters in case they are not set via the properties.)

GetProjects/GetAllProjects
Retrieve (all) the projects for the account. The procedure fills the Projects *(TTMSFNCCloudExceptionlessProjects)* property.

GetProjectByID
Retrieve the *TTMSFNCCloudExceptionlessProject* object that has the requested ID.

GetProjectByName
Retrieve the *TTMSFNCCloudExceptionlessProject* object that has the requested name.

GetStackByID
Retrieve the *TTMSFNCCloudExceptionlessStack* object that has the requested ID.

LogException
Sends an exception to the project with a message and an error type. Can be extended with an optional source, stacktrace, level and timestamp. *(It is not necessary to retrieve the requested project first, if you are connected and you have the ID of the Project, you can immediately log an exception.)*

LogMessage
Sends a message to the project. Can be extended with an optional source, timestamp and level. *(It is not necessary to retrieve the requested project first, if you are connected and you have the ID of the Project, you can immediately log a message.)*

Events

OnConnected
Event fired when the authorization token was retrieved.

OnGetProjects/OnGetAllProjects
Event fired when the/all requested projects for the account are retrieved. This event returns additional to the sender and the request result also the Projects *(TTMSFNCCloudExceptionlessProjects)*.

OnGetStacks/OnGetAllStacks
Event fired when the/all requested stacks from a project are retrieved. This event returns additional to the sender and the request result also the project for which it was returned and the list of stacks (*TTMSFNCCloudExceptionlessStacks)*.

OnGetEvents/OnGetAllEvents
Event fired when the requested events for a stack are retrieved. This event returns additional to the sender and the request result also the stack of the project *(TTMSFNCCloudExceptionlessStack)* and the requested events *(TTMSFNCCloudExceptionlessEvents)*.

OnLogException
Event fired when the created exception is sent to the project. This event returns the sender and the request result.

OnLogMessage
Event fired when the created exception is sent to the project. This event returns the sender and the request result.

TTMSFNCCloudSentry

Usage

TTMSFNCCloudSentry is a component that provides access to the Sentry API which is capable of managing exceptions and log messages.

Authorization information

Authentication token *(via 'Account' > 'API').*

Organisation

Remark

When used on TMS WEB Core, CORS issues may arise with the message 'Origin is invalid'. This can be fixed by createing an integration in the 'Settings' of the project in 'Developer Settings'. Use the token on that page.

TTMSFNCCloudSentry

Properties

AuthenticationToken: *String*
The authentication token of the user with the necessary scopes.

Projects: *TTMSFNCCloudSentryProjects [TTMSFNCCloudSentryProject]*
Contains the different projects that you have set up in Sentry.
These projects each have different issues (*TTMSFNCCloudSentryIssues)* in the property Issues. This is a group of exceptions and messages with the same parameters and can be requested in the same way as the Projects on *TTMSFNCCloudSentryProject* level. These issues in turn have the property Events (*TTMSFNCCloudSentryEvents)*, which is a collection of the exceptions and messages and can be retrieved in the same way on *TTMSFNCCloudSentryIssue* level.

ProjectCursor: *String*
The cursor to add to the GetProjects method to retrieve the following projects.

Methods

Connect
Checks if the authentication token is valid. (Has the possibility to add an authentication token as a parameter in case this was not set via the property.)

GetProjects/GetAllProjects
Retrieve (all) the projects for the account. The procedure fills the Projects *(TTMSFNCCloudSentryProjects)* property.

GetProjectByID
Retrieve the *TTMSFNCCloudSentryProject* object that has the requested ID.

GetProjectByName
Retrieve the *TTMSFNCCloudSentryProject* object that has the requested name.

GetIssueByID
Retrieve the *TTMSFNCCloudSentryIssue* object that has the requested ID.

LogException
Sends an exception to the project with a message, an error type and source. Can be extended with an optional stacktrace, level, timestamp and value. *(It is necessary to retrieve the requested project first via GetProjects/GetAllProjects.)*

LogMessage
Sends a message to the project. Can be extended with an optional level, timestamp, source and stacktrace. *(It is necessary to retrieve the requested project first via GetProjects/GetAllProjects.)*

Events

OnConnected
Event fired when the authorization token is valid.

OnGetProjects/OnGetAllProjects
Event fired when the/all requested projects for the account are retrieved. This event returns additional to the sender and the request result also the Projects *(TTMSFNCCloudSentryProjects).*

OnGetIssues/OnGetAllIssues
Event fired when the/all requested issues from a project are retrieved. This event returns additional to the sender and the request result also the project *(TTMSFNCCloudSentryProject*) for which it was returned and the list of issues (*TTMSFNCCloudSentryIssues).*

OnGetEvents/OnGetAllEvents
Event fired when the requested events for an issue are retrieved. This event returns additional to the sender and the request result also the issue of the project *(TTMSFNCCloudSentryIssue)* and the requested events *(TTMSFNCCLoudSentryEvents).*

OnLogException
Event fired when the created exception is sent to the project. This event returns the sender, the event ID and the request result.

OnLogMessage
Event fired when the created exception is sent to the project. This event returns the sender, the event ID and the request result.

TTMSFNCCloudRollbar

Usage

TTMSFNCCloudRollbar is a component that provides access to the Rollbar API which is capable of managing exceptions and log messages.

Authorization information

Account authentication token with read scopes *(via 'Settings' > 'Account Access Tokens').*
It is also necessary to make a project access token available with the post_client_item scope if you want to log messages and exceptions *(via 'Settings' > 'Project Access Tokens').*

Organisation

TTMSFNCCloudRollbar

Properties

AccountTokenRead: *String*
The account authentication token of the user with the necessary scopes.

Projects: *TTMSFNCCloudRollbarProjects [TTMSFNCCloudRollbarProject]*
Contains the different projects that you have set up in Rollbar.
These projects each have different items (*TTMSFNCCloudRollbarItem)* in the collection Items. This is a group of exceptions and messages with the same parameters and can be requested in the same

way as the Projects on *TTMSFNCCloudRollbarProject* level. These issues in turn have the property Occurences (*TTMSFNCCloudRollbarOccurences)*, which is a collection of the exceptions and messages and can be retrieved in the same way on *TTMSFNCCloudRollbarItem* level.

ProjectCursor: *String*
The cursor to add to the GetProjects method to retrieve the following projects.

Methods

Connect
Checks if the authentication token is valid. (Has the possibility to add an authentication token as a parameter in case this was not set via the property.)

GetProjects/GetAllProjects
Retrieve (all) the projects for the account. The procedure fills the Projects *(TTMSFNCCloudRollbarProjects)* property.

GetProjectByID
Retrieve the *TTMSFNCCloudRollbarProject* object that has the requested ID.

GetProjectByName
Retrieve the *TTMSFNCCloudRollbarProject* object that has the requested name.

GetItemByID
Retrieve the *TTMSFNCCloudRollbarItem* object that has the requested ID.

LogException
Sends an exception to the project with a message, an error type and source. Can be extended with an optional stacktrace, level, timestamp, description and environment which is by default set to ‘production’. *(It is necessary to retrieve the requested project first via GetProjects/GetAllProjects.)*

LogMessage
Sends a message to the project. Can be extended with an optional level, timestamp, source and environment which is by default set to ‘production’. *(It is necessary to retrieve the requested project first via GetProjects/GetAllProjects.)*

Events

OnConnected
Event fired when the authorization token is valid.

OnGetProjects/OnGetAllProjects
Event fired when the/all requested projects for the account are retrieved. This event returns additional to the sender and the request result also the Projects *(TTMSFNCCloudRollbarProjects)* and a possible error message.

OnGetItems/OnGetAllItems
Event fired when the/all requested items from a project are retrieved. This event returns additional to the sender and the request result also the project *(TTMSFNCCloudRollbarProject*) for which it was returned, the list of items (*TTMSFNCCloudRollbarItems)* and a possible error message.

OnGetOccurences/OnGetAllOccurences
Event fired when the requested occurences for an item are retrieved. This event returns additional to the sender and the request result also the item of the project *(TTMSFNCCloudRollbarItem)*, the requested occurences *(TTMSFNCCloudRollbarOccurences)* and a possible error message.

OnLogException
Event fired when the created exception is sent to the project. This event returns the sender, the project *(TTMSFNCCloudRollbarProject*), a possible error message and the request result.

OnLogMessage
Event fired when the created message is sent to the project. This event returns the sender, the project *(TTMSFNCCloudRollbarProject*), a possible error message and the request result.

TTMSFNCCloudLogger

Usage

TTMSFNCCloudLogger is a wrapper component that provides access to the Exceptionless, Sentry and Rollbar API's which are capable of managing exceptions and log messages.

Authorization information

See the documentation of the separate services for more detailed information.
Email address and password are used for Exceptionless and a token is necessary for Sentry and Rollbar.

Organisation

TTMSFNCCloudLogger

Properties

LogService: TTMSFNCCloudLoggerService
The service that is being used. *(lsExceptionless, lsRollbar or lsSentry.)*
When switching between services, all of the stored projects are cleared.

EmailAddress: *String*
The email address of the user to authenticate. *(Used for Exceptionless.)*

Password: *String*
The password of the user to authenticate. *(Used for Exceptionless.)*

Token: *String*
The authentication token of the user with the necessary scopes. *(Used for Sentry and Rollbar.)*

Projects: *TTMSFNCCloudLoggerProjects [TTMSFNCCloudLoggerProject]*
Contains the different projects that you have set up in your service.
These projects each have different stacks/issues/items (*TTMSFNCCloudLoggerStack)* in the collection Stacks. This is a group of exceptions and messages with the same parameters and can be requested in the same way as the Projects on *TTMSFNCCloudLoggerProject* level. These stacks in turn have the property Events (*TTMSFNCCloudLoggerEvents)*, which is a collection of the exceptions and messages and can be retrieved in the same way on *TTMSFNCCloudLoggerStack* level.

ProjectCursor: *String*
The cursor to add to the GetProjects method to retrieve the following projects.

Methods

Connect
Checks if the authentication token is valid or retrieves an authentication token. (Has the possibility to add an email address and password or an authentication token as a parameter in case this was not set via the properties.)

GetProjects/GetAllProjects
Retrieve (all) the projects for the account. The procedure fills the Projects *(TTMSFNCCloudLoggerProjects)* property.

GetProjectByID
Retrieve the *TTMSFNCCloudLoggerProject* object that has the requested ID.

GetProjectByName
Retrieve the *TTMSFNCCloudLoggerProject* object that has the requested name.

GetItemByID
Retrieve the *TTMSFNCCloudLoggerStack* object that has the requested ID.

LogException
Sends an exception to the project with a message, an error type and source. Can be extended with an optional stacktrace, level and timestamp. *(In the Rollbar service the environment is fixed to the default value 'production'.)*

LogMessage
Sends a message to the project. Can be extended with an optional level, timestamp and source. *(In the Rollbar service the environment is fixed to the default value 'production'.)*

Events

OnConnected
Event fired when the authorization token is retrieved or valid.

OnGetProjects/OnGetAllProjects
Event fired when the/all requested projects for the account are retrieved. This event returns additional to the sender and the request result also the Projects *(TTMSFNCCloudLoggerProjects)*.

OnGetStacks/OnGetAllStacks
Event fired when the/all requested stacks from a project are retrieved. This event returns additional to the sender and the request result also the project (*TTMSFNCCloudLoggerProject*) for which it was returned and the list of stacks (*TTMSFNCCloudLoggerStacks)*.

OnGetEvents/OnGetAllEvents
Event fired when the requested events for a stack are retrieved. This event returns additional to the sender and the request result also the stack of the project *(TTMSFNCCloudLoggerStacks)* and the requested events *(TTMSFNCCloudLoggerEvents)*.

OnLogException
Event fired when the created exception is sent to the project. This event returns the sender and the request result.

OnLogMessage
Event fired when the created exception is sent to the project. This event returns the sender and the request result.

TTMSFNCCloudIPLocalization

Usage

TTMSFNCCloudIPLocalization is a wrapper component that gives you the ability to retrieve information based on a given IP address. In this wrapper we have combined the services: Abstract API, ip data and ipstack

Authorization information

An API key is needed for all the services.

Organisation

TTMSFNCCloudIPLocalization

Properties

APIKey: *String*
The API key for the specific service.

Service: TTMSFNCCloudIPLocalizationService
The service that is being used. *(ipsAbstractAPI, ipsIPData, ipsIPStack)*

IPs: *TTMSFNCCloudIPLocalizationIPs [TTMSFNCCloudIPLocalizationIP]*
Contains the data of the different ip's that are requested.

Methods

GetIP
Retrieves the requested ip addres, if this is empty, the local ip data will be retrieved.

GetLocalizedIPByIPAddress
Retrieve the *TTMSFNCCloudLoggerProject* object that has the requested ID.

Events

OnGetIP
Event fired when the IP data is retrieved.

OnGetFlagImage *(not in TMS WEB Core)*
Event fired when the country image of the IP was downloaded.
(This is not necessary on TMS WEB Core as an url to the image is available for each IP via the property Flag.ImageUrl.)

TTMSFNCCloudIPLocalizationIP

Properties

IPAddress: *String*
City: *String*
Region: *String*
RegionCode: *String*
PostalCode: *String*

Country: *String*
CountryCode: *String*
ContinentName: *String*
ContinentCode: *String*

Continent: *TTMSFNCCloudIPLocalizationContinent*
Enumerator of the continent. *(ipcUnknown, ipcAfrica, ipcAsia, ipcEurope, ipcNorthAmerica, ipcOceania, ipcSouthAmerica, ipcAntartica)*

Longitude: *Double*
Latitude: *Double*

IsVPN: *Boolean*
Is a VPN or proxy used by the IP address.

Timezone: TTMSFNCCloudIPLocalizationIPTimeZone
Record that holds the name, abbreviation, GMT offset, current time and if it is daylight savings time.

Flag: *TTMSFNCCloudIPLocalizationIPFlag*
Record that holds the name, imageURl, UniCode and the image stream.

AutonomousSystem: *TTMSFNCCloudIPLocalizationAutonomousSystem*
Record that holds the name, isp, number and connection type.

Methods

GetFlagImage *(not in TMS WEB Core)*
Automatically called when retrieving the IP information.

Persisting Tokens

INI file

Use PersistTokens.Key, this will be the filename of the ini file.

DataBase

Use the PersistTokens.DataSource and the PersistTokens.Location := plDataBase, to connect to a DataSource.

Required fields:

ACCESSTOKEN
AUTHENTICATIONTOKEN
ACCESSTOKENSECRET
AUTHENTICATIONTOKENSECRET
ACCESSTOKENREFRESH

Optional fields:

CLIENTID
KEY
SECRET
CALLBACKURL
EXTRA

Tokens can be saved by using PersistTokens.SaveClientID, SaveKey, SaveSecret, SaveCallBack and SaveExtra.

When calling SaveTokens, the tokens will be saved to the database. This all happens automatically once you properly configure the database, credentials and when you succesfully connect to the service.

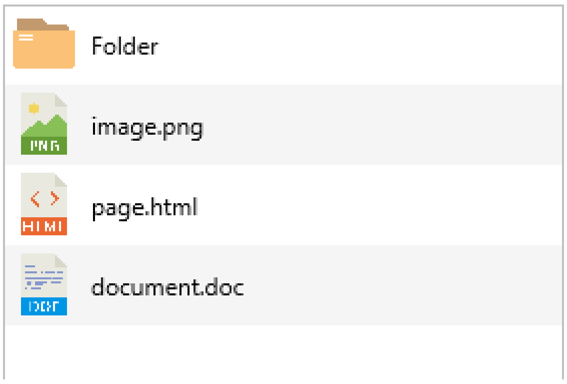
Components

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.



Folder

image.png

page.html

document.doc

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