



## ONEM2M TECHNICAL SPECIFICATION

Document Number	TS-0009-HTTP_Protocol_Binding-V-2014-08
Document Name:	HTTP Protocol Binding Technical Specification
Date:	2014-08-01
Abstract:	HTTP Protocol binding TS

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

18 About oneM2M

19 The purpose and goal of oneM2M is to develop technical specifications which address the  
20 need for a common M2M Service Layer that can be readily embedded within various  
21 hardware and software, and relied upon to connect the myriad of devices in the field with  
22 M2M application servers worldwide.

23 More information about oneM2M may be found at: <http://www.oneM2M.org>

24 Copyright Notification

25 No part of this document may be reproduced, in an electronic retrieval system or otherwise,  
26 except as authorized by written permission.

27 The copyright and the foregoing restriction extend to reproduction in all media.

28 © 2013, oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TTA, TTC).

29 All rights reserved.

30 Notice of Disclaimer & Limitation of Liability

31 The information provided in this document is directed solely to professionals who have the  
32 appropriate degree of experience to understand and interpret its contents in accordance with  
33 generally accepted engineering or other professional standards and applicable regulations.  
34 No recommendation as to products or vendors is made or should be implied.

35 NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS  
36 TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE,  
37 GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO  
38 REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR  
39 FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF  
40 INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE  
41 LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY  
42 THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN  
43 NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER  
44 INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES  
45 ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN  
46 THIS DOCUMENT IS AT THE RISK OF THE USER.

47

# Contents

49	Contents .....	3
50	1 Scope .....	4
51	2 References .....	4
52	2.1 Normative references .....	4
53	2.2 Informative references .....	4
54	3 Definitions, symbols, abbreviations and acronyms .....	5
55	3.1 Definitions .....	5
56	3.2 Symbols .....	5
57	3.3 Abbreviations & Acronyms .....	5
58	4 Conventions.....	5
59	5 Overview of HTTP Binding .....	5
60	5.1 Introduction.....	5
61	5.2 Request-Line.....	5
62	5.3 Status-Line.....	5
63	5.4 Header Fields .....	5
64	5.5 Message-body .....	6
65	6 HTTP Message Mapping .....	6
66	6.1 Introduction.....	6
67	6.2 Request-Line.....	6
68	6.3 Status-Line.....	7
69	6.4 Header Fields .....	8
70	6.5 Message-body .....	10
71	7 Security Consideration .....	10
72	<i>Proforma copyright release text block</i> .....	13
73	<b>Annex &lt;y&gt;: Bibliography.....</b>	<b>13</b>
74	History .....	13
75		
76		

---

# 77 1 Scope

78 The specification will cover the protocol specific part of communication protocol used by oneM2M compliant systems  
79 as RESTful HTTP binding.

80 The scope of this specification is (not limited to as shown below):

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

84 This specification is depending on Core Protocol specification (TS-0004) for data types.

---

# 85 2 References

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

## 89 2.1 Normative references

- 90 [1] RFC2616: “Hypertext Transfer Protocol - HTTP/1.1”, IETF, June 1999
- 91 [2] oneM2M TS-0001: Architecture TS
- 92 [3] oneM2M TS-0004: Core Protocol TS

## 94 2.2 Informative references

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

- 97 [i.1] oneM2M Drafting Rules  
98 ([http://member.onem2m.org/Static\\_pages/Others/Rules\\_Pages/oneM2M-Drafting-Rules-](http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc)  
99 [V1\\_0.doc](http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc))
- 100 [i.2] RFC2617: “HTTP Authentication: Basic and Digest Access Authentication”, IETF, June 1999
- 101 [i.3] RFC6750: “The OAuth 2.0 Authorization Framework: Bearer Token Usage”, IETF, October 2012.
- 102 [i.4] RFC6455: “The WebSocket Protocol”, IETF, December 2011.

103

---

## 104 3 Definitions, symbols, abbreviations and acronyms

### 105 3.1 Definitions

### 106 3.2 Symbols

### 107 3.3 Abbreviations & Acronyms

108 For the purposes of the present document, the following abbreviations and acronyms apply:

109 HTTP Hyper Text Transfer Protocol

---

## 110 4 Conventions

111 The key words “Shall”, ”Shall not”, “May”, ”Need not”, “Should”, ”Should not” in this document are to be interpreted  
112 as described in the oneM2M Drafting Rules [i.1]

---

## 113 5 Overview of HTTP Binding

114 This clause describes what oneM2M primitive parameters can be mapped to HTTP request/response messages.

### 115 5.1 Introduction

116 An HTTP request message consists of Request-Line, headers and message-body. An HTTP response message  
117 consists of Status-Line, headers and message-body [4]. This clause describes how oneM2M request/response primitives  
118 are mapped to HTTP messages at a high level. Corresponding details of each sub-clause are specified in clause 6.

### 119 5.2 Request-Line

120 Method is mapped to the oneM2M *operation* parameter.

121 Request-URI is derived from the oneM2M *to* parameter, including a query string which carries specific primitive  
122 parameters.

123 HTTP-Version is specified in clause 6.

### 124 5.3 Status-Line

125 HTTP Version is specified in clause 6.

126 Status-Code and Reason-Phrase are derived from the oneM2M *responseStatusCode* parameter of the response  
127 primitive.

### 128 5.4 Header Fields

129 Mapping for the following header fields is specified in clause 6:

- 130 ● Accept
- 131 ● Content-Type
- 132 ● Content-Location
- 133 ● From

- Host
- Location

Editor's Note: Additions to this list are FFS.

Extension header fields may also be specified.

## 5.5 Message-body

Primitive parameters which are not contained in Request-Line, Status-Line or header fields are contained in message-body.

# 6 HTTP Message Mapping

## 6.1 Introduction

Mapping between HTTP message and oneM2M primitive shall be applied in the following cases:

- when the Originator sends a request primitive,
- when the Receiver receives a request primitive,
- when the Receiver sends a response primitive,
- when the Originator receives a response primitive.

The following sub-clauses specify how to map each oneM2M primitive parameter to a corresponding HTTP message field to compose a HTTP request/response message..

## 6.2 Request-Line

### 6.2.1 Method

Mapping between HTTP method in an HTTP request message and oneM2M *operation* parameter in a request primitive (clause 7.2.1.1.1 [6]) shall be applied in the following cases:

- when the Originator sends a request primitive,
- when the Receiver receives a request primitive.

The oneM2M operations shall be mapped as follows.

**Table 6.2.1-1: HTTP Method Mapping**

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

Editor's Note: Update operation mapping is TBD until the attribute level manipulation scheme is resolved in TS-0001 and TS-0004.

161 At the Receiver, an HTTP request message with POST method shall be mapped to a oneM2M Create or Notify request  
162 primitive in accordance with the value of the *operation* parameter.

## 163 6.2.2 Request-URI

164 Request-URI shall be mapped to the *to* parameter of the request primitive. Note: This may not include host and port  
165 number. Host and port number are carried in the Host header (see clause 6.3.1).

166 *filterCriteria, nm, and ty* parameters of the request primitive shall be mapped to query in the Request-URI.

167 Editor's Note: parameters inclusion in query string or extended HTTP header is TBD.

## 168 6.2.3 HTTP-Version

169 This specification supports binding to HTTP 1.1, so the version field shall be set to "HTTP/1.1".  
170

## 171 6.3 Status-Line

### 172 6.3.1 HTTP-Version

173 This specification supports binding to HTTP 1.1, so the version field shall be set to "HTTP/1.1".

### 174 6.3.2 Status-Code

175 The oneM2M responseStatusCode shall be mapped to HTTP Status-Code. Since the responseStatusCode have been  
176 defined more specifically, one or more responseStatusCodes may be mapped to Status-Code. The original  
177 responseStatusCode parameter shall be carried in message-body (see clause 6.5).

178 N:1 status code mapping from the oneM2M request primitive to HTTP request message shall be:

179

**Table 6.3.2-1: Status Code Mapping**

oneM2M Response Status Codes	HTTP Status Codes
Success	200 (OK)
Accepted	202 (Accepted)
Location info not authorized	403 (Forbidden)
Unsupported resource	404 (Not Found)
Unsupported attribute	404 (Not Found)
Cannot forward, target not reachable	404 (Not Found)
Cannot forward, other reason TBD	404 (Not Found)
Create error - no privilege	403 (Forbidden)
Create error - already exists	403 (Forbidden)
Create error - missing mandatory parameter	400 (Bad Request)
Retrieve error - no privilege	403 (Forbidden)
Retrieve error - does not exist	404 (Not Found)
Update error - no privilege	403 (Forbidden)

Update error - does not exist	404 (Not Found)
Update error - unacceptable contents	415 (Unsupported Media Type)
Delete error - does not exist	404 (Not Found)
Delete error - no privilege	403 (Forbidden)
Create delivery - not able to take on responsibility	403 (Forbidden)
Create fanoutpoint - group request identifier exists	409 (Conflicts)
Retrieve fanoutpoint - group request identifier exists	409 (Conflicts)
Update fanoutpoint - group request identifier exists	409 (Conflicts)
Delete fanoutpoint - group request identifier exists	409 (Conflicts)
Create mgmtObj - memory shortage	403 (Forbidden)
External object not found	404 (Not Found)
Cancel execlInstance - not cancellable	403 (Forbidden)
Cancel execlInstance - already complete	403 (Forbidden)
Delete execlInstance - not cancellable	403 (Forbidden)
Delete execlInstance - already complete	403 (Forbidden)
Retrieve CSEBase - format error	400 (Bad Request)
CMDH rules -non compliant	403 (Forbidden)
Target is not subscribable	403 (Forbidden)
Cannot initiate subscription verification	403 (Forbidden)
Subscription verification failed	403 (Forbidden)

180 **Editor's Note: This table needs to be updated aligning with TS-0004.**

### 181 6.3.3 Reason-Phrase

182 Reason-Phrase shall be mapped to the description of the corresponding responseStatusCode of the response primitive  
183 (see clause 6.5.4 [3]).

## 185 6.4 Header Fields

### 186 6.4.1 Host

187 A Host header shall be included in an HTTP request message. Other headers may be included in an HTTP  
188 request/response message.

189 Host shall be derived from the *to* parameter of the request primitive. This shall consist of the host and optionally port  
190 number.

### 191 6.4.2 Accept

192 An Originator may use the Accept header to indicate which content-type is supported by the Originator. The Accept  
193 header shall be set to "application/onem2m-resource+xml" or "application/onem2m-resource+json".



194 Editor's Note: Supported types are TBD.

### 195 6.4.3 Content-type

196 Any request or response containing message-body shall include the Content-type header set to “application/onem2m-  
197 resource+xml” or “application/onem2m-resource+json”.

198 Editor's Note: Supported types are TBD.

### 199 6.4.4 Content-Location

200 The Content-Location header shall be set to the URI of:

- 201 • the created resource, when responding to a Create request primitive;
- 202 • the retrieved resource, when responding to a Retrieve request primitive if the retrieved resource location is  
203 different from the requested resource location;
- 204 • the updated resource, when responding to a Update request primitive

205 Editor's Note: Usage of this header needs more investigation.

206 Editor's Note: How to indicate the oneM2M URI scheme is TBD

### 207 6.4.5 Content-Length

208 If message-body is included, the Content-Length header shall be included indicating the length of the message-body in  
209 octets (8-bit bytes).

### 210 6.4.6 ETag

211 A retrieve response primitive corresponding to a resource retrieval request primitive should include an ETag header  
212 together with the resource representation [1].

213 ETag facilitates the use of conditional requests (i.e. using the if-match and if-none-match HTTP headers).

214 If a CSE supports the ETag header, then the CSE shall support conditional requests.

### 215 6.4.7 From

216 The From header shall be mapped to the *from* parameter of the request/response primitive, and shall contain the  
217 oneM2M specified ID of the Originator (e.g. CSE-ID or AE-ID).

### 218 6.4.8 Location

219 The Location header shall be mapped to the URI of the created resource. This header shall be present in the response  
220 primitive corresponding to a create request and shall not be present in any other request or response.

221 Editor's Note: if Originator's nm parameter can be modified and accepted by the Hosting CSE is TBD.

### 222 6.4.9 X-M2M-RI

223 All requests and responses shall include an oneM2M defined header field called X-M2M-RI that contains the requestID.

224 Editor's Note: mapping request ID can be query or this extension header, this is TBD.

### 225 6.4.10 Other Header Fields

226 Other HTTP header fields shall be mapped with oneM2M primitive parameters.

227 Editor's Note: The list of the fields and details are FFS.

228  
229  
230  
231  
232  
233  
234  
235  
236

## 6.5 Message-body

Depends on the operation type and the reference point, the combination of the primitive parameters may be different (see clause 7.2.1.1 [3]).

In HTTP request message, among the request parameters (see Table 7.2.1.1-1 [3]) Message-body shall include mandatory parameters except *primitive type*, *to*, *from*, *request identifier* parameter and may include conditional/optional parameters.

In HTTP response message, among the response parameters (see Table 7.2.1.1-2 and 7.2.1.1-3 [3]) Message-body shall include mandatory parameters except *primitive type* parameter and may include conditional/optional parameters.

237

---

## 7 Security Consideration

238

### 7.1 Authentication on HTTP Request Message

239  
240

When sending the credential to be checked by Registrar CSE, Proxy-Authorization header should be used as specified in HTTP/1.1 (see RFC2617).

241  
242

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

243  
244

When the credential to be checked by Hosting CSE is an Access Token which is compatible with OAuth 2.0 framework, the Bearer authentication scheme shall be used as specified in RFC6750.

245

Note: TS-0003 [oneM2M Security Solutions] does not provide any details on usage or provisioning the token.

246

### 7.2 Transport Layer Security

247  
248

oneM2M primitive parameters contained in HTTP messages may be protected by TLS as hop-by-hop manner, not end-to-end. For the details, see clause 6.1 [TS-0003]

249

250

251

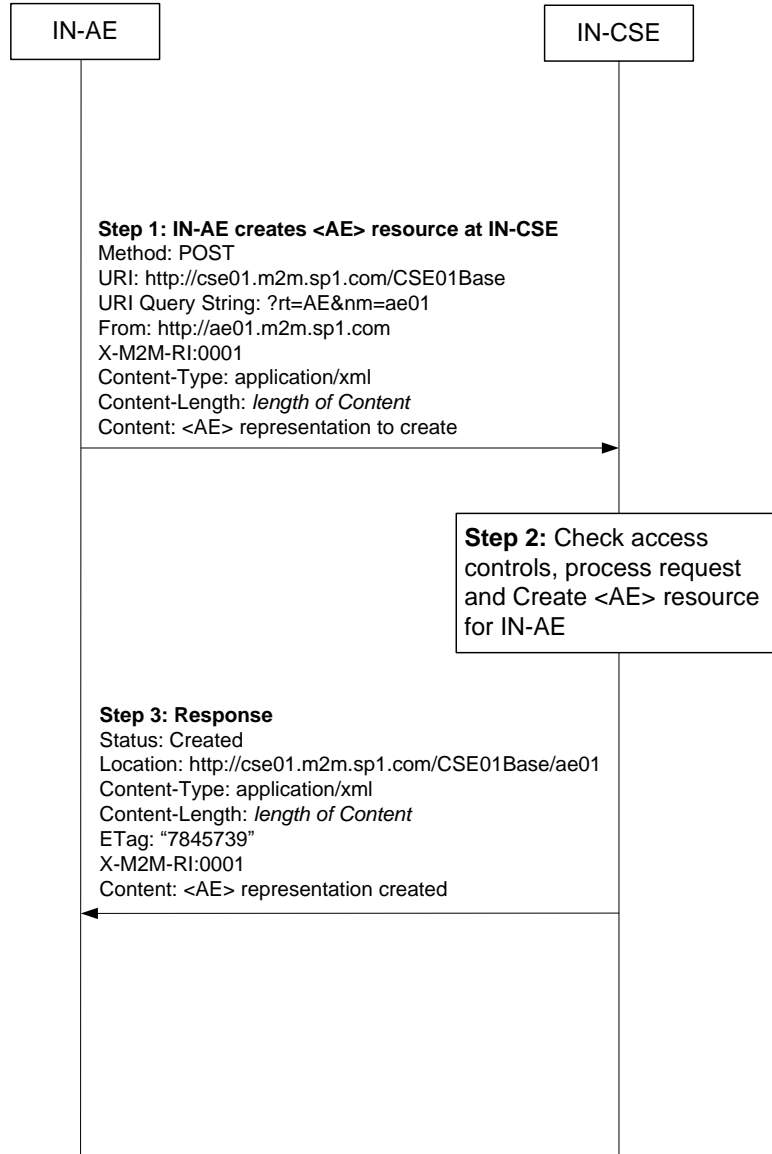
# Annex A (Informative): Example Procedures

252

## A.1 AE Registration

253

The following diagram is HTTP mapping of procedure described in clause 7.3.4.2.1.



254

255

Figure A.1-1 oneM2M HTTP Binding Example – AE Registration

256

257

---

## Annex B (Informative) WebSocket

258

### B.1 Notification using WebSocket

259

WebSocket [3] 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.

260

261

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.

262

263

264

265

266

267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290

---

## Proforma copyright release text block

*This text box shall immediately follow after the heading of an element (i.e. clause or annex) containing a proforma or template which is intended to be copied by the user. Such an element shall always start on a new page.*

Notwithstanding the provisions of the copyright clause related to the text of the present document, oneM2M grants that users of the present document may freely reproduce the <proformatype> proforma in this {clause annex} so that it can be used for its intended purposes and may further publish the completed <proformatype>.
---

<PAGE BREAK>

---

## Annex <y>: Bibliography

*The annex entitled "Bibliography" is optional.*

*It shall contain a list of standards, books, articles, or other sources on a particular subject which are not mentioned in the document itself*

*It shall not include references mentioned in the document.*

*Use the **Heading 9 style** for the title and B1+ or Normal for the text.*

- <Publication>: "<Title>".

OR

<Publication>: "<Title>".

<PAGE BREAK>

---

## History

*This clause shall be the last one in the document and list the main phases (all additional information will be removed at the publication stage).*

Publication history		
V1.1.1	<dd-Mmm-yyyy>	<Milestone>

<b>Draft history</b> (to be removed on publication)		
v.0.1.0	2014-Jan-10	Initial version of the TS
v.0.1.1	2014-Mar-04	The first baseline TS with table of contents agreed with: 1. PRO-2014-0107R01-HTTP Binding TS TOC
v.0.2.0	2014-Apr-22	Includes an agreed contribution at PRO#9.3 meeting: 1. PRO-2014-0125R01-HTTP_Basic_Flows  Includes agreed contributions at PRO#10 F2F meeting: 1. PRO-2014-0131R03-HTTP-REST Overview 2. PRO-2014-0147-HTTP_binding_TS_clause2_clause3 3. PRO-2014-0148R01-HTTP_binding_TS_clause5 4. PRO-2014-0149R02-HTTP_binding_TS_clause6_clause7 5. PRO-2014-0159R03-HTTP Authentication 6. PRO-2014-0160R03-WebSocket based Notification  Includes Rapporteur's input: 1. Re-numbering clause numbers, figure numbers in clause 5 and a table number in clause 6.1
v0.3.2	2014-Jun-18	Includes agreed contribution at PRO#10.6 and PRO#10.7 meetings: 1. PRO-2014-0201R02-CRUD_mapping_on_HTTP 2. PRO-2014-0207-more_mapping_of_HTTP_status_codes  Includes agreed contributions at PRO#11 meeting: 1. PRO-2014-0226R01-Send_Request_in_HTTP 2. PRO-2014-0240-TS-0009 cleanup
v0.4.0	2014-Jul-30	Includes agreed contributions at PRO#12 meeting: 1. PRO-2014-0352R01-TS-0009_overview_of_HTTP_binding 2. PRO-2014-0353R01-TS-0009_header_mapping 3. PRO-2014-0354R03-TS-0009_cleanup 4. PRO-2014-0358R03-TS-0009_body_mapping 5. PRO-2014-0407-TS-0009_response_code
v0.4.3	2014-Aug-01	Correcting clause number in 7.2