

OCF “Dubai” – Remove all reference to CREATE with PUT from the Core Spec – Core Technology WG CR 2747

Legal Disclaimer

THIS IS A DRAFT SPECIFICATION DOCUMENT ONLY AND HAS NOT BEEN ADOPTED BY THE OPEN CONNECTIVITY FOUNDATION. THIS DRAFT DOCUMENT MAY NOT BE RELIED UPON FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT OF THIS DRAFT DOCUMENT. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT DOCUMENT IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT DOCUMENT AT ANY TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT DOCUMENT IN ANY WAY, INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES. IMPLEMENTATION OF THIS DRAFT DOCUMENT IS DONE AT YOUR OWN RISK AMEND AND IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE. IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS TO THIS DRAFT DOCUMENT, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED HERewith INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL AS CLAIMS OF DETRIMENTAL RELIANCE.

The OCF logo is a trademark of Open Connectivity Foundation, Inc. in the United States or other countries. *Other names and brands may be claimed as the property of others.

Copyright © 2019 Open Connectivity Foundation, Inc. All rights reserved.

Copying or other form of reproduction and/or distribution of these works are strictly prohibited.

1

2

3

4

5

6

7

8

9

10

11

12 Messaging

12.1 Introduction

This clause specifies the protocol messaging mapping to the CRUDN messaging operations (Clause 8) for each messaging protocol specified (e.g., CoAP.). Mapping to additional protocols is expected in later version of this document. All the Property information from the Resource model shall be carried within the message payload. This payload shall be generated in the Resource model layer and shall be encapsulated in the data connectivity layer. The message header shall only be used to describe the message payload (e.g., verb, mime-type, message payload format), in addition to the mandatory header fields defined in messaging protocol (e.g., CoAP) specification. If the message header does not support this, then this information shall also be carried in the message payload. Resource model information shall not be included in the message header structure unless the message header field is mandatory in the messaging protocol specification.

When a Resource is specified with a RESTful description language like OpenAPI 2.0 then the HTTP syntax definitions are used in the description (e.g., HTTP syntax for the CRUDN operations, status codes, etc). The HTTP syntax will be mapped to the actual used web transfer protocol (e.g., CoAP).

12.2 Mapping of CRUDN to CoAP

12.2.1 Overview

A Device implementing CoAP shall conform to IETF RFC 7252 for the methods specified in clause 12.2.3. A Device implementing CoAP shall conform to IETF RFC 7641 to implement the CoAP Observe option. Support for CoAP block transfer when the payload is larger than the MTU is defined in 12.2.8.

12.2.2 URIs

An OCF: URI is mapped to a coap: URI by replacing the scheme name "ocf" with "coap" if unsecure or "coaps" if secure before sending over the network by the requestor. Similarly on the receiver side, the scheme name is replaced with "ocf".

Any query string that is present within the URI is encoded as one or more URI-Query Options as defined in IETF RFC 7252 clause 6.4.

12.2.3 CoAP method with request and response

12.2.3.1 Overview

Every request has a CoAP method that realizes the request. The primary methods and their meanings are shown in Table 43, which provides the mapping of GET/POST/DELETE methods to CREATE, RETRIEVE, UPDATE, and DELETE operations. The associated text provides the generic behaviours when using these methods, however Resource OCF Interfaces may modify these generic semantics. The HTTP codes in the RESTful descriptions will be translated as described in IETF RFC 8075 clause 7 Response Code Mapping. CoAP methods not listed in Table 43 are not supported.

Table 1 – CoAP request and response

Method for CRUDN	(mandatory) Request data	(mandatory) Response data
GET for RETRIEVE	<ul style="list-style-type: none"> - Method code: GET (0.01) - Request URI: an existing URI for the Resource to be retrieved 	<ul style="list-style-type: none"> - Response code: success (2.xx) or error (4.xx or 5.xx) - Payload: Resource representation of the target Resource (when successful)
POST for CREATE	<ul style="list-style-type: none"> - Method code: POST (0.02) - Request URI: an existing URI for the Resource responsible for the creation - Payload: Resource presentation of the Resource to be created 	<ul style="list-style-type: none"> - Response code: success (2.xx) or error (4.xx or 5.xx) - Payload: the URI of the newly created Resource (when successful).
POST for UPDATE	<ul style="list-style-type: none"> - Method code: POST (0.02) - Request URI: an existing URI for the Resource to be updated. - Payload: representation of the Resource to be updated. 	<ul style="list-style-type: none"> - Response Code: success (2.xx) or error (4.xx or 5.xx)
DELETE for DELETE	<ul style="list-style-type: none"> - Method code: DELETE (0.04) - Request URI: an existing URI for the Resource to be deleted. 	<ul style="list-style-type: none"> - Response code: success (2.xx) or error (4.xx or 5.xx)

12.2.3.2 CREATE with POST

POST shall be used only in situations where the request URI is valid, that is it is the URI of an existing Resource on the Server that is processing the request. If no such Resource is present, the Server shall respond with an error response code of 4.xx. The use of POST for CREATE shall use an existing request URI which identifies the Resource on the Server responsible for creation. The URI of the created Resource is determined by the Server and provided to the Client in the response.

A Client shall include the representation of the new Resource in the request payload. The new resource representation in the payload shall have all the necessary Properties to create a valid

Resource instance, i.e. the created Resource should be able to properly respond to the valid Request with mandatory OCF Interface (e.g., "GET with ?if=oic.if.baseline").

Upon receiving the POST request, the Server shall either:

- Create the new Resource with a new URI, respond with the new URI for the newly created Resource and a success response code (2.xx); or
- respond with an error response code (4.xx or 5.xx).

POST is unsafe and is the supported method when idempotent behaviour cannot be expected or guaranteed.

12.2.3.3 RETRIEVE with GET

GET shall be used for the RETRIEVE operation. The GET method retrieves the representation of the target Resource identified by the request URI.

Upon receiving the GET request, the Server shall either:

- Send back the response with the representation of the target Resource with a success response code (2.xx); or
- respond with an error response code (4.xx or 5.xx) or ignore it (e.g. non-applicable multicast GET).

GET is a safe method and is idempotent.

12.2.3.4 UPDATE with POST

POST shall be used only in situations where the request URI is valid, that is it is the URI of an existing Resource on the Server that is processing the request. If no such Resource is present, the Server shall respond with an error response code of 4.xx. A client shall use POST to UPDATE Property values of an existing Resource.

Upon receiving the request, the Server shall either:

- Apply the request to the Resource identified by the request URI in accordance with the applied OCF Interface (i.e. POST for non-existent Properties is ignored) and send back a response with a success response code (2.xx); or
- respond with an error response code (4.xx or 5.xx). Note that if the representation in the payload is incompatible with the target Resource for POST using the applied OCF Interface (i.e. the overwrite semantic cannot be honored because of read-only Property in the payload), then the error response code 4.xx shall be returned.

POST is unsafe and is the supported method when idempotent behaviour cannot be expected or guaranteed.

12.2.3.5 DELETE with DELETE

DELETE shall be used for DELETE operation. The DELETE method requests that the Resource identified by the request URI be deleted.

Upon receiving the DELETE request, the Server shall either:

- Delete the target Resource and send back a response with a success response code (2.xx); or
- respond with an error response code (4.xx or 5.xx).

DELETE is unsafe but idempotent (unless URIs are recycled for new instances).

DRAFT