International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.4500.9

(03/2018)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Internet of things and smart cities and communities – Frameworks, architectures and protocols

oneM2M – HTTP protocol binding

Recommendation ITU-T Y.4500.9

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	**Y.4400–Y.4549**
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.4500.9

oneM2M – HTTP protocol binding

Summary

Recommendation ITU-T Y.4500.9 specifies the protocol specific part of the communication protocol used by oneM2M compliant systems as RESTful HTTP binding.

The scope of the present Recommendation 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.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4500.9	2018-03-01	20	11.1002/1000/13504

Keywords

oneM2M, HTTP.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

NOTE – This Recommendation departs slightly from the usual editorial style of ITU-T Recommendations to preserve existing cross-referencing from external documents.

Table of Contents

Recommendation ITU-T Y.4500.9

oneM2M – HTTP protocol binding

1 Scope

This Recommendation specifies the protocol specific part of the communication protocol used by oneM2M compliant systems as RESTful HTTP binding.

The scope of this Recommendation 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 Recommendation contains oneM2M Release 2 specification – oneM2M HTTP Protocol Binding V2.6.1 and is equivalent to standards of oneM2M partners including Association of Radio Industries and Businesses (ARIB), Alliance for Telecommunications Industry Solutions (ATIS) [b-ATIS.oneM2M.TS009], China Communications Standards Association (CCSA) [b-CCSA M2M-TS-0009-V2.6.1], European Telecommunications Standards Institute (ETSI) [b-ETSI TS 118 109 V2.6.1], Telecommunications Industry Association (TIA), Telecommunications Standards Development Society India (TSDSI) [b-TSDSI STD TS-0009], Telecommunications Technology Association (TTA) [b-TTA MM-TS.0009] and Telecommunication Technology Committee (TTC) [b-TTC TS-M2M-0009].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.4500.1] Recommendation ITU-T Y.4500.1 (2018), *Functional architecture*.

[ITU-T Y.4500.4] Recommendation ITU-T Y.4500.4 (2018), *Service layer core protocol*.

[ETSI TS 118 103] ETSI TS 118 103 (2016), *oneM2M; Security solutions*.

[IETF RFC 3986] IETF RFC 3986 (2005), *Uniform Resource Identifier (URI): Generic Syntax*.

[IETF RFC 6750] IETF RFC 6750 (2012), *The OAuth 2.0 Authorization Framework: Bearer Token Usage*.

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

[IETF RFC 7232] IETF RFC 7232 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*.

[IETF RFC 7235] IETF RFC 7235 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Authentication*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application entity (AE) [b-ITU-T Y.4500.11]: Represents an instantiation of application logic for end-to-end M2M solutions.

3.1.2 common services entity (CSE) [b-ITU-T Y.4500.11]: Represents an instantiation of a set of common service functions of the M2M environments. Such service functions are exposed to other entities through reference points.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AE Application Entity

CSE Common Services Entity

HTTP Hyper Text Transfer Protocol

RESTful Representational state transfer

TLS Transport Layer Security

URI Uniform Resource Identifier

5 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described:

Shall/Shall not:

Requirements

1) effect on this Recommendation: This Recommendation needs to describe the required feature (i.e., specify a technical solution for the Requirement);
2) effect on products: every implementation (M2M Solution that complies to this Standard) must support it;
3) effect on deployments: every deployment (M2M Service based on this Standard) must use the Standardized feature where applicable – otherwise e.g., interoperability problems with other services could arise.

Should/Should not:

Recommendation

1) effect on this Recommendation: This Recommendation needs to describe a solution that allows the presence and the absence of the feature;
2) effect on products: an implementation may or may not support it, however support is recommended;
3) effect on deployments: a deployment may or may not use it, however usage is recommended.

May/Need not:

Permission/Option

1) effect on this Recommendation: This Recommendation needs to describe a solution that allows the presence and the absence of the required feature;
2) effect on products: an implementation may or may not support it;
3) effect on deployments: A deployment may or may not use it.

6 Overview on HTTP binding

6.0 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.

6.1 Introduction

Figure 6.1-1 illustrates an example oneM2M system configuration and its correspondence to an HTTP-based information system if HTTP binding as defined in this Recommendation is applied. The upper diagram in Figure 6.1-1 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]. 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 6.1-1, 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 TS-0004 [ITU-T Y.4500.4] and TS-0001 [ITU-T Y.4500.1]). See Appendix II for information on notification using WebSocket.

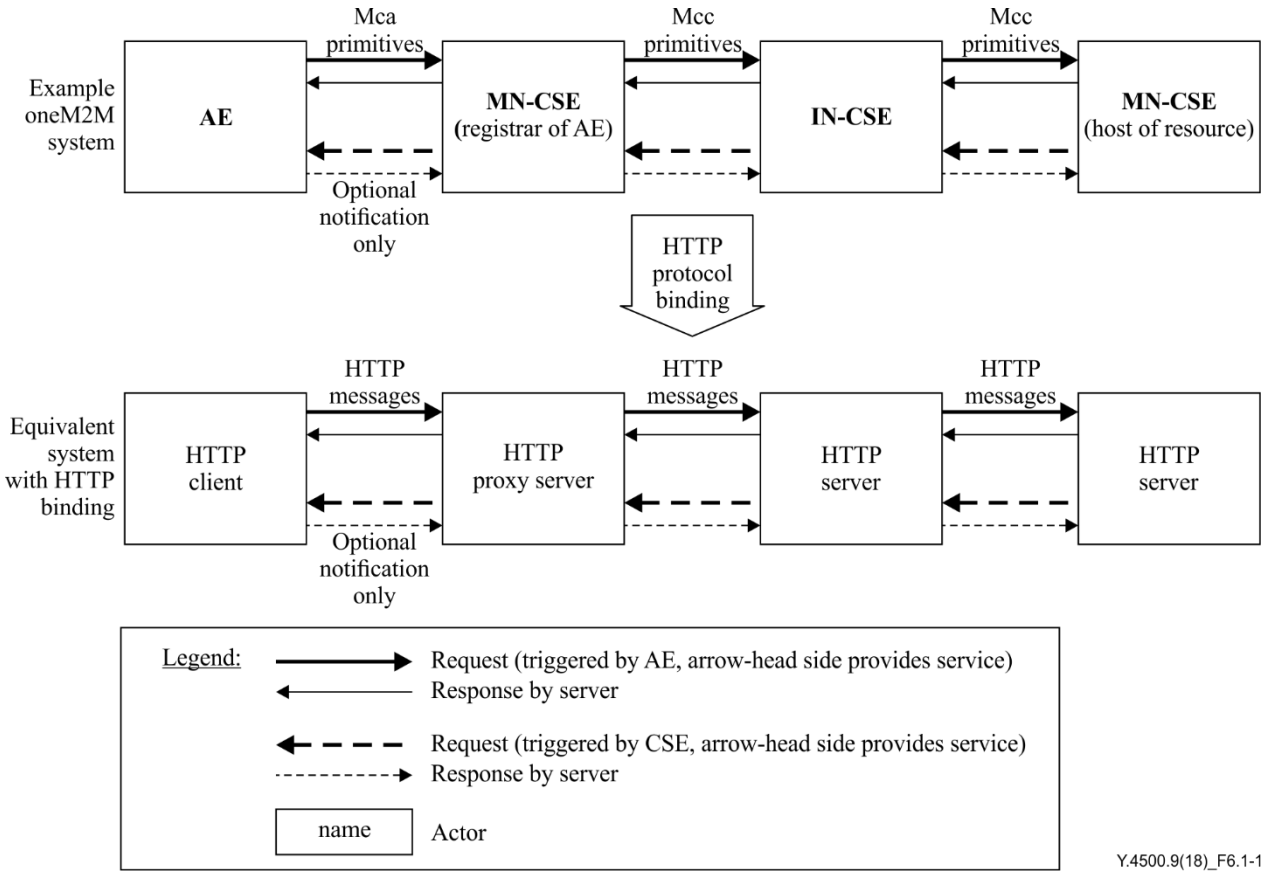

Figure 6.1-1 – Correspondence between oneM2M entities and HTTP client and server

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]. 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 7. See Appendix I for example procedures.

6.2 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 7.

6.3 Status-Line

HTTP Version is specified in clause 7.

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.

7 HTTP message mapping

7.1 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.

7.2 Parameter mappings on Request-Line

7.2.1 Method

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

Table 7.2.1-1 – 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 7.4.3). The ***Resource Type*** parameter is not present in the content-type header when the HTTP POST request represents a Notify request.

7.2.2 Request-Target

7.2.2.1 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]).

The resource identifier part of the ***To*** parameter can be represented in three different forms (see clause 6.2.3 of oneM2M TS-0004 [ITU-T Y.4500.4] and clause 7.2 of oneM2M TS-0001 [ITU-T Y.4500.1]):

- 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 oneM2M TS-0001 [ITU-T Y.4500.1].

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 7.2.2.1-1 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 7.2.2.1-1 – 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]). 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}

7.2.2.2 Query component

The query component (e.g., query-string) may include the optional primitive parameters listed in Table 7.2.2.2-1 compliant with [IETF RFC 7230]. Each applicable request primitive parameters and elements of ***Filter Criteria*** parameter shown in Table 7.2.2.2-1 shall be represented as pair of field-name and value in query-string. Multiple such pairs shall be concatenated with an ampersand '&' character used as separator between two pairs.

Table 7.2.2.2-1 also shows the permitted multiplicity of occurrence of field names in the query-string. Multiplicity '0..1' means that a parameter is optional and can occur at most once. Parameters with multiplicity '0..n', may occur multiple times in the query-string in the form of <query field name> =

value. For example, if the resourceType element of the ***Filter Criteria*** parameter is represented by a list of 3 values '2 3 4' (see clause 6.3.4.7 in TS-0004 [ITU-T Y.4500.4]), it would be mapped to ty=2+3+4 in the query-string. At the receiver side, this query string can be reverted back into the list type of representation. The same representation shall be applied for multiple occurrences of contentType and labels elements.

The 'attribute' element of the ***Filter Criteria*** request primitive parameter consists of two elements, name and value, which in XML notation would look for example as follows in case of multiplicity 2 (see clause 6.2.4.8 in TS-0004 [ITU-T Y.4500.4]):

```
<attribute>
  <name>attname1</name>
  <value>attvalue1</value>
</attribute>
<attribute>
  <name>attname2</name>
  <value>attvalue2</value>
</attribute>
```

Each name (e.g., attname1 and attname2) shall represent a valid resource attribute name of the resource types indicated in the ty field of the query-string. The sequence of attribute elements as shown in the above example will be mapped into the query-string as attname1=attvalue1&attname2=attvalue2. The attribute names (i.e., attname1 and attname2 in the above example) shall be expressed in the form of short names as defined in clause 8.2.3 of TS-0004 [ITU-T Y.4500.4]. Note that the <attribute> tag of the XML representation is omitted in the HTTP binding.

Examples of valid Request-Target representations are the following:

EXAMPLE 1): Request-Target for 'nonBlockingRequestSynch'

Primitive parameters: To: /CSE1234/RCSE78/container234 (SP-Relative-Resource-ID)

Response Type: responseType = 1 (nonBlockingRequestSynch)

Result Persistence: P1Y2M3DT10H1M0S

Request-Target: /CSE1234/RCSE78/container234?rt=1&rp=P1Y2M3DT10H1M0S

EXAMPLE 2): Request-Target for Discovery

When the entity wants to discover container resources where the *creator* attribute has the value 'Sam':

Primitive parameters: To: /CSE1234/RCSE78

Filter Criteria: resourceType = 3 (container)

attribute name: creator

attribute value: Sam

filterUsage = discovery

Request-Target: /CSE1234/RCSE78?ty=3&cr=Sam&fu=1

EXAMPLE 3): Semantic Discovery

The entity wants to discover resources whose semantic description stored in the *descriptor* attribute of a <semanticDescriptor> child resource fulfils the semantic filter specified in SPARQL. In this case,

the semantic descriptor of the resource to discover has to contain information about a Thing of type Car based on the concept defined in the "myOnt" ontology.

Due to the use of reserved characters in SPARQL, the semanticsFilter requires "percent-encoding" [IETF RFC 3986].

Primitive parameters: To: /CSE1234/RCSE78

Filter criteria: semanticsFilter =
PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX myOnt: http://www.onem2m.org/ontology/myontology#
SELECT ?car WHERE { ?car rdf:type myOnt:Car }

Request-Target:
/CSE1234/RCSE78?smf=PREFIX%20rdf%3A%20%3Chttp%3A%2F%2Fwww.w3.org%2F1999%2F02%2F22-rdf-syntax-ns%23%3E%20PREFIX%20myOnt%3A%20%3Chttp%3A%2F%2Fwww.onem2m.org%2Fontology%2Fmyontology%23%3E%20SELECT%20%3Fcar%20WHERE%20%7B%20%3Fcar%20%20rdf%3Atype%20myOnt%3ACar%20%7D

Any of the short names listed in Table 7.2.2.2-1, with the exception of 'atr', may be used in the query-string. The short name 'atr' itself is not used. Instead, any of the resource attribute short names as listed in Tables 8.2.3-1 to 8.2.3-5 in oneM2M TS-0004 [ITU-T Y.4500.4] may be used in the query-string in representations of attname=attvalue expressions, except those that shall be omitted (see clause 7.3.3.17.9 in oneM2M TS-0004 [ITU-T Y.4500.4]).

Table 7.2.2.2-1 – oneM2M request parameters mapped as query-string field

Request primitive parameter	Query field name	Multiplicity	Note
Response Type	***rt***	0..1	*responseType* element of data type ***responseTypeInfo*** (cf. clause 6.3.4.29 of TS-0004 [ITU-T Y.4500.4])
Result Persistence	***rp***	0..1	
Result Content	***rcn***	0..1	
Delivery Aggregation	***da***	0..1	
createdBefore	***crb***	0..1	filterCriteria condition
createdAfter	***cra***	0..1	filterCriteria condition
modifiedSince	***ms***	0..1	filterCriteria condition
unmodifiedSince	***us***	0..1	filterCriteria condition
stateTagSmaller	***sts***	0..1	filterCriteria condition
stateTagBigger	***stb***	0..1	filterCriteria condition
expireBefore	***exb***	0..1	filterCriteria condition
expireAfter	***exa***	0..1	filterCriteria condition
labels	***lbl***	0..n	filterCriteria condition
resourceType	***ty***	0..n	filterCriteria condition
sizeAbove	***sza***	0..1	filterCriteria condition
sizeBelow	***szb***	0..1	filterCriteria condition
contentType	***cty***	0..n	filterCriteria condition

Table 7.2.2.2-1 – oneM2M request parameters mapped as query-string field

Request primitive parameter	Query field name	Multiplicity	Note
limit	***lim***	0..1	filterCriteria condition
attribute	***atr***	0..n	filterCriteria condition
filterUsage	***fu***	0..1	filterCriteria condition
semanticsFilter	***smf***	0..n	filterCriteria condition, shall use "percent-encoding" [IETF RFC 3986] where required, see example 3)
filterOperation	***fo***	0..1	filterCriteria condition
contentFilterSyntax	***cfs***	0..1	filterCriteria condition
contentFilterQuery	***cfq***	0..1	filterCriteria condition
level	***lvl***	0..1	filterCriteria condition
offset	***ofst***	0..1	filterCriteria condition
Discovery Result Type	***drt***	0..1	
Role IDs	***rids***	0..n	
Token IDs	***tids***	0..n	
LocalTokenIDs	***ltids***	0..n	
Token Request Indicator	***tqi***	0..n	

For partial Retrieve request primitives, the ***To*** parameter may include the name of a single attribute separated by a '#' character from the resource ID. If multiple resource attributes are to be retrieved with a partial retrieve request primitive, these attributes are included in form of an attributeList object (as specified in clause 6.3.4.9 of TS-0004 [ITU-T Y.4500.4] with empty values) in the ***Content*** parameter.

In both cases, the short resource attribute name(s) shall be included into the fragment component of request-target, i.e., it shall follow any required query-string separated by '#' character. If more than a single attribute name is included into the fragment component, these shall be separated by a '+' character.

For example, if three resource attributes with long names resourceID, labels and requestReachability are indicated in the ***Content*** primitive parameter, the query component atrl=ri+lbl+rr is attached to the request-target. In case just a single attribute "rr" is indicated in the ***To*** parameter separated by '#' character, the query component atrl=rr is attached to the request-target. The '#' character and following attribute name shall be omitted from the path component of the request line.

At the HTTP server side, the reverse operation shall take place, when constructing the retrieve request primitive from the receive HTTP request message. Single attribute names in the query component may either be mapped back into the ***To*** parameter following a '#' character, or included into the ***Content*** parameter using the attributeList format with just a single list element included. Multiple attributes shall be included into the ***Content*** parameter as specified in oneM2M TS-0004 [ITU-T Y.4500.4].

7.2.3 HTTP-Version

This specification defines binding compliant with HTTP 1.1 [IETF RFC 7230]. The HTTP version field in HTTP request messages shall be set to "HTTP/1.1".

7.3 Status-Line

7.3.1 HTTP-Version

The HTTP version field in HTTP response messages shall be set to "HTTP/1.1".

7.3.2 Status-Code

The ***Response Status Code*** parameter of response primitives shall be mapped to the HTTP Status-Code. Since the ***Response Status Code*** parameter values have been defined with more detailed information than HTTP status codes, one or more ***Response Status Code*** value may be mapped to the same HTTP Status-Code. The original ***Response Status Code*** parameter value shall be carried in the X-M2M-RSC header (see clause 7.4.17).

The mapping of ***Response Status Code*** parameter value of oneM2M request primitive to Status-Code of HTTP request messages is specified in Table 7.3.2-1.

Table 7.3.2-1 – Status code mapping

oneM2M response status codes	HTTP status codes
2000 (OK)	200 (OK)
2002 (DELETED)	
2004 (UPDATED)	
2001 (CREATED)	201 (Created)
1000 (ACCEPTED)	202 (Accepted)
4000 (BAD_REQUEST)	400 (Bad Request)
4102 (CONTENTS_UNACCEPTABLE)	
4110 (GROUP_MEMBER_TYPE_INCONSISTENT)	
6010 (MAX_NUMBER_OF_MEMBER_EXCEEDED)	
6022 (INVALID_CMDTYPE)	
6023 (INVALID_ARGUMENTS)	
6024 (INSUFFICIENT_ARGUMENTS)	
6028 (ALREADY_COMPLETE)	
6029 (MGMT_COMMAND_NOT_CANCELLABLE)	
4101 (SUBSCRIPTION_CREATOR_HAS_NO_PRIVILEGE)	403 (Forbidden)
4103 (ORIGINATOR_HAS_NO_PRIVILEGE)	
5105 (RECEIVER_HAS_NO_PRIVILEGE)	
5106 (ALREADY_EXISTS)	
5203 (TARGET_NOT_SUBSCRIBABLE)	
5205 (SUBSCRIPTION_HOST_HAS_NO_PRIVILEGE)	
4106 (ORIGINATOR_HAS_NOT_REGISTERED)	
4107 (SECURITY_ASSOCIATION_REQUIRED)	
4108 (INVALID_CHILD_RESOURCE_TYPE)	
4109 (NO_MEMBERS)	
4111 (ESPRIM_UNSUPPORTED_OPTION)	
4112 (ESPRIM_UNKNOWN_KEY_ID)	
4113 (ESPRIM_UNKNOWN_ORIG_RAND_ID)	

Table 7.3.2-1 – Status code mapping

oneM2M response status codes	HTTP status codes
4114 (ESPRIM_UNKNOWN_RECV_RAND_ID)	
4115 (ESPRIM_BAD_MAC)	
4116 (ESPRIM_IMPERSONATION_ERROR)	
5208 (DISCOVERY_DENIED_BY_IPE)	
4004 (NOT_FOUND)	404 (Not Found)
5103 (TARGET_NOT_REACHABLE)	
6003 (EXTERNAL_OBJECT_NOT_REACHABLE)	
6005 (EXTERNAL_OBJECT_NOT_FOUND)	
4005 (OPERATION_NOT_ALLOWED)	405 (Method Not Allowed)
5207 (NOT_ACCEPTABLE)	406 (Not Acceptable)
4008 (REQUEST_TIMEOUT)	408 (Request Timeout)
4104 (GROUP_REQUEST_IDENTIFIER_EXISTS)	409 (Conflict)
4105 (CONFLICT)	
5000 (INTERNAL_SERVER_ERROR)	500 (Internal Server Error)
5204 (SUBSCRIPTION_VERIFICATION_INITIATION_FAILED)	
5209 (GROUP_MEMBERS_NOT_RESPONDED)	
5210 (ESPRIM_DECRYPTION_ERROR)	
5211 (ESPRIM_ENCRYPTION_ERROR)	
5212 (SPARQL_UPDATE_ERROR)	
6020 (MGMT_SESSION_CANNOT_BE_ESTABLISHED)	
6021 (MGMT_SESSION_ESTABLISHMENT_TIMEOUT)	
6025 (MGMT_CONVERSION_ERROR)	
6026 (MGMT_CANCELLATION_FAILED)	
5001 (NOT_IMPLEMENTED)	501 (Not Implemented)
5206 (NON_BLOCKING_REQUEST_NOT_SUPPORTED)	

7.3.3 Reason-Phrase

The Reason-Phrase shall be omitted in HTTP response messages.

7.4 Header fields

7.4.0 Introduction

The header fields listed in this clause shall be supported by all entities of the oneM2M system when using HTTP binding. Any other unrecognized HTTP headers shall be ignored by the HTTP client and server.

7.4.1 Host

The Host header shall be present in each HTTP request message.

While the Request-Target indicates a target resource on the Hosting CSE, the Host header indicates the FQDN or IP address of the Receiver CSE of the next hop in multi-hop communication scenarios. Therefore, the Request-Target is not changed but the Host header is changed each time when a request is forwarded to the next hop CSE.

When no HTTP proxy is used, the Host header shall be set as one of the pointOfAccess attribute values of the Receiver (i.e., pointOfAccess attribute of the corresponding <remoteCSE> resource). Selection of the appropriate Receiver is described in oneM2M TS-0004 [ITU-T Y.4500.4]. In this case the origin-form of target URI shall be used (see clause 7.2.2).

If the HTTP request message is sent to a HTTP proxy rather than to the next hop CSE, the Host header shall be set to the FQDN or IP address of the proxy. In this case the absolute-form of target URI shall be used (see clause 7.2.2).

7.4.2 Accept

The Originator may use the Accept header to indicate which media types are acceptable for the response. The Accept header shall be mapped to a set of media types among "application/xml", "application/json", "application/cbor" or the oneM2M defined media types defined in clause 6.7 of oneM2M TS-0004 [ITU-T Y.4500.4]. Note that some of the oneM2M defined media types defined in clause 6.7 of oneM2M TS-0004 [ITU-T Y.4500.4] are not applicable for the response. Note that this information is not included in a request primitive.

7.4.3 Content-Type

Any HTTP request or response containing message-body shall include the Content-type header set to one of "application/xml", "application/json", or the oneM2M defined media types defined in clause 6.7 of oneM2M TS-0004 [ITU-T Y.4500.4].

Content-Type of the HTTP response should be chosen by the Hosting CSE considering the Accept header given in the HTTP request.

The value of the Resource Type primitive parameter, which is present in Create request primitives only, shall be appended to the Content-type of the corresponding HTTP request message in the form ty=value, separated by a semicolon character. A valid Content-Type header in this case looks e.g., as follows:

Content-Type: application/vnd.onem2m-res+xml; ty=3
application/vnd.onem2m-res+json; ty=3
application/vnd.onem2m-res+cbor; ty=3

7.4.4 Content-Location

The Content-Location header of HTTP response messages shall be set to the URI of the created resource, when responding to a Create request primitive. The URI shall be retrieved from the ***Content*** parameter of the response primitive. See clause 7.3.3.12 "Create a success response" in oneM2M TS-0004 [ITU-T Y.4500.4].

7.4.5 Content-Length

If message-body is included into HTTP request or response messages, the Content-Length header shall be included indicating the length of the message-body in octets (8-bit bytes).

7.4.6 Etag

A response primitive sent in reply to a resource retrieval request primitive should include an Etag header [IETF RFC 7232] in combination with the resource representation in the HTTP message body.

Etag facilitates the use of conditional requests (i.e., using the if-match and if-none-match HTTP headers) [IETF RFC 7232].

If a CSE supports the Etag header, then the CSE shall support conditional requests compliant with [IETF RFC 7232].

7.4.7 X-M2M-Origin

The X-M2M-Origin header shall be mapped to the ***From*** parameter of request and response primitives and vice versa, if applicable.

7.4.8 X-M2M-RI

The X-M2M-RI header shall be mapped to the ***Request Identifier*** parameter of request and response primitives and vice versa.

7.4.9 Void

This clause is intentionally left blank.

7.4.10 X-M2M-GID

The X-M2M-GID header shall be mapped to the ***Group Request Identifier*** parameter of request primitives and vice versa, if applicable.

7.4.11 X-M2M-RTU

The X-M2M-RTU header shall be mapped to the *notificationURI* element of the ***Response Type*** parameter of request primitives and vice versa, if applicable. If there are more than one value in the element, then the values shall be combined with "&" character.

7.4.12 X-M2M-OT

The X-M2M-OT header shall be mapped to the ***Originating Timestamp*** parameter of request and response primitives, and vice versa, if applicable.

7.4.13 X-M2M-RST

The X-M2M-RST header shall be mapped to the ***Result Expiration Timestamp*** parameter of request and response primitives, and vice versa, if applicable.

7.4.14 X-M2M-RET

The X-M2M-RET header shall be mapped to the ***Request Expiration Timestamp*** parameter of request primitives and vice versa, if applicable.

7.4.15 X-M2M-OET

The X-M2M-OET header shall be mapped to the ***Operation Execution Time*** parameter of request primitives and vice versa, if applicable.

7.4.16 X-M2M-EC

The X-M2M-EC header shall be mapped to the ***Event Category*** parameter of request and response primitives, and vice versa, if applicable.

7.4.17 X-M2M-RSC

The X-M2M-RSC header in a HTTP response message shall be mapped to the ***Response Status Code*** parameter of response primitives and vice versa only if the mapping between the ***Response Status Code*** and the HTTP Status Code is N:1 relationship (e.g., ***Response Status Code*** 4000 and 4102 are mapped to HTTP Status Code 400 in the Table 7.3.2-1).

7.4.18 X-M2M-ATI

The X-M2M-ATI header in a HTTP response message shall be mapped to the ***Assigned Token Identifiers*** parameter of response primitives and vice versa.

The format of the X-M2M-ATI header shall be represented as a sequence of lti-value:tkid-value pairs separated by a colon ":' and multiple pairs appended with '+' character.

EXAMPLE: The header looks as follows:

X-M2M-ATI: lti-value1:tkid-value1 + lti-value2:tkid-value2 + …

if the XML representation of the ***Assigned Token Identifiers*** parameter is given as (using short element names):

```
<ati>
    <ltia>
        <lti>lti-value1</lti>
        <tkid>tkid-value1</tkid>
    </ltia>
    <ltia>
       <lti>lti-value2</lti>
       <tkid>tkid-value2</tkid>
    </ltia>
    …
 </ati>
```

The data type m2m:dynAuthlocalTokenIdAssignments of the ***Assigned Token Identifiers*** parameter is defined in clause 6.3.5.43 of TS-0004 [ITU-T Y.4500.4].

7.4.19 Authorization

If a request primitive includes a ***Tokens*** parameter it shall be mapped to the Authorization header.

The ***Tokens*** primitive parameter is represented as a space separated list of JSON Web Signature (JWS) and JSON Web Encryption (JWE) strings in Compact Serialization format of datatype m2m:dynAuthJWT as defined in clause 6.3.3 of TS-0004 [ITU-T Y.4500.4].

When mapped into the Authorization header, each individual token in the ***Tokens*** primitive parameter shall be separated by '+' character.

For example, if the ***Tokens*** parameter consists of a list of two JWS/JWE Tokens,

eyJ0eXAiOiJK.eyJpc3MiOiJqb2UiLA0KIC.dBjftJeZ4CVP
eyJ0eXAiOiJK.eyJpc3MiOiJqb2UiLA0KIC.dBjftJeZ4CVP.5eym8TW_c8SuK.SdiwkIr3a.XFBoMYUZo

the Authorization header looks as follows:

Authorization: eyJ0eXAiOiJK.eyJpc3MiOiJqb2UiLA0KIC.dBjftJeZ4CVP+
eyJ0eXAiOiJK.eyJpc3MiOiJqb2UiL

A0KIC.dBjftJeZ4CVP.5eym8TW_c8SuK.SdiwkIr3a.XFBoMYUZo

The line break in the above example is for illustrative purposes and shall not be included into the Authorization header.

7.4.20 X-M2M-CTS

The X-M2M-CTS header shall be mapped to the ***Content Status*** parameter of response primitives and vice versa, if applicable.

7.4.21 X-M2M-CTO

The X-M2M-CTO header shall be mapped to the ***Content Offset*** parameter of response primitives, and vice versa, if applicable.

7.5 Message-body

Message-body shall be mapped to the ***Content*** parameter of request and response primitives, and vice versa, if applicable. This applies to the ***Content*** parameter of all primitives with the following exceptions:

1. For partial Retrieve request primitives. Attributes contained in the Content parameter of Retrieve request primitive shall be mapped to the fragment component of request-target, as specified in clause 7.2.2.2, and vice versa.
2. A ***Token Request Information*** parameter included in a response primitive shall be mapped into the message-body either as a XML or JSON serialized object. The Content-Type and Content-Length headers shall be set compliant with the data representation (i.e., Content-Type: application/xml or application/json depending on the serialization format). Note that the ***Token Request Information*** parameter is used in oneM2M error response primitives (X-M2M-RSC: 4103 "ORIGINATOR_HAS_NO_PRIVILEGE") only, which do not carry any other primitive content.

   Error response messages which include the ***Token Request Information*** parameter in the Message-Body shall not include any debugging information.

7.6 Message routing

HTTP request and response message routing shall be performed as described in HTTP/1.1 [IETF RFC 7230].

8 Security consideration

8.1 Authentication on HTTP Request message

When sending the credential to be checked by the Registrar CSE, Proxy-Authorization header should be used as specified in HTTP/1.1 (see [IETF RFC 7235]).

When sending the credential to be checked by Hosting CSE, Authorization header should be used as specified in HTTP/1.1 [IETF RFC 7235].

When the credential to be checked by Hosting CSE is an Access Token which is compatible with OAuth 2.0 framework (see [IETF RFC 6750]), the Bearer authentication scheme shall be used as specified in OAuth 2.0 framework.

NOTE – The oneM2M Security solutions [ETSI TS 118 103] does not provide any details on usage or provisioning of the token.

8.2 Transport layer security

oneM2M primitive parameters contained in HTTP messages may be protected by TLS in a hop-by-hop manner. For the details, see the oneM2M Security solutions specification [ETSI TS 118 103].

NOTE – Some provisioning schemes of oneM2M TS-0003 [ETSI TS 118 103] enable the provisioning of end-to-end credentials, but protocols to establish security associations between non-adjacent nodes are not addressed by oneM2M in the present document.

Annex A

oneM2M specification update and maintenance control procedure

(This annex forms an integral part of this Recommendation.)

The provisions of Annex L in [ITU-T Y.4500.1] regarding the oneM2M Specification update and maintenance control procedure shall apply to this Recommendation.

Appendix I

Example procedures

(This appendix does not form an integral part of this Recommendation.)

I.1 <container> resource creation

Figure I.1-1 is an HTTP mapping of the procedure described in clause 7.4.7.2.1 of [ITU-T Y.4500.4]. Note the example shown in the figure applies under the following assumptions:

- "CSE1" is the name (i.e., value of the resourceName attribute) of the <CSEBase> resource of the registrar CSE
- "cont1" is the name of the created <container> resource chosen by the registrar CSE.

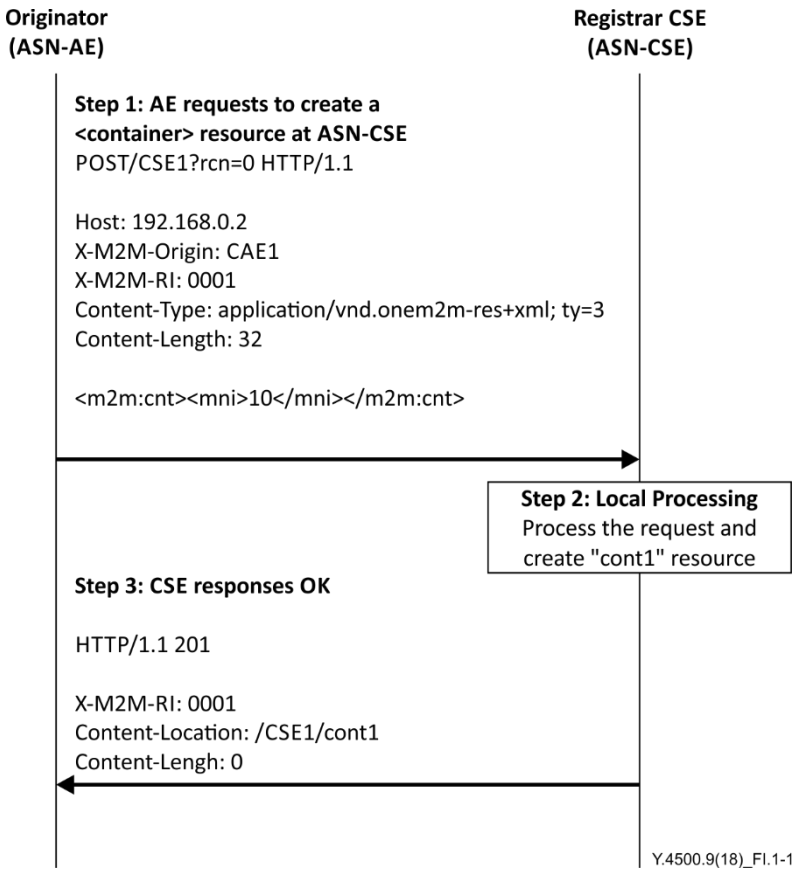

Figure I.1-1 – oneM2M HTTP binding example - Container creation

Appendix II

WebSocket

(This appendix does not form an integral part of this Recommendation.)

II.1 Notification using WebSocket

WebSocket [b-IETF RFC 6455] can be used for transporting notifications to an AE/CSE. This can be useful for an AE/CSE which is not server-capable or cannot be reachable for delivery of unsolicited requests.

For example, when an AE needs to receive a notification message from the CSE, the AE establishes a WebSocket connection to a CSE. When a new notification message is generated, the notification will be sent to the AE as the data frame of the WebSocket.

Bibliography

[b-ITU-T Y.4500.11] Recommendation ITU-T Y.4500.11 (2018), *Common terminology*.

[b-ATIS.oneM2M TS009] Alliance for Telecommunications Industry Solutions, ATIS.oneM2M.TS0009V101-2015, *HTTP Protocol Binding*.

[b-ETSI TS 118 109] European Telecommunications Standards Institute, ETSI TS 118 109 V2.6.1 (2016), *oneM2M; HTTP Protocol Binding*.

[b-IETF RFC 6455] IETF RFC 6455 (December 2011), *The WebSocket Protocol*.

[b-TSDSI STD TS-0009] Telecommunications Standards Development Society India, TSDSI STD T1.oneM2M TS-0009-2.6.1 V1.0.0, *HTTP Protocol Binding*.

[b-TTA MM-TS.0009] Telecommunications Technology Association, TTAT.MM-TS.0009 v2.6.1 (2017), *oneM2M - oneM2M - HTTP Protocol Binding*.

[b-TTC TS-M2M-0009] Telecommunication Technology Committee, TTC TS-M2M-0009 v2.6.1 (2016), *HTTP Protocol Binding*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2018