



Reference Guide

RG 005-0-EN / Rev. 1.3

Data Mailbox

Talk2M SDK - 1.3.0.26452

The Data Mailbox is a Talk2M service that gathers eWON historical data and makes it available for third party applications in an easy way.



Table of Contents

- 1. Introduction 4**
 - 1.1. Prerequisites 4
 - 1.2. Talk2M SDK 5
 - 1.2.1. M2Web API 5
 - 1.2.2. DataMailbox 5
- 2. eWON Configuration 7**
 - 2.1. Data synchronizing parameters 7
 - 2.2. Synchronization using script 8
 - 2.3. Synchronization status 9
- 3. DMWeb API 10**
 - 3.1. API Request Structure 10
 - 3.2. Response Structure 10
 - 3.3. Credentials 10
 - 3.3.1. GET or POST 11
 - 3.4. Dates format 11
 - 3.5. Read Data Services 12
 - 3.5.1. getstatus 12
 - 3.5.1.1. Request 12
 - 3.5.1.2. Response 12
 - 3.5.1.3. Example 12
 - 3.5.2. getewons 13
 - 3.5.2.1. Request 13
 - 3.5.2.2. Response 13
 - 3.5.2.3. Example 13
 - 3.5.3. getewon 14
 - 3.5.3.1. Request 14
 - 3.5.3.2. Response 14
 - 3.5.3.3. Example 14
 - 3.5.4. getdata 14
 - 3.5.4.1. Request 14
 - 3.5.4.2. Response 15
 - 3.5.4.3. Examples 16
 - 3.5.5. syncdata 16
 - 3.5.5.1. Request 16
 - 3.5.5.2. Response 16
 - 3.5.5.3. Examples 17
 - 3.6. Delete the Data 17
 - 3.6.1. delete 17
 - 3.6.1.1. Request 17
 - 3.6.1.2. Response 18
 - 3.6.1.3. Examples 18
 - 3.6.2. clean 18
 - 3.6.2.1. Request 18
 - 3.6.2.2. Response 18
 - 3.6.2.3. Examples 19



Chapter 1
Endnotes

Revision **20**
Revision History 20

1. Introduction

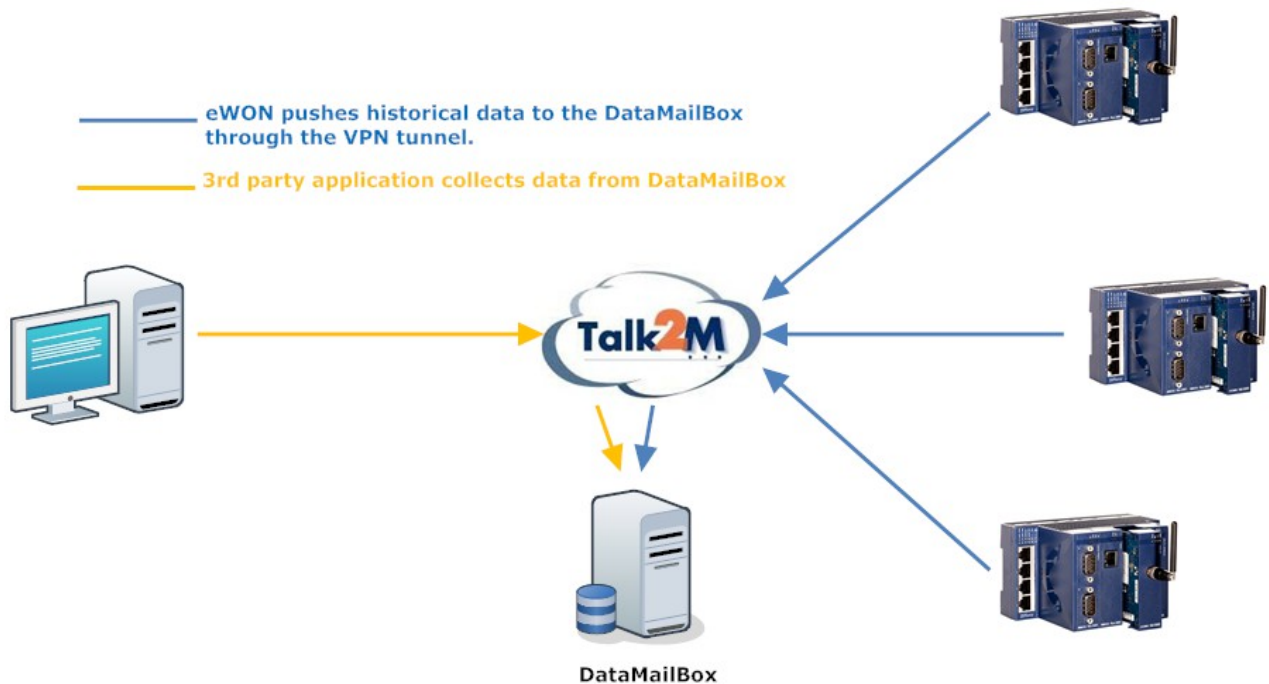
The Talk2M Service "Data Mailbox" allows easy retrieval of eWON historical data. Application developers can easily write code to retrieve historical data of multiple eWONs using the Data Mailbox without the need of learning a whole new environment.

The eWON pushes its historical data to the Data Mailbox running on Talk2M servers. This historical data is temporarily stored and is then available using the DMWeb protocol.

If the Data Mailbox act as a temporary storage, there is a limitation in the capacity that can be stored:

- Maximum 500.000 NEW points per month per DataMailbox on FREE+ account
- 3.000.000 NEW points per month per Datamailbox included on a Talk2M PRO account. Additional points can be bought .
- A point can be stored for 10 days maximum, beginning when it was sent to the Data Mailbox.

This HTTP based protocol allows third party applications to retrieve data from the Data Mailbox in an easy way.



1.1. Prerequisites

To use correctly the DMWeb Protocol, you need some prerequisites such as:

- **eWON(s)**
The eWONs need to have the ability to perform the historical logging. This means you either have an eWON that belongs to the CD 400# series or the Flexy family.
You then have to make sure the tags are correctly configured and have enabled this historical logging option.
- **Talk2M Account**
The eWONs which you want to retrieve some tags values from have to be associated with a Talk2M account.

- **Talk2M Developer ID**

You also need a Talk2M Developer ID in order to send requests and receive responses to/from the Talk2M servers. This Talk2M Developer ID can be requested by filling in a web form on [eWON Developers](#) website.

If one of these prerequisites are not fulfilled, we highly suggest to visit [eWON Support](#) to fix the missing parts before going any further.

1.2. Talk2M SDK

In the Talk2M SDK you downloaded, two different folders can be found: DataMailbox and M2Web API.

1.2.1. M2Web API

The M2Web API exposes a set of HTTPS web services based on the Talk2M M2Web HTTPS service.

The API exposes web services aimed at querying M2Web information and at accessing to the eWON web server. More information, please refer to [eWON Developers](#) website or to the Reference Guide located in that folder.

1.2.2. DataMailbox

The DataMailbox is explained in the present document. The DataMailbox folder from the Talk2M SDK should contain:

- **This Reference Guide**
- **Viewer**

Talk2M DataMailbox Viewer is a software that offers the possibility to generate the needed URL to retrieve the different values of the API requests.

Instead of writing manually each parameter of the URL, this software helps you by letting you select the request you want to send to the DataMailbox and shows its result immediately.

It is useful to check if the request will succeed or not, which information you want to retrieve but most certainly how to write the URL.

To use the Viewer software, simply double-click on the "DMBoxViewer.exe" file.

- **DataMailbox Samples**

This folder contains applications with their source codes. Applications that are:

- My Little Historian

Sample C# console application that retrieves the contents of the DataMailbox using the [DMWeb syncdata mechanism](#), outputs the contents in text files (one sub-directory per eWON, one text file per tag) and deletes the contents of the DataMailbox.

This sample program shows how to download data using compressed HTTPS in C#, turns it into a dynamic .NET object, browses through its contents and uses the DMWeb transaction mechanism.

Requirement: Visual Studio 2010+ / .NET 4.0+ / C#

- DMBoxViewer

The source code of theTalk2M Datamailbox Viewer which is explained here above can be found in this folder.

It demonstrates the creation of a DMWeb URL, including the various options of each API call and the download of the DataMailbox contents.

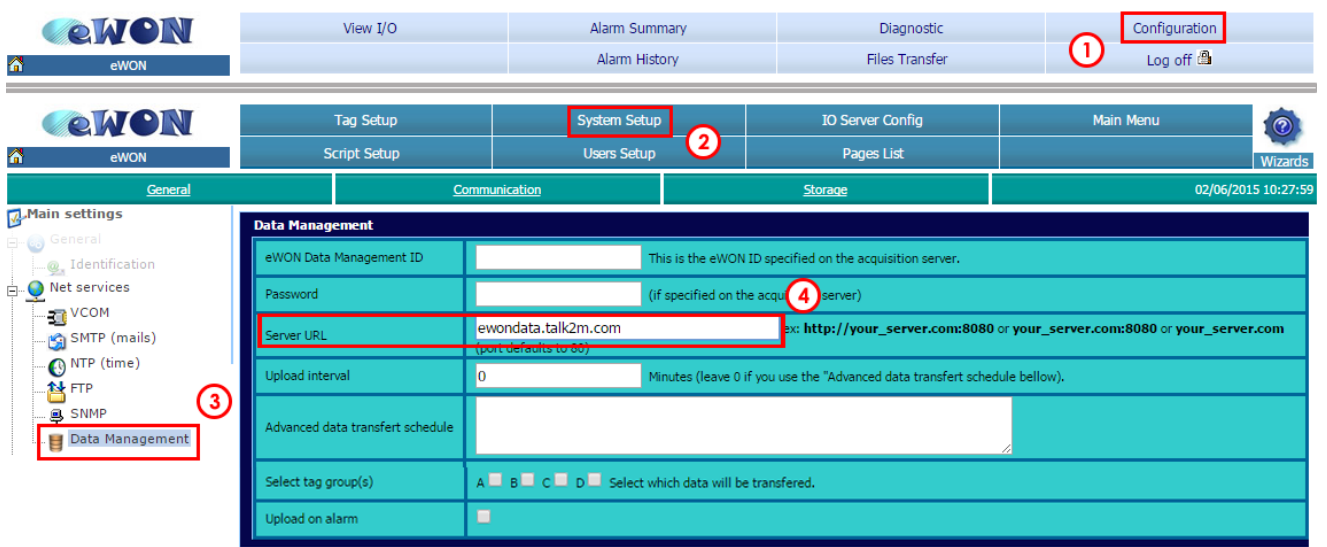
Requirement: Visual Studio 2010+ / .NET 4.0+ / C#

2. eWON Configuration

2.1. Data synchronizing parameters

In order to send its historical data to the Data Mailbox, the Server URL configured in the Data Management page of the eWON must be set to **ewondata.talk2m.com**.

You must also configure an "Upload interval" or an "Advanced data transfer schedule" so the eWON can send periodically its data.



The screenshot shows the eWON web interface. At the top, there is a navigation bar with 'View I/O', 'Alarm Summary', 'Diagnostic', and 'Configuration' (highlighted with a red box and number 1). Below this is another bar with 'Tag Setup', 'System Setup' (highlighted with a red box and number 2), 'IO Server Config', and 'Main Menu'. The main content area is divided into sections: 'General', 'Communication', and 'Storage'. On the left, there is a 'Main settings' sidebar with 'Data Management' highlighted (number 3). The 'Data Management' section is expanded, showing fields for 'eWON Data Management ID', 'Password', 'Server URL' (highlighted with a red box and number 4, containing 'ewondata.talk2m.com'), 'Upload interval' (set to 0), 'Advanced data transfer schedule', 'Select tag group(s)', and 'Upload on alarm'.

- Note -

eWON Data Management ID and Password are not used by the DataMailbox.

Several options that can be modified:

- **Upload interval**
Specifies the synchronization interval. For example "60", to synchronize every 60 minutes.
- **Advanced data transfer schedule**
This field allows to encode a more complex synchronization schedule than when using the Upload interval option.

To use the advanced scheduler set the "Upload interval" field to 0 and use the same syntax as the one used for the eWON Task Planner.

For example: 0 */6 * * * triggers the upload every 6 hours (at 0, 6, 12 and 18 o'clock).

Refer to the eWON [General Reference Guide](#) for detailed syntax information (RG-001-0-EN)

- **Select tag group(s)**
This option allows to synchronize the historical data only for a few tags instead of all eWON tags. When a Tag group is selected, the eWON exports the historical data values only for the tags belonging to the selected group.

- Note -

This filter applies only to the historical stored values of the eWON. The filter is not applied on the realtime and alarm values. So the realtime values and alarm info of all tags will still be synchronized.

- **Upload on alarm**
When selected, the data synchronization will also be triggered when one of the tags of the selected groups rises an alarm.

2.2. Synchronization using script

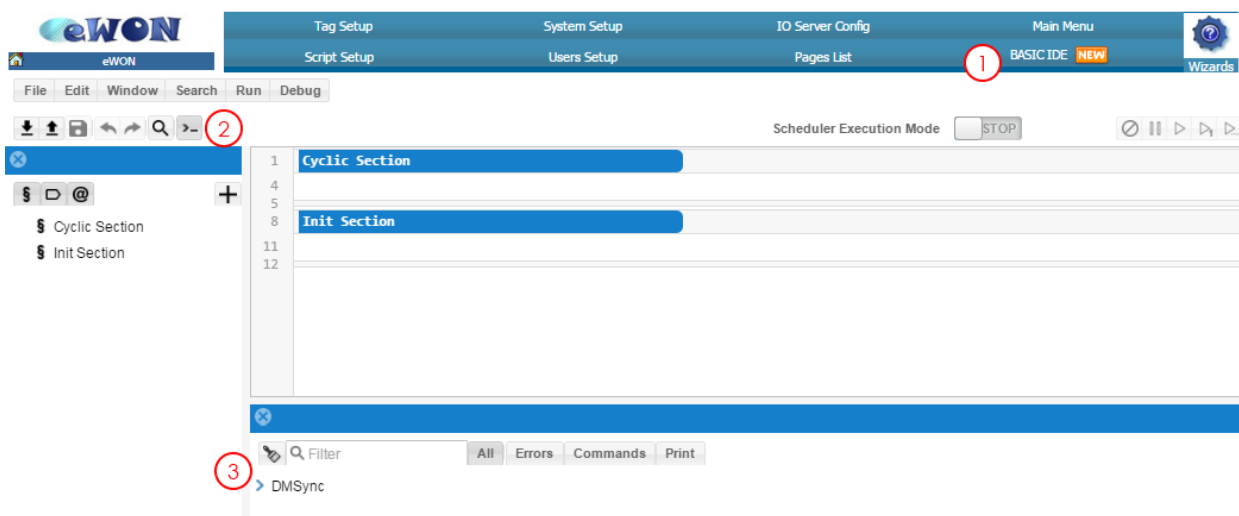
Next to the automatic synchronization described here above, it is also possible to trigger the data synchronization using basic script.

The basic function "DMSync" will trigger the eWON in order to synchronize its data.

For example to trigger a single synchronization, you can proceed as follows:

- Open the eWON Basic IDE section,
- Make sure the console frame is opened,
- Enter "DMSync" into the field on the bottom of window and hit "Enter".

A data synchronization is then immediately started.

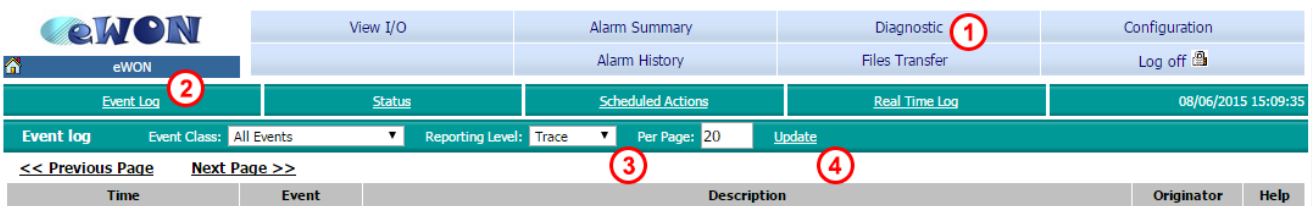


2.3. Synchronization status

To check if the synchronization succeeded, open the eWON diagnostic window using following link:

```
Main Menu → Diagnostic → Event Log
```

In the Green menu select "Trace" from the Reporting Level box and click on the "Update" button.



The event Log should contain 2 messages:

- one for the synchronization start.
- one for the synchronization end.

As this means that the eWON has transferred the data to the DataMailbox, you can now have a look on the Server side to visualize the received data.

3. DMWeb API

The DMWeb API is used to retrieve the historical data of all your eWONs from the DataMailbox. This API allows to :

- read the data stored in the DataMailbox
- delete the data from the DataMailbox.

3.1. API Request Structure

The hostname of the DMWeb API is:

```
https://data.talk2m.com/
```

3.2. Response Structure

The API response is formatted in [JSON](#). It always contains the following value:

- **success**: is a boolean value. It is set to *true* when the API was treated without any issue. And *false* when an error occurred during the treatment of the response.

When *success* is *false*, it also contains the following values:

- **code**: an error code.
- **message**: a description explaining the issue that occurred.

3.3. Credentials

The credentials mechanism is similar to the [M2Web API](#)'s ones.

Credentials have to be defined in the HTTP GET query string or in the content of the HTTP POST query. The following parameters must be provided:

- **t2maccount**
- **t2musername**
- **t2mpassword**
- **t2mdevid**

- Note -

The *t2mdevid* is a Talk2M Developer ID specific to the DataMailbox. To request one, fill in the web form that can be found on [eWON Developers](#) website.

3.3.1. GET or POST

Both GET and POST protocols are secure when using the M2Web API. Even so, we recommend using POST requests since they add another layer of security.

Get request is still a valid choice: GET is being used through SSL connections which assures a secure environment. But alternative methods can be used by hackers to steal credentials such as browser history, server logs... POST requests do resist better to those kinds of attack.

- Note -

In the following examples of this document, we use the GET method to ease the comprehension of the different queries but in practice POST requests should be used, especially when sending credential parameters such as t2mdevid, t2mpassword.

3.4. Dates format

The format of the dates used by the DataMailbox follows the standard [ISO 8601](#).

In JSON, the dates have the following format:

```
2015-02-24T03:03:42Z
```

For the dates used as parameters, they may have the following format :

```
2015-02-24T03:03:42Z
```

```
2015-02-24T03:03:42
```

```
2015-02-24T03:03 (meaning 2015-02-24T03:03:00Z)
```

```
2015-02-24 (meaning 2015-02-24T00:00:00Z)
```

When the dates are in the URL, they must be encoded in a special format. The column (":") character needs to be encoded as "%3A":

```
2015-02-24T03%3A03%3A42Z
```

3.5. Read Data Services

As a reminder to [GET or POST](#) chapter, we recommend using the POST query.

In order to ease the comprehension of the following examples, these ones are explained in the GET method.

3.5.1. getstatus

3.5.1.1. Request

```
https://data.talk2m.com/getstatus
```

3.5.1.2. Response

This returns information about the storage consumption of the account and of each eWON. The result contains the following information:

- number of history points currently stored in the DataMailbox for this account
- number of eWONs which sent data in the DataMailbox
- for each eWON:
 - the id of the eWON
 - the name of the eWON
 - the number of history points currently stored in the DataMailbox for this eWON
 - the date of the first history point currently saved in the DataMailbox for this eWON
 - the date of the last history point currently saved in the DataMailbox for this eWON

3.5.1.3. Example

```
https://data.talk2m.com/getstatus?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevide=1234
```

The result would be:

```
{  
  "historyCount": 20732,  
  "ewonsCount": 2,  
  "ewons":
```

```
[
  {
    "id":2,
    "name":"Paris",
    "historyCount": 2702,
    "firstHistoryDate": "2015-07-16T16:04:25Z",
    "lastHistoryDate": "2015-07-17T17:43:36Z"
  },
  {
    "id": 190,
    "name": "Brussels",
    "historyCount": 18030,
    "firstHistoryDate": "2015-08-24T08:56:44Z",
    "lastHistoryDate": "2015-08-24T14:57:13Z"
  }
]
```

3.5.2. getewons

3.5.2.1. Request

```
https://data.talk2m.com/getewons
```

3.5.2.2. Response

This service returns the list of eWONs that send data to be stored in the DataMailbox.

The result contains the following information:

- name and id of the eWONs
- number of tags
- date of the last data upload performed by the eWON to the Data Mailbox

3.5.2.3. Example

```
https://data.talk2m.com/getewons?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=61e9ddf4-6ade-493e-ad4ea79dde788140
```

To simplify the display of the code, we will replace the **t2mdevid= 61e9ddf4-6ade-493e-ad4ea79dde788140** parameter (which is long and unreadable) by **t2mdevid=1234**

3.5.3. getewon

3.5.3.1. Request

```
https://data.talk2m.com/getewon
```

3.5.3.2. Response

This returns the configuration of the eWON as seen by the DataMailbox.

The eWON can be identified using one of the following parameters :

- **id** : the ID of the eWON as returned by the getewons API call.
- **name** : the name of the eWON as returned by the getewons API call.

3.5.3.3. Example

```
https://data.talk2m.com/getewon?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&id=411
```

3.5.4. getdata

3.5.4.1. Request

```
https://data.talk2m.com/getdata
```

The *getdata* request can be filtered to retrieve only part of the mailbox contents. The filters consist in a series of string parameters:

- **eworldid**: (optional) Retrieve info only for a specific eWON.
- **tagid**: (optional) Retrieve info only for a specific tag
- **from**: (optional) time stamp after which data should be returned. No data older than this time stamp will be sent.
- **to**: (optional) time stamp before which data should be returned. No data newer than this time stamp will be sent.

- **fullConfig** (no value): (optional) By default, *getdata* returns config info only for eWONs/tags that contain historical data. Tags and eWONs with no historical data do not appear in the data set returned to the caller. When *fullConfig* is used in the request, all tags/eWONs appear in the data set, even if they do not contain any historical data.
- **Limit** : (optional) the maximum amount of historical data returned. If the size of the historical data saved in the Data Mailbox exceeds this limit, only the oldest historical data will be returned and the result contains a *moreDataAvailable* value indicating that more data is available on the server.
If the limit parameter is not used or is too high, the Data Mailbox uses a limit pre-defined in the system.

The parameters can be combined. For example :

- **eworld + from**: returns the historical data of one eWON since a given date

- Warning -

The *getdata* and *syncdata* are different and should not be interchanged.

Getdata is used as a "one-shot" to retrieve filtered data based on specific criteria. It is not destined to grab historical data with the same timestamp or enormous data that involves the use of the "moreData" flag.

Syncdata is used to retrieve all the data. This command is destined to grab the whole set of data regardless the amount.

3.5.4.2. Response

This returns the content of the DataMailbox : Configuration, tag history and alarm history.

The result of the *getdata* API call contains :

- A flag *moreDataAvailable* set to *true* if some historical data satisfying the criteria is available on the Data Mailbox but could not be returned. See the *limit* parameter.
- List of eWONs. For each eWON :
 - Id
 - eWON configuration (Id, name, data type and description)
 - Alarm hint
- List of tags. For each tag:
 - Id
 - Configuration (name, description, data type,...)
 - Last known value.

- Alarm state
- Quality (of instant value)
- History of tag values (date, value, quality)
- History of tag alarms

- Note -

*The quality of a tag can receive 3 possible values : **good**, **bad** and **uncertain**.
The quality of a historical tag value will only be displayed if quality is different from good
whereas quality of an instant tag value will always show up.*

3.5.4.3. Examples

```
https://data.talk2m.com/getdata?t2maccount=test-  
account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevId=1234&ewonId=411&tag  
Id=1
```

```
https://data.talk2m.com/getdata?t2maccount=test-  
account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevId=1234&from=2014-04-  
17T14%3A34%3A56Z
```

3.5.5. syncdata

This is used for data synchronization. It allows the retrieval of all data of a Talk2M account incrementally. Therefore, only new data is returned at each call to the API.

3.5.5.1. Request

```
https://data.talk2m.com/syncdata
```

`syncdata` has the following parameters:

- **lastTransactionId**: Retrieves data after the indicated transaction ID.
- **createTransaction**: Indicates to the server a new transaction should be created for this request.

- Warning -

The `getdata` and `syncdata` are different and should not be interchanged.

Getdata is used as a "one-shot" to retrieve filtered data based on specific criteria. It is not destined to grab historical data with the same timestamp or enormous data that involves the use of the "moreData" flag.

Syncdata is used to retrieve all the data. This command is destined to grab the whole set of data regardless the amount.

3.5.5.2. Response

The behaviour is the following:

- The first time the data is retrieved, the user specifies `createTransaction` but not `lastTransactionId`. The system returns all the data of the eWONs of the account and a transaction id.
- For the next calls to the API, the client specifies `createTransaction` and `lastTransactionId`, integrating the id of the transaction that was returned by the last `syncdata`. The system returns all the historical data that has been received by the datamailbox since the last transaction and of course a new transaction Id.

3.5.5.3. Examples

```
https://data.talk2m.com/syncdata?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&id=411&createTransaction=true
```

```
https://data.talk2m.com/syncdata?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&id=411&createTransaction=true&lastTransactionId=32745
```

The result is the same as the [getdata](#)'s one except this result can contain the following additional value:

- `transactionId`: this value contains the id of the transaction that was created on the API call you just sent to the server. This value is returned only when the `createTransaction` parameter is used.

As for `getdata`, a flag `moreDataAvailable` indicates whether a part of the data could not be returned. A new call to `syncdata` must then be done in order to retrieve additional data.

3.6. Delete the Data

As the DataMailbox is a temporary storage, the data must be deleted at one point. For example, it is common practice once the data has been retrieved and treated by your application to delete that data from the DataMailbox.

The deletion can be done either automatically or manually.

Automatically means that the DataMailbox keeps the data only for a short period of time. The DataMailbox was designed in order to offer a temporary storage to let you retrieve the data from a cloud solution.

If you don't erase the data when retrieving it then the DataMailbox will perform the delete request on its own after a limited time.

Or you can do it, by yourself, by sending the delete request explained here below.

3.6.1. delete

3.6.1.1. Request

```
https://data.talk2m.com/delete
```

The parameters provide some criteria on the data to delete:

- **all** (used without any value): Empty the data mailbox from historical data.
- **transactionId**: Delete all historical data previously retrieved using the transaction with this id.
- **eworldId**: Delete all historical data of this eWON.
- **to**: Delete all history data up to the given timestamp. Easily helps to get rid of old data.

3.6.1.2. Response

This returns a message with a success status. See [3.2.Response Structure](#).

3.6.1.3. Examples

```
https://data.talk2m.com/delete?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&eworldId=411
```

```
https://data.talk2m.com/delete?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&transactionId=4
```

3.6.2. clean

This is stronger than delete. Delete only deletes historical data. It preserves config data including indicators that allow eWONs to send only their “new” contents.

When using clean, even the config is deleted. It means that upon the next eWON Data upload, eWON will send its whole content again.

3.6.2.1. Request

```
https://data.talk2m.com/clean
```

Parameters:

- **all** (used without any value): Empty the data mailbox and its *metadata* (eWONs and tags config, DMBin related info).
- **ewonId**: Delete all historical and configuration data related to a specific eWON. Also deletes the eWON *metadata*.

3.6.2.2. Response

This returns a message with a success status. See [3.2.Response Structure](#).

3.6.2.3. Examples

```
https://data.talk2m.com/clean?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&all
```

```
https://data.talk2m.com/clean?t2maccount=test-account1&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234&ewonId=412
```



Revision

Revision History

Revision Level	Date	Description
1.0	08/06/2015	Original Document
1.1	24/06/2016	Added: GET or POST content
1.2	31/08/2016	Version adjustment to match SDK version
1.3	02/02/2018	ADDED: getstatus sub-chapter ADDED: Quality tag parameter of getData CHANGED: syncdata typo

Document build number: 129

Note concerning the warranty and the rights of ownership:

The information contained in this document is subject to modification without notice. Check <https://ewon.biz/support> for the latest documents releases.

The vendor and the authors of this manual are not liable for the errors it may contain, nor for their eventual consequences.

No liability or warranty, explicit or implicit, is made concerning the quality, the accuracy and the correctness of the information contained in this document. In no case the manufacturer's responsibility could be called for direct, indirect, accidental or other damage occurring from any defect of the product or errors coming from this document.

The product names are mentioned in this manual for information purposes only. The trade marks and the product names or marks contained in this document are the property of their respective owners.

This document contains materials protected by the International Copyright Laws. All reproduction rights are reserved. No part of this handbook can be reproduced, transmitted or copied in any way without written consent from the manufacturer and/or the authors of this handbook.

HMS Industrial Networks sa