M2Web API

Talk2M SDK – 1.3.1.27564

REFERENCE GUIDE

RG-0004-00 1.5 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.

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
#	2013-09-23	First release
#	2013-11-13	Changed: Clarified Developer ID
1.0	2014-07-03	Added: Account Type property
1.1	2015-01-21	Added: APIs examples
1.2	2016-08-31	Changed: Talk2M SDK version
1.3	2018-02-02	Changed: Talk2M SDK version
1.4	2019-01-21	Changed: Talk2M SDK version
1.5	2019-08-20	Changed: Encode URL parameters in *Session Management, p. 9* and *Stateless Requests, p. 7*.

1.3 Related documents

Document	Author	Document ID
Web Reference Guide	HMS	RG-0003-00

1.4 Trademark information

2 Introduction

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 (basically the contents found in the M2Web portal) and access to the Ewon web server (and possible web servers on Ewon LAN).

Examples of features of the API:

- Stateless M2Web sessions: one can *POST* or *GET* to the Ewon using a single API call.
- Stateful requests (explicit session management) remain supported.
- Large volume applications: The API allows concurrent access to different Ewons such as *simultaneously* sending a single request to a large set of Ewons. The API (*stateless* mode) is not subject to concurrent M2Web connections.
- The information available in the portal can be retrieved in a machine-friendly format (JSON). This mostly consists in a list of Ewons and their state (online, offline, send wake-up SMS,...).

2.1 Keywords

The following keywords appear in the document:

ESP	Ewon Solution Provider. A third-party supplier (hardware, software, services,…) that delivers business solutions involving Ewons.
DevID	Talk2M Developer ID. A unique identifier owned by every developer of a program or application that uses the M2Web API. DevID example: *21EC2020-3AEA-1069-A2DD-08002B30309D*.

3 Talk2M Developer

3.1 Registration

All Talk2M users or third parties (such as ESPs) that wish to use a Talk2M API such as the M2Web API must first register as a Talk2M developer.

To do so, fill in the web form that can be found on Ewon Developers website. As a result of the registration process, the registrant receives a unique identifier.

In this document, we call it the Talk2M Developer ID, or DevID for short version.

The DevID must be specified in every M2Web API call in order to identify the 3rd party. This is vital as M2Web needs to validate the DevID to send back data: API calls with missing or incorrect DevID are obviously rejected.

3.2 Talk2M SDK

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

3.2.1 DataMailbox

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

For more information, please refer to Ewon Developers website.

3.2.2 M2Web API

The M2Web API is explained in the present document. The M2Web API folder from the Talk2M SDK should contain:

- **This Reference Guide**
- **NET**

  This is actually a library that helps you create .NET (C#/VB) programs using the M2Web API. It was programmed using Visual Studio 2010. Open the Solution file (M2Web.sln) and read the contents of Example1/Program.cs to see how to use the library.
- **PHP**

  Sample M2Web Library source code. Delivered as M2Web API sample code.

4 General Principles

4.1 HTTPS API

The M2Web API is a set of HTTPS requests.

The API supports both *GET* and *POST* methods. API parameters can be put either through the query string part of the URL (*GET*) or through the request body (*POST*).

The API exposes services that can either return information about M2Web itself (account information, Ewon connectivity status,...) or forward requests to Ewons (or Ewon LAN devices).

4.1.1 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.

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 and t2mdevicepassword.

4.2 Stateless Requests vs Sessions (aka Stateful Sequests)

Programs must login exactly as a human user of M2Web would do which implies the creation of a session.

M2Web API users may choose between two session models:

1. The traditional login/logout session where a ***sessionId*** is returned by the login service. This ***sessionId*** must be in every subsequent request.

   Sessions are closed by calling the logout service or after an inactivity timeout.

   This is known as *stateful requests*.

2. Stateless requests: There is no login/logout. Instead of passing a session id to the web services, the consumer sends its credentials (account, username and password) in every call. This model frees the developer from maintaining sessions.

Both models allow using the same set of API web services. The consumer simply chooses one model or the other by passing either credentials or a session id to those web services.

4.3 API Request Structure

As a reminder to *GET or POST, p. 6*, we recommend using the POST query.

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

4.3.1 URL

All API calls start with **t2mapi/**

```
https://m2web.talk2m.com/t2mapi/...
```

4.3.2 Request Parameters

API request parameters are passed as query string arguments. Their name starts with ***t2m*** in order to avoid collisions with application parameters that must be forwarded to the Ewon.

All requests require a mandatory ***t2mdeveloperid*** parameter that identifies the 3rd party (not the T2M account!). Requests with missing or incorrect DevID are rejected by the M2Web server. To request a Talk2M Developer ID, please refer to *Talk2M Developer, p. 5*.

All requests require credential parameters (either a session id in case of stateful request or identity parameters in case of stateless or login requests).

4.4 API Response Structure

4.4.1 Response

All the responses generated by the M2Web server itself (as opposed to responses sent by the Ewon and forwarded by the M2Web server) are formatted in:

- JSON (http://www.w3schools.com/json/).
- MIME type = *application/json*
- Encoding = UTF-8

All responses contain a boolean ***success*** parameter.

Example of successful response (HTTP Status code = 200):

```
{
 "success": true,
 "t2msession":
}
```

4.4.2 Errors

In case of error, the response is sent with an HTTP status code that reflects the error.

The response created by the M2Web server (as opposed to responses sent by the Ewon and forwarded by the M2Web server) indicate errors as follows:

- HTTP Status Code indicates error (!= 2xx)
- The response body consists in a JSON object that contains detailed error information:
  - “success”: false
  - “code”: x (same as HTTP status code)
  - “message”: "human readable error message"

Example of error response (HTTP status code != 2xx):

```
{
 "success": false,
 "code": 403,
 "message": "Invalid credentials"
}
```

The full list of error codes is documented in *Error Codes, p. 17*.

4.5 Stateless Requests

Those requests do the following in one single HTTPS request:

- Log in M2Web and create the session.
- Send the request to the Ewon (or to the portal, or to the device behind the Ewon) and return the reply.
- Close the session.

Any M2Web request can be turned into a stateless request by adding the following parameters in the query string:

```
?t2maccount=...&t2musername=...&t2mpassword=...
```

Those parameters are caught on-the-fly by M2Web and removed from the query string forwarded to the Ewon. M2Web validates those parameters and forwards the request to the Ewon as if there were a session cookie.

> **!** 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”, “ω” becomes “%E2%8D%B5”, “#” becomes %23.

4.6 Session Management

Session management is also known as “stateful requests”

When the session model is used (as opposed to stateless requests), the session id obtained from the *login* API call must be passed as a query string parameter to every subsequent API call:

```
?t2msession=...
```

4.7 Talk2M Developer ID

In addition to Talk2M user credentials, all API calls must contain the “t2mdeveloperid=...” query string parameter that identifies the program author as a registered Talk2M developer.

The Talk2M Developer ID is a unique string received upon registration. For more information, please refer to *Talk2M Developer, p. 5*.

DevID example: 21EC2020-3AEA-1069-A2DD-08002B30309D

The M2Web API uses the GUID notation without curly braces.

Every M2Web API call must contain this DevID as a query string parameter (in addition to other query string parameters such as credentials):

```
?t2mdeveloperid=...
```

4.8 No Rewriting of Ewon Response

The regular M2Web service rewrites absolute links in HTTP responses (HTML, CSS,...) returned by Ewons to make those links work through M2Web.

Since the M2Web API is aimed at providing machine-to-machine communication rather being used by a browser, there is no such rewriting mechanism.

5 Services

All URLs start with the same beginning:

```
https://m2web.talk2m.com/t2mapi/
```

In order to simplify the notation in this document, we no longer indicate the https protocol and hostname part of the URL.

5.1 Session Management

5.1.1 Login

This service is similar to the existing one used by the M2Web UI. However, the response is M2M-oriented: It returns a success/error indication (+session id) rather than redirecting to the account page.

> **!** The parameters used to initiate the session, 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", "ω" becomes "%E2%8D%B5", "#" becomes %23.

Request:

```
t2mapi/login?t2maccount=...&t2musername=...&t2mpassword=...
```

Response:

```
{
  "success": true,
  "t2msession": "123abc456"
}

or

{
  "success": false,
  "code": 403,
  "message": "Invalid credentials"
}
```

The values for t2msession is an example. Its actual value of course varies from call to call.

The session can be used only from the IP address that created it.

5.1.2 Logout

Closes an existing stateful API session.

> *Stateless requests do not require to log out.*

Request:

```
t2mapi/logout?t2msession=...
```

Response:

```
{
```

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

Success can be false such as in case of a wrong t2msession value. See above for the structure of a response that indicates an error.

5.2 Portal Services

All requests must contain either a *t2msession* parameter (stateful request) or the *t2maccount*, *t2musername* and *t2mpassword* credentials parameters (stateless request).

As all API calls, it is also available as a stateless call:

In addition they must also contain the *t2mdeveloperid* parameter.

5.2.1 getaccountinfo

The “getaccountinfo” command retrieves the basic account information (reference, name, company).

It also retrieves the set of pools visible by user : pairs name/id as well as the name of each custom attribute.

Request:

```
t2mapi/getaccountinfo?t2msession=...
t2mapi/getaccountinfo?t2maccount=...&t2musername=...&t2mpassword=...
```

Response:

```
{
  "success": true,
  "accountReference": "8435",
  "accountName": "ACME",
  "company": "ACME, Inc.",
  "customAttributes": [ "attribute1", "", "third attribute" ],
  "pools": [
      { "id": 8735, "name": "site 1" },
      { "id": 9541, "name": "site 2" },
      { "id": 723, "name": "devices" }
      ],
  "accountType": "Pro"
}
```

There are always 3 custom attributes listed. Some or all of them may be empty.

The “accountType” attribute will either be “Free” (for non paying account) or “Pro” (for paying account).

5.2.2 getewons

The “getewons” command retrieves the set of Ewons visible by user along with their properties: displayable names, link names, status, description, the 3 custom attributes, preferred m2web server hostname (currently always m2web.talk2m.com).

Request:

```
t2mapi/getewons?t2msession=...[&pool=...]
```

where “pool” is the optional numerical id of the pool from which theEwons should be retrieved. Pool id's are retrieved using a “getaccountinfo” call.

If “pool” is not specified, all visible Ewons are listed in the response.

Response:


```
{
  "success": true,
  "ewons": [
      { "id": 14235,
        "name": "IP Camera",
        "encodedName": "IP%20Camera",
        "status": "online",
        "description": "IP Camera displaying Hall 47",
        "customAttributes": [ "Omron CJ1G", "LAN", "192.168.140.3" ],
        "m2webServer": "m2web.talk2m.com"
      },
      { "id": 14758,
        "name": "Robot-T800",
        "encodedName": "Robot-T800",
        "status": "offline",
        "description": "Robot – 1st generation terminator",
        "customAttributes": [ "Arnold", "", "No problemo" ],
        "m2webServer": "m2web.talk2m.com"
      }
      ]
}
```


The following paramaters are explained

encodedName	The URL-friendly version of the name intended to be used in the “get” service.
m2webServer	The hostname to be used when using the “get” service to retrieve information from this Ewon.

The “accountType” attribute will either be “Free” (for non paying account) or “Pro” (for paying account).

5.2.3 getewon

The “getewon” retrieves the same set of information as “getewons” but is limited to one single Ewon.

Request:

```
t2mapi/getewon?name=...&t2msession=...
```

where “name” is the value as the name of the Ewon.

Another possibility to identify the Ewon is to use the “id” parameter instead of “name”. The Ewon id is returned by “getewons”.

Response:


```
{
  "success": true,
  "ewon" : {
        "id": 14235,
        "name": "IP Camera",
        "encodedName": "IP%20Camera",
```

```
        "status": "online",
        "description": "IP Camera displaying Hall 47",
        "customAttributes": [ "Omron CJ1G", "LAN", "192.168.140.3" ],
        "m2webServer": "m2web.talk2m.com"
      }
}
```

5.2.4 wakeup

The “wakeup” command sends a wake-up SMS to the Ewon.

The Ewon that needs the wake-up is indicated either through its *name* or its *id*.

Request:

```
t2mapi/wakeup?name=...&t2msession=...
t2mapi/wakeup?id=...&t2msession=...
```

where *name* is the name of the Ewon or *id* is the Ewon id returned by the “getewons” command.

Response:

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

5.2.5 sendoffline

The “sendoffline” command sends a cellular-based Ewon offline.

The Ewon is indicated either through its *name* or its *id*.

Request:

```
t2mapi/sendoffline?name=...&t2msession=...
t2mapi/sendoffline?id=...&t2msession=...
```

where *name* is the name of the Ewon or *id* is the Ewon id returned by the “getewons” command.

Response:

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

5.3 Reaching Ewons

5.3.1 URL

The URL used to reach an Ewon page, through a GET or POST request, is different from the browser-based M2Web URL scheme used when browsing the Ewon as stateful:

```
t2mapi/get/<ewon name>/<path>?<query>&t2msession=...
```

than it is when sending the request as stateless:

```
t2mapi/get/<ewon name>/<path>?<query>&t2maccount=...
```

The URL part as of *<path>* is forwarded to the Ewon. All *t2mxxx* parameters are deleted from the requested URL before forwarding it to the Ewon.

Reaching devices behind the Ewons is done using the same URL syntax as in a browser:

```
t2mapi/get/<ewon name>/proxy/<device IP>/<path>?<query>&t2msession=...
```

This “get” service is used for both GET and POST requests.

As indicated in a former paragraph, the Ewon response is not rewritten. It is therefore important to make sure that the Ewon’s response does not contain any absolute URL.

5.3.2 Hostname

The hostname for M2Web is:

```
https://m2web.talk2m.com
```

The Talk2M servers farm contains several M2Web servers. Talk2M decides which server should be used to reach each Ewon.

The hostname of each Ewon is contained in the *m2webServer* property of the “getewon” and the “getewons” services.

The default value remains *m2web.talk2m.com* but may change at any time in the future.

Example 1: Hostname of an Ewon

```
// If "m2WebServer" = "us.m2web.talk2m.com"
// The URL to reach the Ewon becomes:
https://us.m2web.talk2m.com/t2mapi/get/<ewon name>/
<path>?<query>&t2maccount=...
```

M2Web does not use HTTP redirections to handle such redirections because not all client-side programs technologies support HTTP redirections.

Sessions cannot be used across servers. A program that uses sessions (aka stateful requests) must log in again on the other M2Web server (*us.m2web.talk2m.com* in this example) before it uses the “get”service.

The *m2webServer* is a per-Ewon setting. It may change at any time (but is not likely to) as it is related to criteria such as the geographical location of the Ewon and the Talk2M account type (Free+ vs Pro).

5.3.3 Ewon Authentication

Retrieving information from Ewons requires to log in in the Ewon.

The “get” API accepts 2 additional parameters to create the Ewon login:

```
?...&t2mdeviceusername=...&t2mdevicepassword=...
```

M2Web will create the appropriate *Authorization* HTTP header for the given *t2mdeviceusername* & *t2mdevicepassword* parameters.

6 Ewon Services

Ewon web services exist in many different variations, such as:

- Standard web services (e.g.: *UpdateTagForm*)
- Standard data download using Export Block Descriptors
- Custom web pages using SSI tags. Those pages can be used to read and/or write data to the Ewon

Those services help retrieve or update various kinds of information, including:

- Ewon configuration, such as the list of tags,
- instantaneous tag values,
- tags history,
- alarms definition and state. Acknowledgment of alarms,
- remote execution of scripts.

Detailed information about those techniques can be found in the Web Reference Guide from the Ewon Developer website.

This page intentionally left blank

A Error Codes

Error codes and messages can be returned by M2Web, as described in *API Response Structure, p. 7*:

The table of error codes is the following one:

Code	HTTP Status	Message	Context (API)
400	Bad Request	Service Unavailable	get (with proxy)
400	Bad Request	Missing argument	all
403	Forbidden	Invalid method	N/A
403	Forbidden	Invalid session ID	All (stateful)
403	Forbidden	Invalid credentials	All (stateless)
403	Forbidden	Permission denied	get
410	Gone	Device not found	get/getEwons
429	Too Many Requests	Too many concurrent web connections	All (stateful)
500	Internal Server Error	Internal server error	all
500	Internal Server Error	Unexpected exception	all
502	Bad Gateway	Error while getting proxy to remote device (behind the Ewon)	get (with proxy)
502	Bad Gateway	Error while proxying request	get
503	Service Unavailable	Error while sending device offline	sendOffline
503	Service Unavailable	Error while sending device offline	wakeUp

Box 4126
300 04 Halmstad, Sweden

info@hms.se