



DMWeb API for DataMailbox

Talk2M SDK – 1.3.1.27564

REFERENCE GUIDE

RG-0005-00 1.8 en-US ENGLISH

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Industrial Networks of any inaccuracies or omissions found in this document. HMS Industrial Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Industrial Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Industrial Networks and is subject to change without notice. HMS Industrial Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

Table of Contents

Page

1	Preface	3
1.1	About this document	3
1.2	Document history	3
1.3	Related documents	3
1.4	Trademark information	3
2	Introduction.....	4
2.1	Prerequisites	4
2.2	Talk2M SDK.....	5
3	Ewon configuration	7
3.1	Data synchronizing parameters.....	7
3.2	Synchronization by script.....	8
3.3	Synchronization status	9
4	DMWeb API	10
4.1	API request structure	10
4.2	Response structure.....	10
4.3	Credentials	10
4.4	Date format	11
4.5	Data services.....	14

This page intentionally left blank

1 Preface

1.1 About this document

This document explains how to use the DMWeb API to generate RESTful API requests to the Talk2M DataMailbox and to easily retrieve data stored in the Talk2M DataMailbox.

For additional related documentation and file downloads, please visit developer.ewon.biz.

1.2 Document history

Version	Date	Description
1.0	2018-05-03	First release
1.1	2018-08-29	Added: GET or POST content.
1.2	2016-08-31	Changed: Version adjustment to match SDK version
1.3	2018-02-02	Added: getstatus sub-chapter Added: Quality tag parameter of getData Changed: syncdata typo
1.4	2018-11-09	Changed: Date format, p. 11
1.5	2019-01-21	Changed: Talk2M SDK version
1.6	2019-04-03	Changed: Examples in syncdata, p. 20
1.7	2019-07-31	Added: “ewonIds” parameter in syncdata, p. 20
1.8	2019-08-20	Changed: Encode URL parameters in Credentials, p. 10 .

1.3 Related documents

Document	Author	Document ID
General Reference Guide	HMS	RG-0001-00

1.4 Trademark information

Ewon® is a registered trademark of HMS Industrial Networks SA. All other trademarks mentioned in this document are the property of their respective holders.

2 Introduction

The DMWeb API is a RESTful web service combined with the Talk2M service “DataMailbox” which allows easy retrieval of Ewon device historical data.

Application developers can easily write code to retrieve historical data of multiple Ewon using the DataMailbox without the need of learning a whole new environment.

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

If the DataMailbox acts as a temporary storage, there is a limitation in the capacity that can be stored:

- 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 DataMailbox.

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

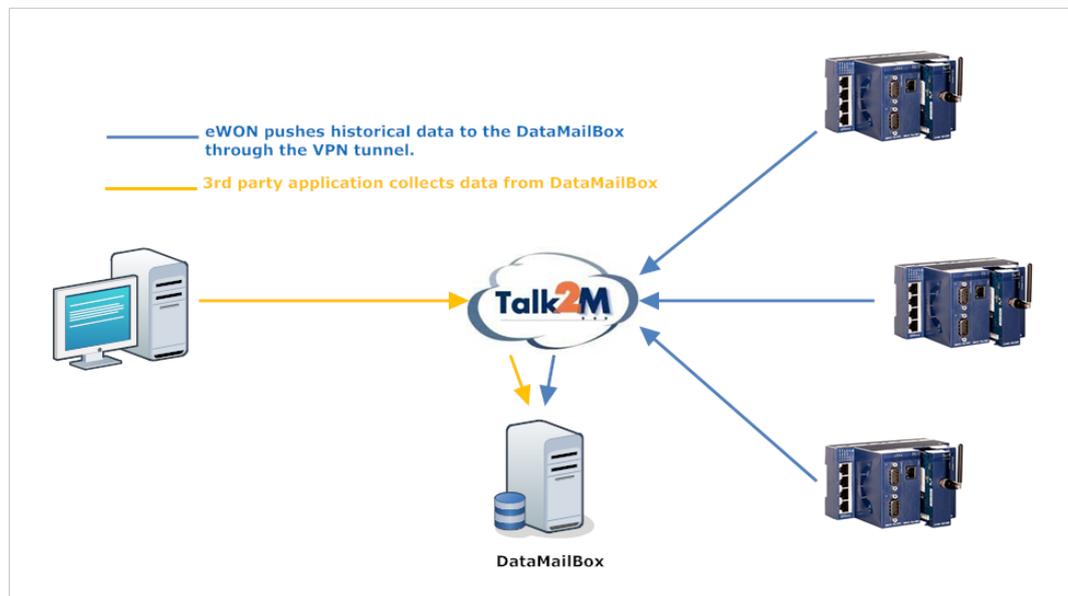


Fig. 1 General principle of DataMailbox

2.1 Prerequisites

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

- **Ewon device**
The Ewon needs to have the ability to record historical logs. This targets the CD 400# series or the Flexy family.
You then have to make sure the tags are correctly configured and that you have enabled this historical logging option.
- **Talk2M Account**
The Ewon which you want to retrieve 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 sending a web form on [eWON Developer](#) 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.

2.2 Talk2M SDK

This document is part of the Talk2M SDK which contains two different folders: *DataMailbox* and *M2Web API*.

2.2.1 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. For more information, please refer to [eWON Developers](#) or to the reference guide document located in the Talk2M SDK.

2.2.2 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 proposing which request should be sent 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#

3 Ewon configuration

3.1 Data synchronizing parameters

To enable the historical data of your Ewon in the DataMailbox, go to: **Setup > System > Main > Net services > Data Management**.

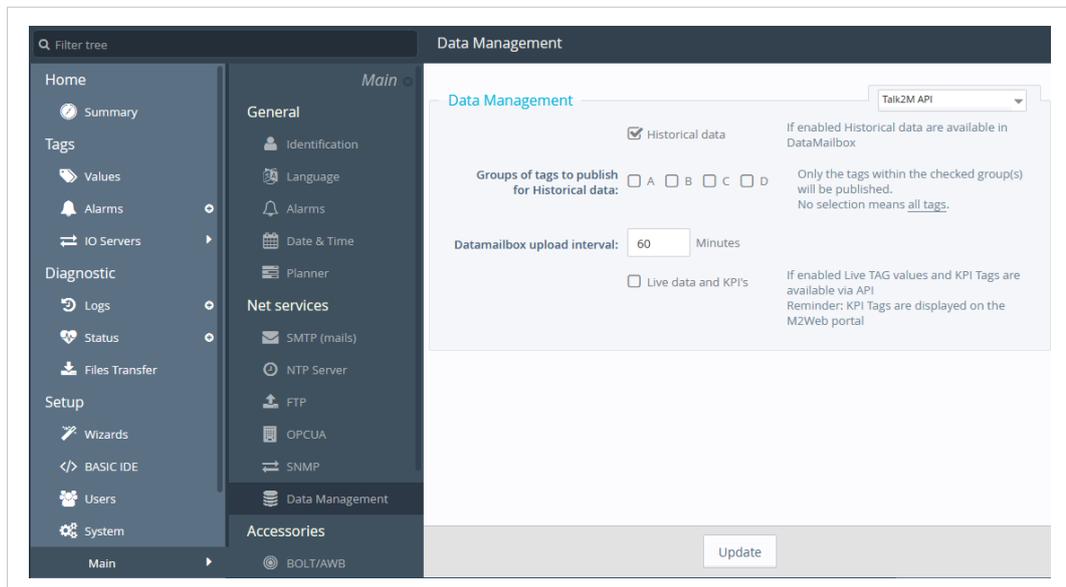


Fig. 2 Data Management web configuration

3.1.1 Talk2M API

As the Data Management is set to “Talk2M API”, check the “Historical data” box. This will tell your Ewon to send its historical data to the DataMailbox.

Several options can be modified:

Groups of tags to publish	This option allows the synchronization of 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. If no group is selected, than all tags are sent.
Upload interval	Specifies the synchronization interval. For example “45”, to synchronize every 45 minutes. Default value: 60.

The group of tags filter applies only to the historical stored values of the Ewon. The filter is not applied on the real-time and alarm values. Therefore, the real-time values and alarm info of all tags will still be synchronized.

3.1.2 eSync

As the Data Management is set to “eSync”, check the “Historical data” box. This will tell your Ewon to send its historical data to the server mentioned in the “Server URL” field.

Several options can be modified (in addition to the ones already explained in [Talk2M API, p. 7](#)):

Data Management ID	The username / ID used to log in to the server that will receive the historical data. This field is not required if the DataMailbox is the server.
Password	The password used to log in to the server that will receive the historical data. This field is not required if the DataMailbox is the server.
Server URL	The URL of the server that will receive the historical data. Default value: ewondata.talk2m.com (which is the URL to the DataMailbox).
Advanced data transfer schedule	This field allows the configuration of 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 (Related documents, p. 3).
Upload on alarm	When selected, the data synchronization will also be triggered when one of the tags of the selected groups triggers an alarm.

3.2 Synchronization by script

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

The BASIC function “DMSync” triggers the Ewon to synchronize its data.

To trigger a single synchronization, you can proceed as follows:

- open the Flexy Basic IDE section,
- make sure the console frame is opened,
- Type “DMSync” in the console frame and hit “Enter”.

A data synchronization is then immediately initiated.

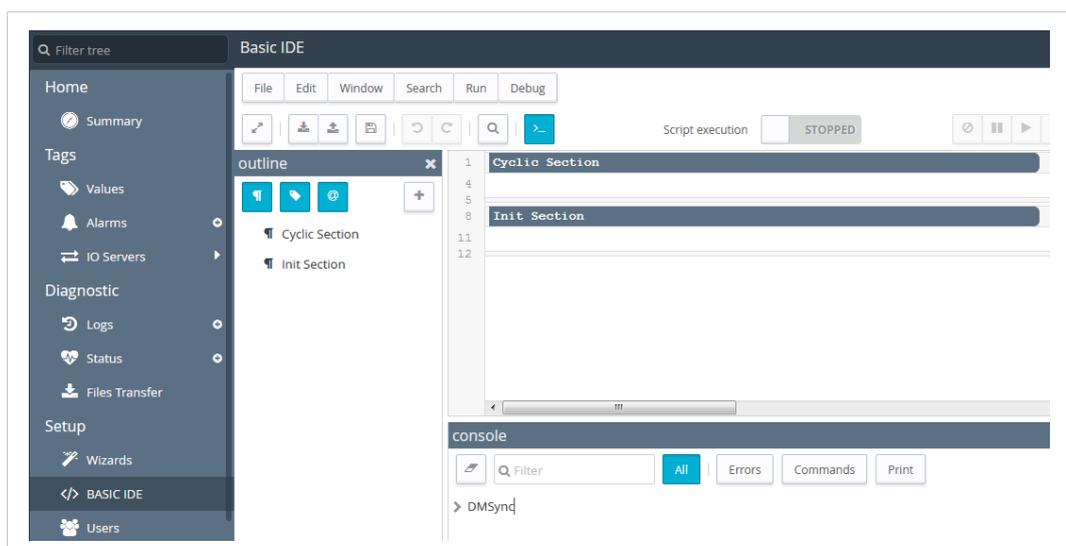


Fig. 3

3.3 Synchronization status

To check if the synchronization succeeded, open the Ewon diagnostic window by going to: **Diagnostic > Logs > Event Logs**.

Make sure the level of the logs is set *Trace*.

The event log should contain 2 messages:

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

If so, the Ewon has transferred the data to the DataMailbox. You can now have a look on the server side to visualize the received data.

4 DMWeb API

The DMWeb API is used to retrieve the historical data of all your Ewons from the DataMailbox. Using this API, you will be able:

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

4.1 API request structure

The hostname of the DMWeb API is:

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

4.2 Response structure

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

success	A boolean value. It is set to <i>true</i> when the API was treated without any issue. It is set to <i>false</i> when an error occurred during the treatment of the response.
code	An integer The error code displayed when the request didn't succeed. This field is shown only when "success" returns <i>false</i> .
message	A string The description explaining the issue that occurred. This field is shown only when "success" returns <i>false</i> .

4.3 Credentials

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



The "t2mdevid" is a Talk2M Developer ID specific to the DataMailbox (and M2Web). To request one, fill in the web form that can be found on [eWON Developers](#) website.



Those parameters, especially the password parameter, must be encoded as UTF-8 strings. The way to "percent encode" a UTF-8 character is to encode each bytes of its UTF-8 sequence (e.g.: "é" becomes "%C3%A9", "ø" becomes "%E2%8D%B5", "#" becomes %23).

4.3.1 GET or POST

Both GET and POST protocols are secure when using the DMWeb 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.

In the examples available in 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" or "t2mpassword".

4.4 Date format

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

The JSON feed sent by the DataMailbox API responses shows the dates under the following format:

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

For the dates used as parameters sent to the DataMailbox, 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
```

The dates used in the URL must be encoded in a special format. The colon ":" character needs to be encoded as "%3A":

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

4.4.1 Ewon timestamps logged in UTC

Before firmware 13.2, the Ewon is always logging data in local time.

As of firmware 13.2, the Ewon has the option to record data using UTC timestamps.

To switch to UTC timestamps, enable the "Record data in UTC" option which is represented by a checkbox in the configuration of your Ewon date & time.

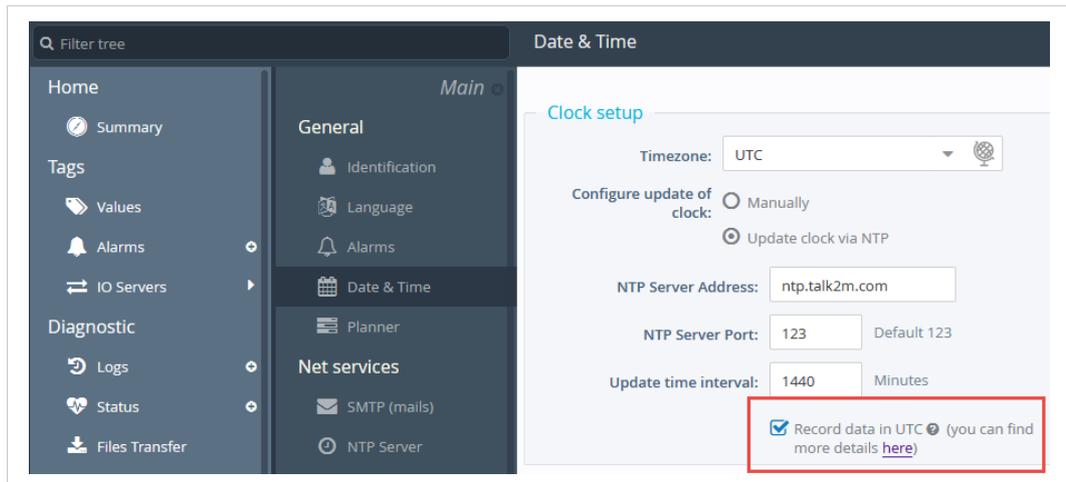


Fig. 4 Record data using UTC timestamps

If the “Record data in UTC” option is checked, the DMWeb API will add an extra parameter in its JSON feed. This extra parameter tells the applications using the DMWeb API that the data is recorded in UTC.

This extra parameter is called “timeZone” and returns the UTC + timezone of your Ewon.

If your Ewon does not record in UTC, as the “Record data in UTC” option is unchecked, the “timeZone” parameter will not appear in the DMWeb JSON feed.



The value of the “timeZone” parameter is based on TZ time zone. For more information, check https://www.wikiwand.com/en/List_of_tz_database_time_zones.

This parameter comes with conditions:

- The time zone is maximum 40 characters long.
- The format of the dates does not change. It remains the same regardless how the data is logged in the Ewon.

Request concerned by the “timeZone” parameter

The following DMWeb API requests will have the “timeZone” parameter if the “Record data in UTC” option – from the Ewon date & time configuration – is enabled.

getewons

This request returns the “timeZone” field (if available) for each Ewon listed in the response feed.

Example 1: “timeZone” inside the getewons response

```
{
  "success":true,
  "ewons":[
    {
      "id":3003,
      "name":"ExampleName",
      "timeZone":"Europe/Brussels",
      "lastSynchroDate":"2018-10-30T16:02:01Z"
    }
  ]
}
```

getewon

This request returns the “timeZone” field (if available) of a specific Ewon listed in the response feed.

Example 2: “timeZone” inside the getewon response

```
{
  "success":true,
  "id":3003,
  "name":"ExampleName",
  "tags":[
    {
      "id":3223,
      "name":"TagName",
      "dataType":"Float",
      "description":"Description for tag",
      "alarmHint":"",
      "value":19.0,
      "quality":"good",
      "ewonTagId":3
    }
  ],
  "timeZone":"Europe/Brussels",
  "lastSynchroDate":"2018-10-30T16:02:01Z"
}
```

getdata

This request returns the “timeZone” field (if available) of a specific Ewon listed in the response feed.

Example 3: “timeZone” inside the getdata response

```
{
  "success":true,
  "ewons":[
    {
      "id":3003,
      "name":"ExampleName",
      "tags":[
        {
          "id":3223,
          "name":"TagName",
          "dataType":"Float",
          "description":"Description for tag",
          "alarmHint":"",
          "value":19.0,
          "quality":"good",
          "ewonTagId":3,
          "history":[
            {
              "date":"2014-04-16T16:34:16Z",
              "value":"4.0"
            },
            {
              "date":"2014-04-17T16:34:26Z",
              "value":5.0
            }
          ]
        }
      ]
    }
  ],
  "timeZone":"Europe/Brussels",
}
```

```

    "lastSynchroDate": "2018-10-30T16:02:01Z"
  }
]
}

```

syncdata

This request returns the “timeZone” field (if available) of a specific Ewon listed in the response feed.

Example 4: “timeZone” inside the syncdata response

```

{
  "success": true,
  "ewons": [
    {
      "id": 3003,
      "name": "ExampleName",
      "tags": [
        {
          "id": 3223,
          "name": "TagName",
          "dataType": "Float",
          "description": "Description for tag",
          "alarmHint": "",
          "value": 19.0,
          "quality": "good",
          "ewonTagId": 3,
          "history": [
            {
              "date": "2014-04-16T16:34:16Z",
              "value": "4.0"
            },
            {
              "date": "2014-04-17T16:34:26Z",
              "value": 5.0
            }
          ]
        }
      ]
    },
    {
      "timeZone": "Europe/Brussels",
      "lastSynchroDate": "2018-10-30T16:02:01Z"
    }
  ]
}

```

4.5 Data services

4.5.1 getstatus

Request

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

Response

The “getstatus” service returns 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 send data to the DataMailbox.

The result also contains the following information for each Ewon listed:

- its id,
- its name,
- its number of history points currently stored in the DataMailbox,
- its date of the first history point currently saved in the DataMailbox,
- its date of the last history point currently saved in the DataMailbox.

Example 5: getstatus request

```
https://data.talk2m.com/getstatus?t2maccount=test-account1
&t2musername=admin&t2mpassword=aaaa.1234&t2mdevId=1234
```

Example 6: getstatus response

```
{
  "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"
    }
  ]
}
```

4.5.2 getewons

Request

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

Response

The “getewons” service returns the list of Ewons sending data to be stored in the DataMailbox.

The result contains the following information for each Ewon:

- its name and id,
- its number of tags,
- the date of its last data upload to the Data Mailbox.

Example 7: getewons request

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

Example 8: getewons response

```
{  
  "success": true,  
  "ewons": [  
    {  
      "id": 538608,  
      "name": "ltn_205",  
      "lastSynchroDate": "2018-08-10T13:14:13Z"  
    },  
    {  
      "id": 632943,  
      "name": "4G_test_Mobistar_eupen",  
      "lastSynchroDate": "2018-07-10T16:35:26Z"  
    },  
    {  
      "id": 508238,  
      "name": "ltn_flexy",  
      "lastSynchroDate": "2018-11-08T16:46:19Z",  
      "timeZone": "UTC"  
    }  
  ]  
}
```

4.5.3 getewon

Request

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

The Ewon can be identified using one of the following filter:

id	ID of the Ewon as returned by the “getewons” API request.
name	Name of the Ewon as returned by the “getewons” API request.

Response

The “getewon” service returns the configuration of the targeted Ewon as seen by the DataMailbox.

Example 9: getewon request

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

Example 10: getewon response

```
{
  "success": true,
  "id": 508238,
  "name": "ltn_flexy",
  "tags": [
    {
      "id": 780589,
      "name": "TAG_1",
      "dataType": "Float",
      "description": "",
      "alarmHint": "",
      "value": 1502,
      "quality": "good",
      "ewonTagId": 1
    },
    {
      "id": 780591,
      "name": "TAG_2",
      "dataType": "Float",
      "description": "",
      "alarmHint": "",
      "value": 1506,
      "quality": "good",
      "ewonTagId": 2
    }
  ],
  "lastSynchroDate": "2018-11-08T16:56:35Z",
  "timeZone": "Europe/Brussels"
}
```

4.5.4 getdata

Request

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



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

getdata is used as a “one-shot” request to retrieve filtered data based on specific criteria. It is not destined to grab historical data with the same timestamp or enormous data involving the use of the “moreData” filter.

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

You can filter the “getdata” request to retrieve only part of the mailbox contents. The filters consist of a series of string parameters and are all optional:

ewonId	ID of the singleEwon for which data from DataMailbox is requested.
tagId	ID of the single tag for which data from DataMailbox is requested.
from	Timestamp after which data should be returned. No data older than this time stamp will be sent.
to	Timestamp before which data should be returned. No data newer than this time stamp will be sent.
fullConfig	By default, <i>getdata</i> returns configuration information only for Ewons / tags that contain historical data. If the request contains “fullConfig” as parameter, all tags / Ewons will appear in the data set, even if they do not contain historical data. “fillConfig” doesn’t accept any value. It is used as is.
limit	The maximum amount of historical data returned. If the size of the historical data saved in the DataMailbox exceeds this limit, only the oldest historical data will be returned and the result contains a <i>moreDataAvailable</i> value indicating that more data is available on the server.If the limit parameter is not used or is too high, the DataMailbox uses a limit pre-defined in the system.

The parameters can be combined. For example: **ewonId + from** returns the historical data of an Ewon since a given date

Response

The “getdata” service returns the content of the DataMailbox: configuration, tag history and alarm history.

The result of the “getdata” API call contains, in a general scope:

- a flag “moreDataAvailable” set to *true* if some historical data satisfying the criteria is available on the DataMailbox but could not be returned. See the *limit* parameter;
- a list of Ewons.

The result also contains the following information for each Ewon listed:

- its ID;
- its configuration: ID, name, data type and description;
- its alarm hint;
- its list of tags.

The result also contains the following information for each tag of each Ewon listed:

- its ID as registered in the DataMailbox;
- its configuration: name, data type and description;
- its alarm state;
- its last known value;
- the quality of the instant value;
- the history of tag values: date, value and quality;
- the history of tag alarms: date, status and type.

The quality of a tag can receive 3 possible values: good, bad and uncertain. It is always shown in the feed.

The quality of an historical tag value will only be displayed if quality is different than good.

Example 11: getdata request

```
https://data.talk2m.com/getdata?t2maccount=test-account1
&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234
&ewonId=508238&tagId=1
```

```
https://data.talk2m.com/getdata?t2maccount=test-account1
&t2musername=admin&t2mpassword=aaaa.1234&t2mdevid=1234
&from=2014-04-17T14%3A34%3A56Z
```

Example 12: getdata response

```
{
  "success": true,
  "moreDataAvailable": true,
  "ewons": [
    {
      "id": 508238,
      "name": "ltn_flexy",
      "tags": [
        {
          "id": 780591,
          "name": "TAG_2",
          "dataType": "Float",
          "description": "",
          "alarmHint": "",
          "value": 1510,
          "quality": "good",
          "ewonTagId": 2,
          "history": [
            {
              "date": "2018-11-08T14:17:58Z",
              "quality": "initialGood",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:00Z",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:02Z",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:04Z",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:06Z",
              "value": 0
            }
          ]
        }
      ]
    }
  ],
  "lastSynchroDate": "2018-11-09T09:47:00Z",
  "timeZone": "Europe/Brussels"
}
```

4.5.5 syncdata

Request

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



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

getdata is used as a “one-shot” request to retrieve filtered data based on specific criteria. It is not destined to grab historical data with the same timestamp or enormous data involving the use of the “moreData” filter.

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

The “syncdata” service retrieves all data of a Talk2M account incrementally. Therefore, only new data is returned on each API request.

You can filter the request with the following parameters:

lastTransactionId	The ID of the last set of data sent by the DataMailbox. By referencing the “lastTransactionId”, the DataMailbox will send a set of data more recent than the data linked to this transaction ID.
createTransaction	The indication to the server that a new transaction ID should be created for this request.
ewonIds	A comma separated list of Ewon IDs. If “ewonIds” is used, DataMailbox sends values history of the targeted Ewons. If not used, DataMailbox sends the values history of all Ewons.

As a “syncdata” request should be used to retrieve only incremental data, the normal process of the “syncdata” requests is:

- When the data is retrieved for the first time, the user specifies only the “createTransaction” filter.
The DataMailbox returns all the data of all Ewons – with historical data – of the account along a transaction ID.
- For the next calls to the API, the client specifies both “createTransaction” and “lastTransactionId” filters. The “lastTransactionId” is the ID of the transaction that was returned by the latest *syncdata* request.
The system returns all the historical data that has been received by the DataMailbox since the last transaction and a new transaction ID.
- *ewonIds* is an additional filter on the returned result.

You must be cautious when using the combination of *lastTransactionId*, *createTransaction* and *ewonIds*.

lastTransactionId is first used to determine what set of data — newer than this transaction ID and from all the Ewons — must be returned from the DataMailbox, then *ewonIds* filters this set of data to send data only from the desired Ewons.

If a first request is called with *lastTransactionId*, *createTransaction* and *ewonIds*, the following request — implying a new *lastTransactionId* — does not contain values history from the previous *lastTransactionId* of the Ewons that were not in the *ewonIds* from previous request.

Response

The result of the “syncdata” request is the same as the “getdata” request. However, the “syncdata” result can contain an additional value:

transactionId	The transaction ID of the “syncdata” response. This value is returned only when the “createTransaction” filter is used and should be mentioned as “lastTransactionId” in the following “syncdata” request to retrieve only new data.
----------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

As for “getdata”, a flag “moreDataAvailable” indicates whether a part of the data could not be returned. A new call to *syncdata* with a new transaction ID must be done to retrieve the additional data.

Example 13: syncdata request

```
https://data.talk2m.com/syncdata?t2maccount=test-account1
&t2musername=admin&t2mpassword=aaaa.1234&t2mdevId=1234
&createTransaction=true

https://data.talk2m.com/syncdata?t2maccount=test-account1
&t2musername=admin&t2mpassword=aaaa.1234&t2mdevId=1234
&createTransaction=true&lastTransactionId=32745
```

Example 14: syncdata response

```
{
  "success": true,
  "moreDataAvailable": true,
  "ewons": [
    {
      "id": 508238,
      "name": "ltn_flexy",
      "tags": [
        {
          "id": 780591,
          "name": "TAG_2",
          "dataType": "Float",
          "description": "",
          "alarmHint": "",
          "value": 1510,
          "quality": "good",
          "ewonTagId": 2,
          "history": [
            {
              "date": "2018-11-08T14:17:58Z",
              "quality": "initialGood",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:00Z",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:02Z",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:04Z",
              "value": 0
            },
            {
              "date": "2018-11-08T14:18:06Z",
              "value": 0
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
    ],
    "lastSynchroDate": "2018-11-09T09:47:00Z",
    "timeZone": "Europe/Brussels"
  }
]
}

```

4.5.6 delete

Request

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

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

Deleting data can be done either automatically or manually.

As the DataMailbox keeps the data for 10 days maximum, it erases automatically any data with a timestamps longer than those 10 days.

Or you can do it manually by sending the delete request.

Delete only deletes historical data. It preserves configuration data including indicators that allow Ewons to send only their new content.

You can filter the request with the following parameters:

all	Empty the data mailbox from historical data. This parameter is used as standalone. Value is not needed.
transactionId	Delete all historical data prior to this transaction ID.
ewonId	Delete all historical data of the Ewon.
to	Delete all history data up to the given timestamp.

Response

The result of the “delete” request is a message with a success status. Check [Response structure](#), p. 10.

Example 15: delete request

```

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

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

```

Example 16: delete response

```

{
  "success": true
}

```

4.5.7 clean

Request

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

The “clean” request is stronger than the “delete” request. Using clean, even the configuration is deleted. Upon next data upload to the DataMailbox, your Ewon will send its whole content again.

You can filter the request with the following parameters:

all	Empty the DataMailbox and its metadata such as Ewons and tags config, DMBin related info. This parameter is used as standalone. Value is not needed.
ewonId	Delete all historical and configuration data related to a specific Ewon. Also deletes the Ewon metadata.

Response

The result of the “clean” request is a message with a success status. Check [Response structure, p. 10](#).

Example 17: clean request

```
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
```

Example 18: clean response

```
{
  "success": true
}
```

