

**OCF 2.3 – new Interface for CREATE, DELETE & LINK\_UPDATE operations – CTWG CR 1229**

## 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 © 2018 Open Connectivity Foundation, Inc. All rights reserved.

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

\*\*\*\*\* First Change

### 7.6.3 OCF Interface methods

#### 7.6.3.1 Overview

The defined OCF Interfaces are listed in the table below:

**Table 1. Standard OCF Interfaces**

OCF Interface	Name	Applicable Operations	Description
baseline	"oic.if.baseline"	RETRIEVE, NOTIFY, UPDATE	The baseline OCF Interface defines a view into all Properties of a Resource including the Meta Properties. This OCF Interface is used to operate on the full Representation of a Resource.
links list	"oic.if.ll"	RETRIEVE, NOTIFY	The 'links list' OCF Interface provides a view into Links in a Collection (Resource). Since Links represent relationships to other Resources, the links list OCF Interface may be used to discover Resources with respect to a context. The discovery is done by retrieving Links to these Resources. For example: the Core Resource "/oic/res" uses this OCF Interface to allow discovery of Resource "hosted" on a Device.
batch	"oic.if.b"	RETRIEVE, NOTIFY, UPDATE	The batch OCF Interface is used to interact with a Collection of Resources at the same time. This also removes the need for the Client to first discover the Resources it is manipulating – the Server forwards the requests and aggregates the responses
read-only	"oic.if.r"	RETRIEVE NOTIFY	The read-only OCF Interface exposes the Properties of a Resource that may be 'read'. This OCF Interface does not provide methods to update Properties or a Resource and so can only be used to 'read' Property Values.
read-write	"oic.if.rw"	RETRIEVE, NOTIFY, UPDATE	The read-write OCF Interface exposes only those Properties that may be both 'read' and "written" and provides methods to read and write the Properties of a Resource.
actuator	"oic.if.a"	RETRIEVE, NOTIFY, UPDATE	The actuator OCF Interface is used to read or write the Properties of an actuator Resource.
sensor	"oic.if.s"	RETRIEVE, NOTIFY	The sensor OCF Interface is used to read the Properties of a sensor Resource.
create	"oic.if.create"	RETRIEVE, CREATE, NOTIFY	The create OCF Interface is used to create new Resources in a Collection. Both the Resource and the Link pointing to it are created in a single atomic operation.
delete	"oic.if.delete"	RETRIEVE, DELETE, NOTIFY	The delete OCF Interface is used to delete Resources. If a Collection is deleted, all of the items within it are recursively deleted.
Link add	"oic.if.linkadd"	RETRIEVE, UPDATE, NOTIFY	The Link add OCF Interface is used to add Links to a Collection.
Link remove	"oic.if.linkremove"	DELETE, RETRIEVE, NOTIFY	The Link remove OCF Interface is used to remove Links from a Collection.

\*\*\*\*\* Second Change

### **7.6.3.9 Create OCF Interface**

#### **7.6.3.9.1 Overview**

The create OCF Interface is used to create Resource instances in a Collection. An instance of a Resource and a Link pointing to the Resource are created together, atomically, according to a Client-supplied representation.

The create OCF Interface name is "oic.if.create".

A Collection which restricts which Resource Types it supports for created Resources shall expose the "rts" Property. If a Client attempts to create a Resource Type which is not supported by the Collection, the Server shall return an appropriate error status code, for example "Bad Request".

Successful CREATE operations shall return an appropriate success code, for example "Created".

#### **7.6.3.9.2 Data Format for CREATE**

The data format for the create OCF Interface is similar to the data format for the batch OCF Interface. The create OCF Interface format consists of a set of Link Parameters and a "rep" Parameter which contains a representation for the created Resource.

The representation supplied for the Link pointing to the newly created Resource shall contain at least the "rt" and "if" Link Parameters as shown in the first two elements in the example below.

The Link Parameter "p" should be included in representations supplied for all created Resources. If the "Discoverable" bit is set, then the supplied Link representation shall be exposed in "/oic/res" of the Device on which the Resource is being created.

Creating a discoverable Resource is the only way to add a Link to "/oic/res".

If the "p" Parameter is not included, the Server shall create the Resource using the default settings of not discoverable, and not observable.

The representation supplied for a created Resource in the value of the "rep" Parameter shall include the "if" Property and the "rt" Property, and shall contain all mandatory Properties required by the Resource Type to be created.

Note that the mandatory "rt" and "if" Property Values are included in both the Link Parameters and the Resource Properties of the Resource creation payload.

If the supplied representation does not contain all of the required Properties and Link Parameters, or if the Link Parameters for "rt" and "if" fail to match the corresponding Resource properties, the Server shall return an appropriate error status code, for example "Bad Request".

Resources that are created, and which support delete operations, should be created with the delete OCF Interface included as shown in the example.

### **Example Representation supplied to a CREATE operation**

```
{
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
  "p": {"bm": 3},
  "rep": {
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
    "temperature": 20
  }
}
```

The representation returned when a Resource is successfully created shall contain the "href", "if", and "rt" Link Parameters in addition to all Link Parameters and Resource Properties that were supplied by the Client in the CREATE operation. The Server may include additional Link Parameters and Properties in the created Resource as required by the application-specific Resource Type. The Server shall assign an "ins" value to each created Link and shall include the "ins" Parameter in the representation of each created Link.

#### Example Response Representation returned from a successful CREATE operation

```
{
  "href": "/3755f3ac",
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
  "ins": 39724818
  "p": {"bm": 3},
  "rep": {
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
    "temperature": 20
  }
}
```

#### 7.6.3.9.3 Use with CREATE

The CREATE request shall be sent to the URI of the Collection in which the Resource is to be created. The query "?if=oic.if.create" shall be included in all CREATE operations.

The Server shall generate a URI for the created Resource and include the URI in the "href" Parameter of the created Link.

When a Server successfully completes a CREATE operation using the "oic.if.create" OCF Interface addressing a Collection, then the Server shall automatically add an "href" Property containing the local path of the created Resource to every "resources" element (in the /oic/sec/acl2 Resource) containing a "href" Property matching the local path of the Collection Resource.

#### Example CREATE

```
Method: POST
URI: /scenes/scene1?if=oic.if.create
Payload:
{
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
  "p": {"bm":3},
  "rep": {
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
    "temperature": 20
  }
}
Response: Created
Payload:
{
  "href": "/3755f3ac",
  "ins": 39724818,
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
  "p": {"bm":3},
  "rep": {
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
    "temperature": 20
  }
}
```

#### 7.6.3.9.4 Use with NOTIFY

NOTIFY may be used with the create OCF Interface. Notifications sent to observers shall only be sent as a result of successful CREATE operations. Notifications on the create OCF Interface shall contain the required payload for the CREATE response as illustrated in the example above. A successful NOTIFY shall return an appropriate success status code, for example "Created"..

#### Example NOTIFY on Create OCF Interface

```
Method: GET + Observe
URI: /scenes/scenel?if=oic.if.create

Response: Valid
Payload: (empty)

➔ Later when a Resource is created:

Response: Created
Payload:
{
  "href": "/3755f3ac",
  "ins": 39724818
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
  "p": {"bm":3},
  "rep": {
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a", "oic.if.baseline", "oic.if.delete"],
    "temperature": 20
  }
}
```

#### 7.6.3.9.4 Use with RETRIEVE

RETRIEVE is supported with the create OCF Interface. A successful RETRIEVE shall return a status code indicating success with empty payload (e.g. Valid). A successful Observe operation, consisting of a RETRIEVE with the Observe option, shall return an appropriate success status code with an empty payload, for example "Valid".

#### 7.6.3.9.5 Use with UPDATE and DELETE

UPDATE and DELETE operations are not supported by the create OCF Interface. Attempts to perform UPDATE or DELETE operations using the create OCF Interface shall return an appropriate error status code, for example "Method Not Allowed".

#### 7.6.3.10 Delete OCF Interface

##### 7.6.3.10.1 Overview

The delete OCF Interface is used to enable removal of Resources from the Server.

The delete OCF Interface name is "oic.if.delete".

The Delete Interface is enabled on a Resource by including oic.if.delete in both the Link Parameters and Baseline Interface Properties in the representation of the Resource to be created.

##### 7.6.3.10.2 Use with DELETE

DELETE is performed using the URI of the Resource to be removed. The query "?if=oic.if.delete" shall be included in all DELETE requests. Upon successful processing of a DELETE operation, the Server shall return an appropriate success status code, for example "Deleted".

When a Client performs a DELETE operation, the Server shall remove the Resource pointed to by the URI of the DELETE operation and, if the Resource is a Collection, the Server shall recursively delete all Resources that were created in the Collection.

When a Client performs a DELETE operation on a Resource, the Server shall remove the Link which was created in the Collection where the Resource was created. If the Resource is Discoverable, the

Server shall remove the Link that was created in "/oic/res" and which points to the Resource being removed.

Deleting a discoverable Resource is the only way to remove a Link from "/oic/res".

Upon successful removal of a Resource, the Server shall return a response payload which includes the URI of the deleted Resource in the "href" parameter.

Upon successful removal of a Resource, the Server shall cancel all outstanding Observe operations on the deleted Resource by sending an appropriate error status code, for example "Not Found", in a NOTIFY response to each Observing Client.

#### Example DELETE and response payload

```
Method: DELETE
URI: /3755f3ac?if=oic.if.delete
Payload:
Response: Deleted
Payload:
{
  "href": "/3755f3ac"
}
```

If a Client attempts any reference to a deleted Resource, the Server shall return an appropriate error status code, for example "Not Found".

If a DELETE attempt is unsuccessful, the Resource state shall be unchanged, and the Server shall respond with an appropriate error status code, for example "Bad Request".

#### 7.6.3.10.3 Use with NOTIFY

A Client may use NOTIFY with the Delete OCF Interface, by performing a NOTIFY operation on the Resource to be deleted. Notifications shall only be transmitted upon successful DELETE operations. The Notification payload shall contain the "href" of the deleted Resource.

#### Example NOTIFY on Delete OCF Interface

```
Method: GET + Observe
URI: /3755f3ac?if=oic.if.delete

Response: Valid
Payload: (empty)

→ Later when a Resource is deleted:
Response: Deleted
Payload:
{
  "href": "/3755f3ac"
}
```

#### 7.6.3.10.4 Use with RETRIEVE

RETRIEVE is supported with the delete OCF Interface. A successful RETRIEVE shall return a status code indicating success with an empty payload (e.g. Valid). A successful Observe operation, consisting of a RETRIEVE with the Observe option, shall return an appropriate success status code with an empty payload, for example "Valid".

### **7.6.3.10.5 Use with UPDATE and CREATE**

UPDATE and CREATE operations are not supported by the delete OCF Interface. Attempts to perform UPDATE or CREATE operations using the delete OCF Interface shall return an appropriate error status code, for example "Method Not Allowed".

### **7.6.3.11 Link Add OCF Interface**

#### **7.6.3.11.1 Overview**

The Link add OCF Interface is used to add Links to a Collection.

The Link add OCF Interface name is "oic.if.linkadd".

Collections which support the Link add OCF Interface should support the Link remove OCF Interface.

If there is a need to modify a Link in a Collection, the Link may be removed and a new Link added to replace it.

The Link add OCF Interface shall not be exposed for "/oic/res".

#### **7.6.3.11.2 Use with UPDATE**

To add Links to a Collection, the UPDATE operation is used with a payload containing one or more Links. Links are added to a Collection to point to Resources which are expected to already exist. Links shall point to OCF Resources that expose valid "rt" and "if" Parameters.

The Client shall include the "href" Link Parameter for created Links. Links that are added to a Collection should include all of the "rt" and "if" values supported by the Resource pointed to by the Link.

If the Client does not include all of the required Link Parameters, the Server shall return an appropriate error status code, for example "Bad Request".

The Server shall only allow the addition of Links to a Collection via the "oic.if.linkadd" OCF Interface when the Resource Type ("rt") Property of the to be added link matches a Resource Type defined in Table 7 in the OCF Resource Type Specification.

When Links are added to a Collection using the Link add OCF Interface, the Server shall return a payload consisting of the added Links. The Server shall assign an "ins" value to each created Link and shall include the "ins" Parameter in the representation of each created Link.

### **Example Add Links using the Link add OCF Interface**

```
Method: POST
URI: /scenes/scene1?if=oic.if.linkadd
Payload:
[
  {
    "href": "/sensor/temperature",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.s", "oic.if.baseline"]
  }
]
Response: Changed
Payload:
[
  {
    "href": "/sensor/temperature",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.s", "oic.if.baseline"],
    "ins": 37724819
  }
]
```

#### 7.6.3.11.3 Use with NOTIFY

A Client may use NOTIFY with the Link add OCF Interface, by performing a NOTIFY operation on the Collection Resource. Notifications shall only be transmitted upon successful Link add operations. The Notification payload shall consist of the required response payload for the successful Link add operation as specified above.

#### Example NOTIFY on addition of Links using the Link add OCF Interface

```
Method: GET + Observe
URI: /scenes/scene1?if=oic.if.linkadd

Response: Valid
Payload: (empty)

➔ Later when a Link is added:
Response: Changed
Payload:
[
  {
    "href": "/sensor/temperature",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.s", "oic.if.baseline"],
    "ins": 37724819
  }
]
```

#### 7.6.3.11.4 Use with RETRIEVE

RETRIEVE is supported with the Link add OCF Interface. A successful RETRIEVE shall return a status code indicating success with an empty payload (e.g. Valid). A successful Observe operation,

consisting of a RETRIEVE with the Observe option, shall return an appropriate success status code with an empty payload, for example "Valid".

#### 7.6.3.11.5 Use with DELETE and CREATE

DELETE and CREATE operations are not supported by the Link add OCF Interface. Attempts to perform DELETE or CREATE operations using the Link add OCF Interface shall return an appropriate error status code, for example "Method Not Allowed".

#### 7.6.3.12 Link Remove OCF Interface

##### 7.6.3.12.1

The Link remove OCF Interface is used to remove Links from a Collection.

The Link remove OCF Interface name is "oic.if.linkremove".

The Link remove OCF Interface shall not be exposed for "/oic/res".

##### 7.6.3.12.2 Use with DELETE

To remove Links from a Collection, the DELETE operation is used. A Client shall send a DELETE operation to the Server with an empty payload, and shall include an "ins" URI query Parameter to select a Link for removal. Only one Link may be removed at a time using the Link remove OCF Interface.

If the removal of the Link is successful, the Server shall return an appropriate success code and a payload containing a representation of the Link that was removed.

If the removal of the Links is unsuccessful, or if no Links match the "ins" query Parameter, or the query Parameter is omitted, the Server shall respond with an appropriate error code, for example "Bad Request", and should include a diagnostic payload indicating the failure.

#### Example Delete Links using the Link remove OCF Interface

```
Method: DELETE
URI: /scenes/scene1?if=oic.if.linkremove&ins=37724819
Payload:
(no payload)
Response: Deleted
Payload:
[
  {
    "href": "/sensor/temperature",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.s", "oic.if.baseline"],
    "ins": 37724819
  }
]
```

##### 7.6.3.12.3 Use with NOTIFY

A Client may use NOTIFY with the Link remove OCF Interface, by performing a NOTIFY operation on the Collection Resource. Notifications shall only be transmitted upon successful Link remove operations. The Notification payload shall consist of the required response payload for the successful operation as specified above.

#### Example NOTIFY on Removal of Links using the Link remove OCF Interface

```
Method: GET + Observe
URI: /scenes/scene1?if=oic.if.linkremove

Response: Valid
Payload: (empty)

➔ Later when a Link is removed:
Response: Deleted
Payload:
[
  {
    "href": "/sensor/temperature",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.s", "oic.if.baseline"],
    "ins": 37724819
  }
]
```

#### 7.6.3.12.4 Use with RETRIEVE

RETRIEVE is supported with the Link remove OCF Interface. A successful RETRIEVE shall return a status code indicating success with an empty payload (e.g. Valid). A successful Observe operation, consisting of a RETRIEVE with the Observe option, shall return an appropriate success status code with an empty payload, for example "Valid".

#### 7.6.3.12.5 Use with UPDATE and CREATE

UPDATE and CREATE operations are not supported by the Link remove OCF Interface. Attempts to perform UPDATE or CREATE operations using the Link remove OCF Interface shall return an appropriate error status code, for example "Method Not Allowed".

\*\*\*Third Change

#### 12.2.3.2 CREATE with POST or PUT

##### With POST

Only POST shall be used to create OCF Resources. POST shall only be used 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.~~

#### ~~With PUT~~

~~PUT shall be used to create a new Resource or completely replace the entire representation of an existing Resource. The resource representation in the payload of the PUT request shall be the complete representation. PUT for CREATE shall use a new request URI identifying the new Resource to be created.~~

~~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 Interface (e.g. "GET with ?if=oic.if.baseline").~~

~~Upon receiving the PUT request, the Server shall either~~

- ~~• create the new Resource with the request URI provided in the PUT request and send back a response with a success response code (2.xx); or~~
- ~~• respond with an error response code (4.xx or 5.xx).~~

~~PUT is an unsafe method but it is idempotent, thus when a PUT request is repeated the outcome is the same each time.~~