ETSI TS 118 109 V2.13.1 (2020-03)

oneM2M;
HTTP Protocol Binding
(oneM2M TS-0009 version 2.13.1 Release 2A)

Reference

DTS/oneM2M-000009v2A

Keywords

IoT, M2M, protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Partnership Project oneM2M (oneM2M).

1 Scope

The present document covers the protocol specific part of communication protocol used by oneM2M compliant systems as RESTful HTTP binding.

The scope of the present document is (not limited to as shown below):

- Binding oneM2M Protocol primitive types to HTTP method.
- Binding oneM2M response status codes (successful/unsuccessful) to HTTP response codes.
- Binding oneM2M RESTful resources to HTTP resources.

The present document is depending on Core Protocol specification (ETSI TS 118 104 [3]) for data types.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 7230 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[2] ETSI TS 118 103: "oneM2M; Security solutions (oneM2M TS-0003)".

[3] ETSI TS 118 104: "oneM2M; Service Layer Core Protocol Specification (oneM2M TS-0004)".

[4] IETF RFC 7235 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

[5] IETF RFC 6750 (October 2012): "The Oauth 2.0 Authorization Framework: Bearer Token Usage".

[6] ETSI TS 118 111: "oneM2M; Common Terminology (oneM2M TS-0011)".

[7] ETSI TS 118 101: "Functional Architecture (oneM2M TS-0001)".

[8] IETF RFC 7232 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): "Conditional Requests".

[9] IETF RFC 3986 (January 2005): "Uniform Resource Identifier (URI): Generic Syntax".

[10] IETF RFC 7231 (June 2014): " Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules.

NOTE: Available at http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf.

[i.2] Void.

[i.3] Void.

[i.4] IETF RFC 6455 (December 2011):"The WebSocket Protocol".

[i.5] Void.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 118 111 [6] apply and the following apply:

HTTP	Hyper Text Transfer Protocol
TLS	Transport Layer Security
URI	Uniform Resource Identifier

4 Conventions

The keywords "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 Overview on HTTP Binding

5.0 Overview

HTTP binding specifies the equivalence between oneM2M request and response primitives and HTTP request and response messages, respectively. This clause provides a brief overview on the mapping relationship between oneM2M and HTTP message parameters.

This clause describes how oneM2M request/response primitives can be mapped to HTTP request/response messages and vice versa.

5.1 Introduction

Figure 5.1-1 illustrates an example oneM2M system configuration and its correspondence to an HTTP-based information system if HTTP binding as defined in the present document is applied. The upper diagram in figure 5.1-1 shows with solid line arrows the flow of a request primitive originating from an AE which is registered to an MN-CSE (Registrar of AE). The request primitive is assumed to address a resource which is hosted by another MN-CSE (Host of Resource). Both MN-CSEs are registered to the same IN-CSE.

When applying HTTP binding, the oneM2M entities of the upper diagram take the roles outlined in the lower diagram of a corresponding HTTP information system as defined in IETF RFC 7230 [1]. The AE takes the role of an HTTP client, the MN-CSE (Registrar of AE) takes the role of a HTTP Proxy Server, and both the IN-CSE and MN-CSE (Host of Resource) take the role of a HTTP server for this particular request message.

CSEs may also issue unsolicited request messages, shown with dashed line arrows in figure 5.1-1, and receive associated response messages. Therefore, for HTTP protocol binding, CSEs generally provides capability of both HTTP Server and HTTP Client. AEs may provide HTTP Server capability optionally in order to be able to serve Notification request messages (see ETSI TS 118 104 [3] and ETSI TS 118 101 [7]).

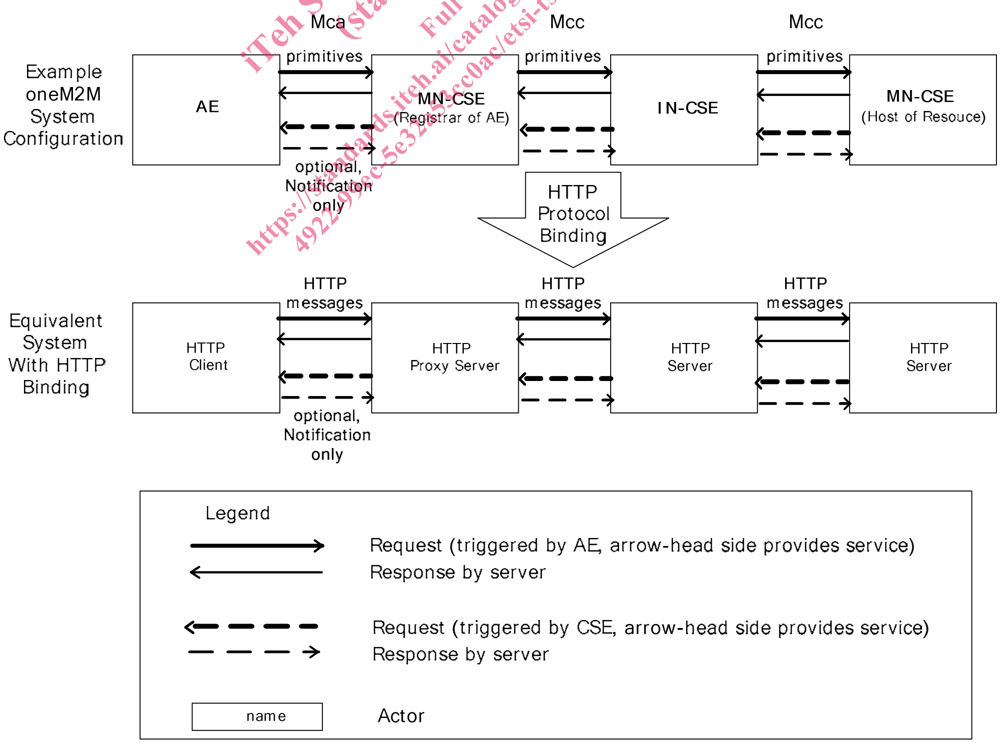

Figure 5.1-1: Correspondence between oneM2M entities and HTTP Client and Server

Each individual request primitive will be mapped to single HTTP request message, and each individual response primitive will be mapped to a single HTTP response message, and vice-versa.

An HTTP request message consists of Request-Line, headers and message-body. An HTTP response message consists of Status-Line, headers and message-body (IETF RFC 7230 [1]). HTTP header names are case-insensitive and a Receiver shall accept headers that are either lower or upper or any mixture thereof. This clause describes how oneM2M request/response primitives are mapped to HTTP messages at a high level. Corresponding details are specified in clause 6.

5.2 Request-Line

The HTTP method of a request message is mapped to the ***Operation*** parameter, and vice-versa.

At the message originator side the HTTP Request-Target is derived from the ***To*** parameter of the request primitive, including a query string which carries other specific primitive parameters.

HTTP-Version is specified in clause 6.

5.3 Status-Line

HTTP Version is specified in clause 6.

The Status-Code of HTTP response messages is derived from the ***Response Status Code*** parameter of the response primitive. The Reason-Phrase is not applicable to oneM2M systems and is omitted.

6 HTTP Message Mapping

6.1 Introduction

Mapping between oneM2M primitives and HTTP messages shall be applied in the following four use cases:

1) Mapping of request primitive to HTTP request message at the request originator (HTTP client).

2) Mapping of HTTP request message to request primitive at the request receiver (HTTP server).

3) Mapping of response primitive to HTTP response message at the request receiver (HTTP server).

4) Mapping of HTTP response message to response primitive at the request originator (HTTP client).

All four use cases also appear at transit CSEs.

The following clauses specify the mapping between each oneM2M primitive parameter and a corresponding HTTP message field to compose a HTTP request/response message.

6.2 Parameter Mappings on Request-Line

6.2.1 Method

The HTTP 'Method' shall be derived from the ***Operation*** request primitive parameter of the request primitive.

Table 6.2.1-1: HTTP Method Mapping

oneM2M Operation	HTTP Method
Create	POST
Retrieve	GET
Update	PUT
Delete	DELETE
Notify	POST

At the Receiver, an HTTP request message with POST method shall be mapped either to a Create or Notify ***Operation*** parameter. Discrimination between Create and Notify operations can be accomplished by inspection of the content-type header. The ***Resource Type*** parameter is present in the content-type header only when the HTTP POST request represents a Create request (see clause 6.4.3). The ***Resource Type*** parameter is not present in the content-type header when the HTTP POST request represents a Notify request.

6.2.2 Request-Target

6.2.2.1 Path component

The path component of the origin-form HTTP Request-Target shall be interpreted as the mapping of the resource identifier part of the ***To*** request primitive parameter. If the HTTP message is sent directly to the next hop CSE, the origin-form of Request-Target shall be employed (see clause 5.3.1 of IETF RFC 7230 [1]).

The resource identifier part of the ***To*** parameter can be represented in three different forms (see clause 6.2.3 of ETSI TS 118 104 [3] and clause 7.2 of ETSI TS 118 101 [7]):

- CSE-Relative-Resource-ID.
- SP-Relative-Resource-ID.
- Absolute-Resource-ID.

Each of the above three formats may include either a structured Resource ID (used for hierarchical addressing) or an unstructured Resource ID (used for non-hierarchical addressing) as defined in clause 7.2 of ETSI TS 118 101 [7].

For CSE-relative Resource ID representation, the path component of the HTTP request message shall be constructed as the concatenation of the literal "/" and the ***To*** request primitive parameter. For SP-relative Resource ID representation, the path component of the HTTP request message shall be constructed as the concatenation of the literal "/~" and the ***To*** request primitive parameter. For Absolute Resource ID representation, the path component of the HTTP request message shall be constructed by replacing the first "/" character of the ***To*** request primitive parameter with "/_".

Table 6.2.2.1-1 shows valid mappings between the ***To*** request primitive parameter and the path component of the origin-form HTTP request target. In the shown examples, /myCSEID and /CSE178 represent applicable CSI-IDs, CSEBase represents the resource name of a <CSEBase> resource, CSEBase/ae12/cont27/contInst696 represents a structured CSE-relative resource ID, and cin00856 an unstructured CSE-relative resource ID.

Table 6.2.2.1-1: Mapping examples between *To* parameter and path component of request-line

Resource-ID Type	*To* parameter value	path component (origin-form)
structured CSE-Relative	CSEBase/ae12/cont27/contInst696	/CSEBase/ae12/cont27/contInst696/
unstructured CSE-Relative	cin00856	/cin00856
structured SP-Relative	/CSE178/CSEBase/ae12/cont27/contInst696	/~/CSE178/CSEBase/ae12/cont27/contInst696
unstructured SP-Relative	/CSE178/cin00856	/~/CSE178/cin00856
structured Absolute	//mym2msp.org/CSE178/CSEBase/ ae12/cont27/contInst696	/_/mym2msp.org/CSE178/CSEBase/ ae12/cont27/contInst696
unstructured Absolute	//mym2msp.org/CSE178/cin00856	/_/mym2msp.org/CSE178/cin00856

At the HTTP server side, the reverse operations shall be applied to the path component of request-line to derive a replica of the original ***To*** request primitive parameter.

If the HTTP message is sent to a HTTP proxy instead directly to the next hop CSE, the absolute-form of Request-Target shall be employed (see clause 5.3.2 of IETF RFC 7230 [1]). The absolute-form is derived by prefixing the origin-form with the schema and the host address of the next hop CSE:

http://{host address of next hop CSE}{origin-form path-component}