



TM Forum Specification

Resource Inventory API REST Specification

TMF639
Release 17.0.1
December 2017

Latest Update: Release 17.0	TM Forum Approved
Version 1.0.1	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2017. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website, but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100
Parsippany, NJ 07054 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE	2
Table of Contents.....	4
List of Tables	7
Introduction	8
SAMPLE USE CASES	9
Resource inventory query for a party.....	9
Resource inventory update as part of resource provisioning.....	9
RESOURCE MODEL	10
Managed Entity and Task Resource Models	10
Resource	10
Logical Resource.....	16
Physical Resource.....	17
Notification Resource Models	20
Resource Creation Notification	21
Resource Attribute Value Change Notification	22
Resource State Change Notification	22
Resource Remove Notification	22
{Resource-Type} Creation Notification.....	23
{Resource-Type} Attribute Value Change Notification.....	23
{Resource-Type} State Change Notification.....	24
{Resource-Type} Remove Notification	24
API OPERATIONS.....	25
Operations on Resource.....	25
List resources	26
Retrieve Resource.....	28
Create resource.....	31

Patch resource	33
PUT resource	34
Delete resource	36
Operations on Logical Resource.....	37
List Logical resources	37
Retrieve Logical Resource.....	38
Create Logical resource.....	39
Patch Logical resource	40
PUT logical resource	42
Delete Logical resource	43
Operations on Physical Resource.....	44
List Physical resources	44
Retrieve Physical Resource.....	45
Create Physical resource.....	47
Patch Physical resource	48
PUT Physical resource	50
Delete Physical resource	52
Operations on {Resource-Type}	53
List {resource-type}.....	53
Retrieve {resource-type}	53
Create {resource-type}.....	54
PATCH {resource-type}	56
PUT {resource-type}	58
Delete {resource-type}.....	60
API NOTIFICATIONS.....	61
Register listener	61
Unregister listener	62

Publish Event to listener	62
Acknowledgements	64
Release History	64

LIST OF TABLES

N/A

INTRODUCTION

The following document is intended to provide details of the REST API interface for Resource Inventory. The intent of this API is to provide a consistent/standardized mechanism to query and manipulate the Resource inventory.

SAMPLE USE CASES

Resource inventory query for a party

The Resource Inventory API can be used to query the resource instances for a party playing the role of customer via Self Service Portal or the Call Centre operator can query the resource instances on behalf of the customer while a customer may have a complaint or a query.

Resource inventory update as part of resource provisioning

The Resource Inventory API can be called by the Resource Order Management to create a new resource instance/ update an existing resource instance in the Resource Inventory.

RESOURCE MODEL

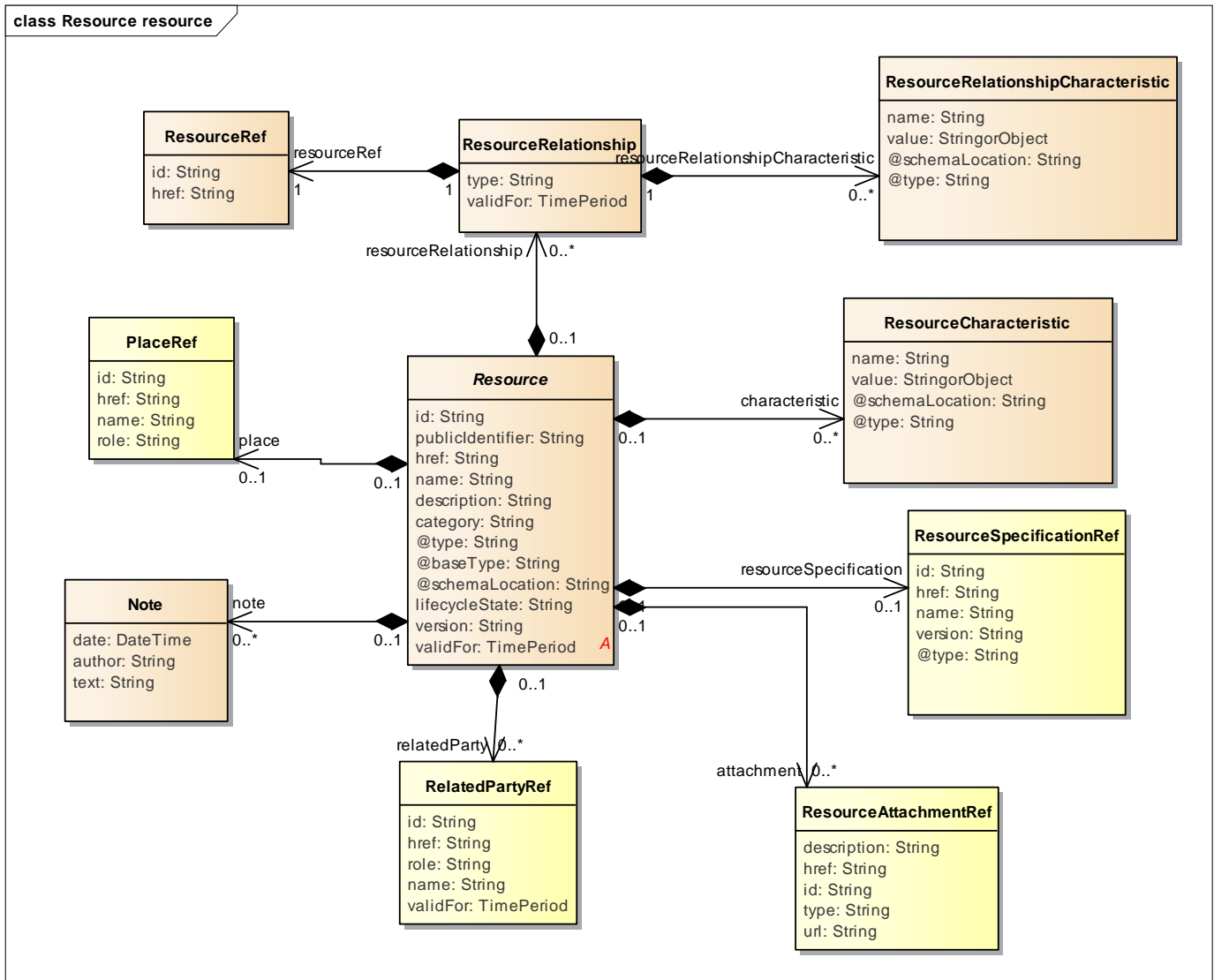
Managed Entity and Task Resource Models

RESOURCE

Resource is an abstract entity. The intent is to describe the common set of attributes shared by all concrete resources (e.g. TPE, EQUIPMENT) in the inventory.

The **@type** attribute must be implemented by all concrete resources and is a reserved attribute. Its value of is equal to the name of the resource. For example a EQUIPMENT resource has the @type value equal to "EQUIPMENT".

Resource model



Field descriptions

Resource fields

href	A string. Reference of the resource.
id	A string. "id" is the ID created for the resource.
publicIdentifier	A string. Public Identifier of the resource. E.g 07467336644 (the value of an MSISDN)
description	A string. free-text description of the resource.
category	A string. Category of the concrete resource. e.g Gold, Silver for MSISDN concrete resource.
validFor	A time period. The period for which the resource is valid
name	A string. "name" is the name of the resource.
lifecycleState	A string. The life cycle state of the resource
@type	Indicates the (class) type of resource. For physical resource this will be 'Equipment', For logical Resource this can be 'Resource Function', 'MSISDN', 'IP Address'.
@baseType	A string. Indicates the base (class) type of the resource. For resources pf type 'Equipment', baseType can be 'PhysicalResource'.
@schemaLocation	A string. This field provides a link to the schema describing this REST resource
version	A string. Version number
resourceRelationship	A list of resource relationships (ResourceRelationship [*]). A migration, substitution, dependency or exclusivity relationship between/among resources.
place	A list of places (Place [*]). Used to defined a place useful for the resource (for example a delivery geographical place).
note	A list of notes (Note [*]). Extra information about the resource instance
resourceSpecification	A resource specification reference (ResourceSpecificationRef). ResourceSpecification(s) required to realize a ServiceSpecification.
relatedParty	A list of related party references (RelatedPartyRef [*]). A related party defines party or party role linked to a specific entity.

resourceCharacteristic	A list of resource characteristics (ResourceCharacteristic [*]).
resourceAttachment	A list of resource Attachment (ResorceAttachment[*]).Complements the description of an element (for instance a product) through video, pictures...

Note sub-resource

Extra information about the resource.

author	A string. Author of the note.
date	A date time (DateTime). Date of the note.
text	A string. Text of the note.

PlaceRef sub-resource

Used to defined a place useful for the resource (for example a delivery geographical place).

href	A string. Reference of a place.
id	A String. Unique identifier of a place.
name	A String. Name of a place.
role	A string. Role of the place (for instance delivery geographical place).

ResourceCharacteristic sub-resource

Describes the resource characteristics.

name	A string. Name of the characteristic.
value	A string (StringorObject). Value of the resource characteristic which can be simple or complex (object).
@schemaLocation	A string. A link to the schema describing this characteristic
@type	A string. (Class) type of the characteristic.

ResourceRelationship sub-resource

Describes links with other resources

type	A string. Describes links with other resource
validFor	A time period. A period for which the resource relationship is valid for.
ResourceRef	A resource reference (ResourceRef). The Resource that is the target of this ResourceRelationship.

ResourceRelationshipCharacteristic sub-resource

Describes the resourceRelationship characteristics.

name	A string. Name of the characteristic.
value	A string (StringorObject). Value of the characteristic which can be simple or complex (object)
@schemaLocation	A string. A link to the schema describing this characteristic
@type	A string. (Class) type of the characteristic

RelatedPartyRef relationship

RelatedParty reference. A related party defines party or party role linked to a specific entity.

href	A string. Reference of the related party, could be a party reference or a party role reference.
id	A string. Unique identifier of a related party.
name	A string. Name of the related party.
role	A string. Role of the related party.
validFor	A time period. Validity period of the related party.

ResourceRef relationship

Resource reference. Useful to link resources.

href	A string. Reference of the resource.
id	A string. Id of the resource.

ResourceAttachment sub-resource

Complements the description of an element (for instance a product) through video, pictures...

description	A string. A narrative text describing the content of the attachment.
href	A string. Reference of the attachment.
id	A string. Unique identifier of the attachment.
type	A string. Attachment type such as video, picture.
url	A string. Uniform Resource Locator, is a web page address (a subset of URI).

ResourceSpecificationRef relationship

Resource specification reference

href	A string. Reference of the ResourceSpecification.
------	---

id	A string. Unique identifier of the ResourceSpecification.
name	A string. Name of the required ResourceSpecification.
version	A string. Resource specification version.
@type	A string, Indicates the (class) type of ResourceSpecification. For resource baseType LogicalResource this can be 'LogicalResource Specification'

Lifecycle

Here is the state machine diagram for a Resource (common for all type of resources). This material is based on TR255.

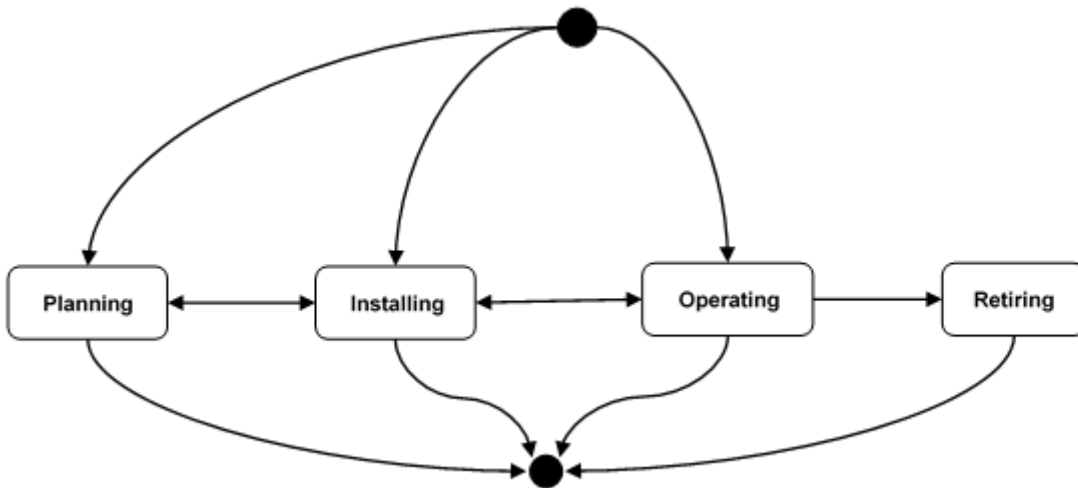


Figure 1 – Resource state model

The following table provides the state description for resources:

Planning	The resource is scheduled for deployment in accordance with a specific plan.
Installing	The physical resource undergoes a full commissioning process until it is finally ready for work. Resources may also be installed in the network regardless of any specific plan
Operating	The resource is fully provisioned and ready to support consumers (assuming the entity is administratively activated and operationally working).
Retiring	The resource undergoes all necessary procedures for its decommissioning and phasing out.

The following table provides the sub-states description for planning composite state:

Proposed	A requirement for the resource has been identified, and the resource has been proposed to address the requirement, but entity characteristics or deployment details have not been agreed.
Feasibility Checked	A check has been to be done to see if the proposed resource can be instantiated as requested. At this point, the design of the resource is not complete and nothing has been ordered with regard to the given resource
Designed	The characteristics of the entity and its deployment have been completely identified but nothing exists in the network at this point in support of the resource. Firm agreement has been reached to satisfy a requirement using the resource
Ordered	An order for delivery of an entity type or an instance of an existing entity type has been agreed.

The following table provides the sub-states description for Operating composite state:

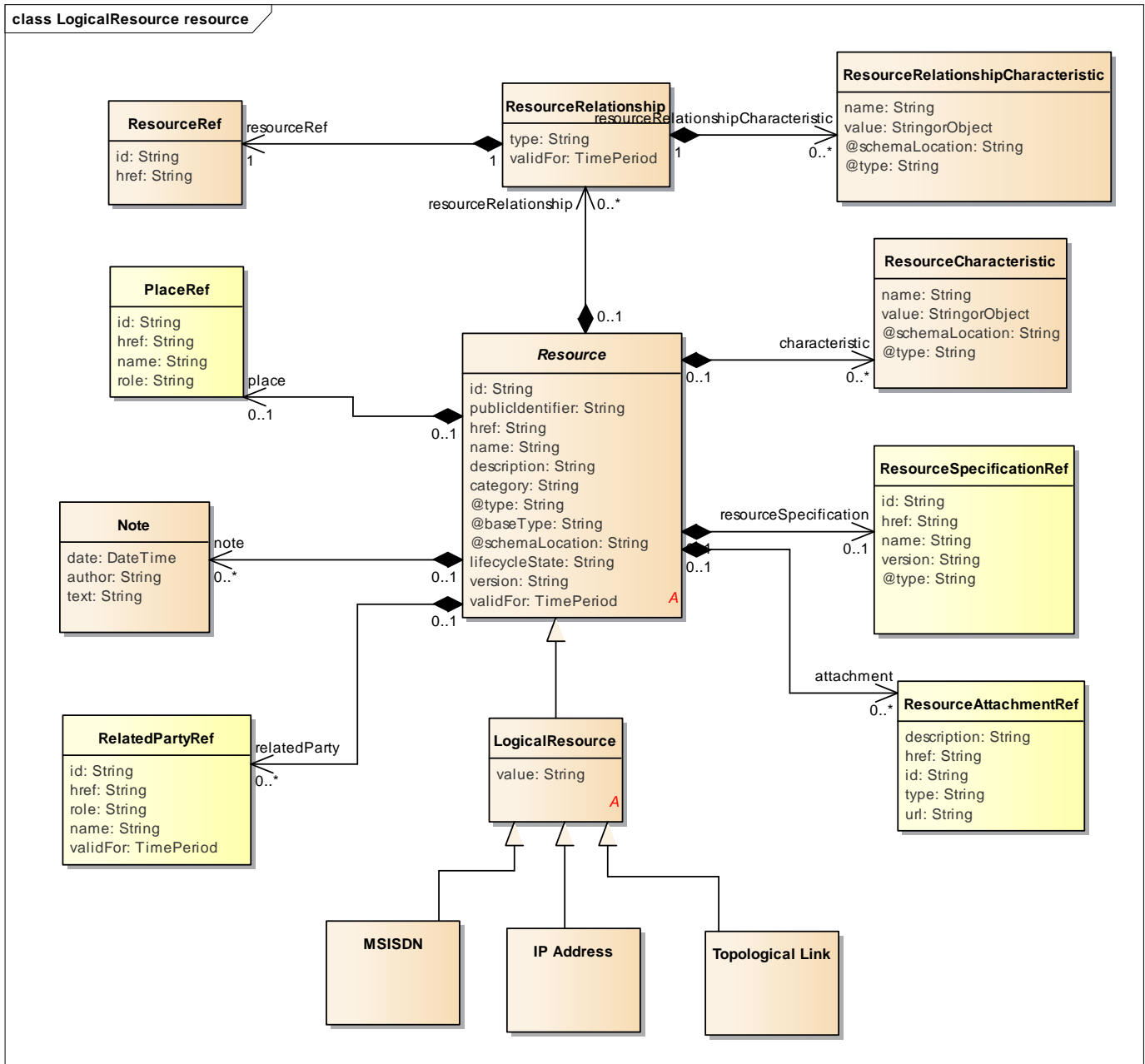
Administrative	<p>The states listed below are typically set by a management / control system or by some policy (e.g., automatically put the entity in Deactivated sub-state once it reaches Installing / Accepted).</p> <ul style="list-style-type: none"> Activated- the entity is working, has been configured and activated and can be used by client(s). The entity is fully operation in the sense that it can meet all requirements for which it was designed. Deactivated - the entity is working, but cannot be used by a client..
Operational	<p>The Operational sub-state gives an indication of how well (or how full) the entity is functioning.</p> <ul style="list-style-type: none"> Working – the resources is completely working as intended. Meeting All SLAs – the resource is presently meeting all the SLAs promised to each of its clients, but is not completely functional. Meeting Some SLAs – the resource is meeting only some of the SLAs promised to its clients Meeting No SLAs – the resource is meeting none of the SLAs promised to its clients Not Working – the resource is completely non-functionally.
Usage	<ul style="list-style-type: none"> Available – the resource is not in use and is available for reservation or assignment. Reserved – the resource is reserved for a client No Available – the resource is not available for reservation. It may be because te resources has been quarantined or port-out. Assigned – the resource has been assigned to one or more products and services

Note: This is for example only and not part of the normative standard

LOGICAL RESOURCE

Logic resource is a type of resource that describes the common set of attributes shared by all concrete logical resources (e.g. TPE, MSISDN, IP Addresses) in the inventory.

Resource model



Note: MSISDN, IP Address, Topological Link are shown on the diagram as examples only.

Field descriptions

LogicalResource fields

value The actual value of the logical resource. For example, for a logical resource of type MSISDN this field will contain the actual value of the MSISDN. e.g. 07474338833.

Json representation sample

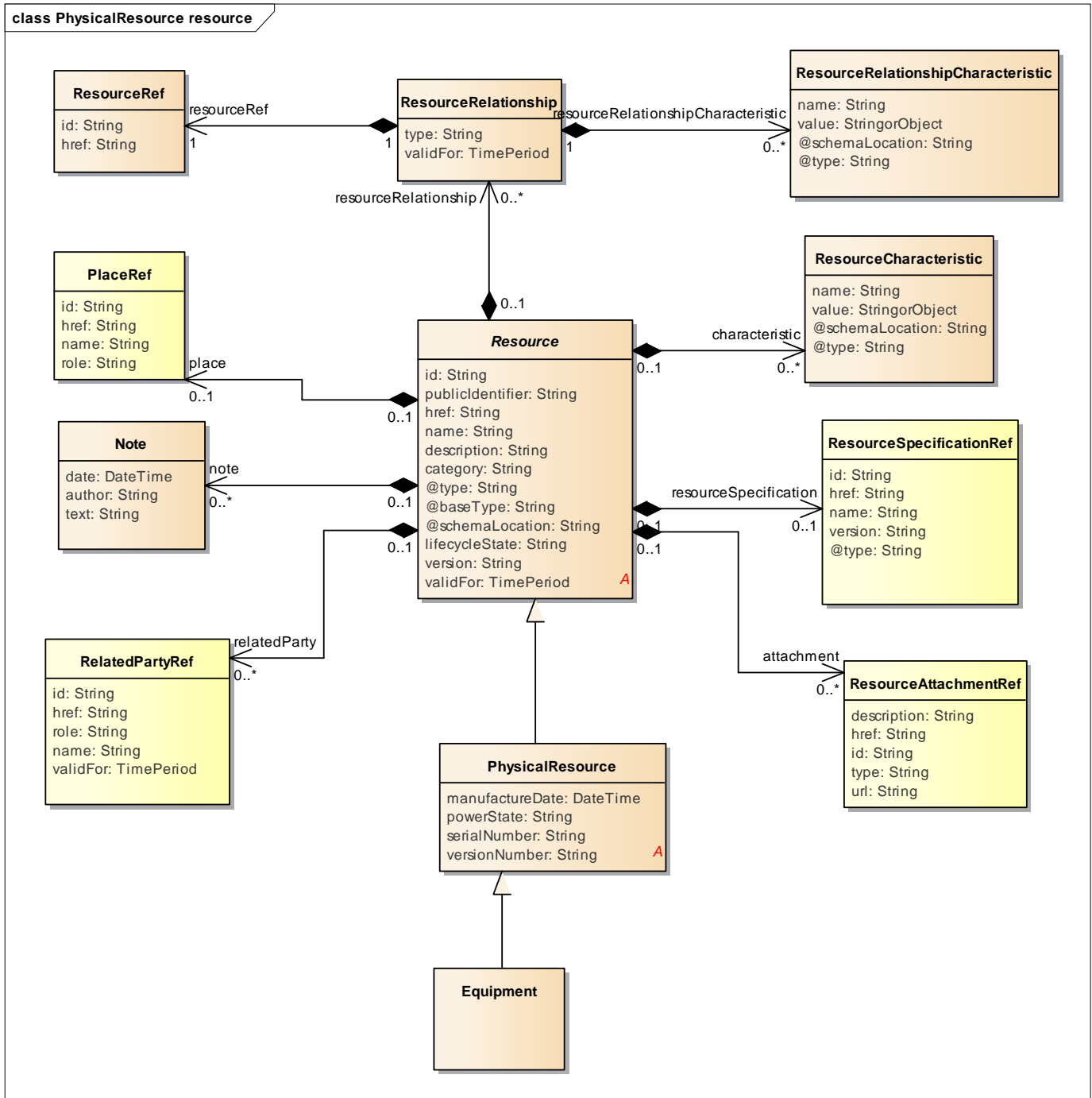
We provide below the json representation of an example (not a full resource representation) of a "Resource" resource object of type MSISDN and baseType ='LogicalResource'

```
{
  "id": "444",
  "href": "http://server:port/resourceInventoryManagement/logicalResource?id=444",
  "category": "Premium",
  "@type": "MSISDN",
  "@baseType": "LogicalResource",
  "value": "07465233456",
  "lifecycleState": "Available",
  "resourceSpecification": {
    "id": "4",
    "href": " http://server:port/resourceCatalogManagement/resourceSpecification/4",
    "@type": "LogicalResourceSpecification"
  },
  "resourceCharacteristic": [{
    "name": "premiumValue",
    "value": "gold"
  }],
  "relatedParty": [{
    "role": "Owner",
    "id": "42",
    "href": "http://serverlocation:port/PartyManagement/organisation/42"
  }
]}
```

PHYSICAL RESOURCE

Physical resource is a type of resource that describes the common set of attributes shared by all concrete physical resources (e.g. EQUIPMENT) in the inventory.

Physical resources can also represent things that are sold, leased, and so forth, like a phone and SIM card.



Field descriptions

PhysicalResource fields

- manufactureDate The date time of manufacture.
- powerState The power state of the physical resource

serialNumber	The serial number of the physical resource
versionNumber	The version number of the physical resource.

We provide below the json representation of an example (not a full resource representation) of a “Resource” resource object of type Equipment:

```
{
  "id": "45",
  "href": "http://server:port/resourceInventoryManagement/physicalResource/45",
  "publicIdentifier": "07467223333",
  "@type": "Equipment",
  "@baseType": "PhysicalResource",
  "@schemaLocation": "//server:port/resourceInventoryManagement/schema/Equipment.yml",
  "category": "Category 1",
  "lifecycleState": "Active",
  "manufactureDate": "2007-04-12",
  "serialNumber": "123456745644",
  "versionNumber": "11",
  "resourceSpecification": {
    "id": "6",
    "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6",
    "@type": "PhysicalResourceSpecification"
  },
  "resourceCharacteristic": [{
    "name": "physicalPort",
    "value": {
      "@type": "physicalPort",
      "name": "LAN Port",
      "isActive": true
    },
    "@schemaLocation": "//host:port/schema/physicalPort.yml"
  },
  {
    "name": "color",
    "value": "red"
  }
  ],
  "resourceRelationship": [{
    "type": "requires",
    "resource": {
      "id": "46",
      "href": " http://server:port/resourceInventoryManagement/logicalResource/46"
    },
    "resourceRelationshipCharacteristic": {
      "name": "priority",
      "value": 2
    }
  },
  {

```

```

        "name": "accuracy",
        "value": {
            "@type": "accuracy",
            "unit": "second",
            "amount": "5"
        },
        "@schemaLocation":
        "http://server:port/resourceInventoryManagement/schema/accuracy.yml"
    }
}
}],
"relatedParty": [{
    "role": "Manufacturer",
    "id": "43",
    "href": "http://serverlocation:port/PartyManagement/individual/43"
}],
"resourceAttachment": [{
    "id": "http://server:port/documentManagement/document/123"
}],
"note": [{
    "text": "something about this resource"
}],
"place": {
    "id": "1979",
    "href": "https://host:port/genericCommon/place/1979",
    "name": "Main Office",
    "role": "default delivery"
}
}
}

```

Notification Resource Models

The following notifications are defined for this API:

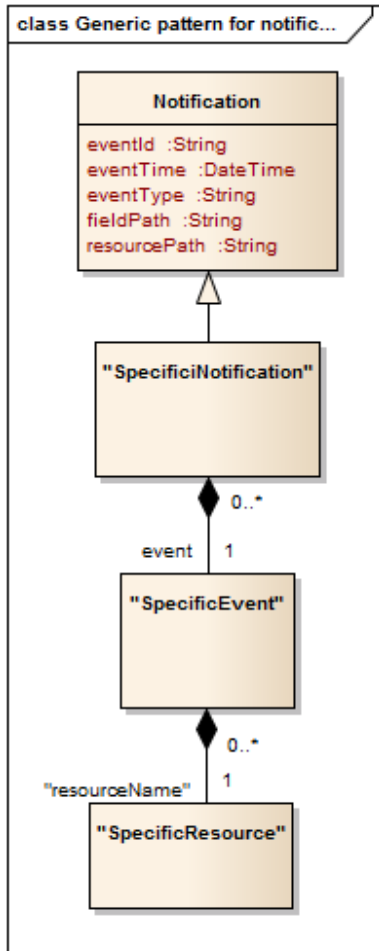
Notifications related to Resource:

- ResourceCreationNotification
- ResourceAttributeValueChangeNotification
- ResourceStateChangeNotification
- ResourceRemoveNotification

Notifications related to {Resource-type}:

- {Resource-Type}CreationNotification
- {Resource-Type}AttributeValueChangeNotification
- {Resource-Type}StateChangeNotification

The notification structure for all notifications in this API follow the pattern depicted by the figure below. A notification resource (depicted by "SpecificNotification" placeholder) is a sub class of a generic Notification structure containing an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the notification resource (eventType). This notification structure owns an event structure ("SpecificEvent" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



RESOURCE CREATION NOTIFICATION

Notification sent when a new Resource resource is created.

Json representation sample

We provide below the json representation of an example of a 'ResourceCreationNotification' notification object

```

{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ResourceCreationNotification",

```

```
"event": {
  "resource" :
    {-- SEE Resource RESOURCE SAMPLE --}
}
```

RESOURCE ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when changing an attribute of a Resource resource.

Json representation sample

We provide below the json representation of an example of a 'ResourceAttributeValueChangeNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ResourceAttributeValueChangeNotification",
  "event": {
    "resource" :
      {-- SEE Resource RESOURCE SAMPLE --}
  }
}
```

RESOURCE STATE CHANGE NOTIFICATION

Notification sent when changing the state of a Resource Resource.

Json representation sample

We provide below the json representation of an example of a 'ResourceStateChangeNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ResourceStateChangeNotification",
  "event": {
    "resource" :
      {-- SEE Resource RESOURCE SAMPLE --}
  }
}
```

RESOURCE REMOVE NOTIFICATION

Notification sent when removing a Resource resource.

Json representation sample

We provide below the json representation of an example of a 'ResourceRemoveNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ResourceRemoveNotification",
  "event": {
    "resource" :
      [-- SEE resource RESOURCE SAMPLE --]
  }
}
```

{RESOURCE-TYPE} CREATION NOTIFICATION

Notification sent when a new {Resource-Type} resource is created.

Json representation sample

We provide below the json representation of an example of a '{Resource-Type}CreationNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"{Resource-Type}CreationNotification",
  "event": {
    "resource" :
      [-- SEE {Resource-Type} RESOURCE SAMPLE --]
  }
}
```

{RESOURCE-TYPE} ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when changing an attribute of a {Resource-Type} resource.

Json representation sample

We provide below the json representation of an example of a '{Resource-Type}AttributeValueChangeNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"{Resource-Type}AttributeValueChangeNotification",
  "event": {
    "resource" :
      [-- SEE {Resource-Type} RESOURCE SAMPLE --]
  }
}
```

```
}
```

{RESOURCE-TYPE} STATE CHANGE NOTIFICATION

Notification sent when changing the state of a {Resource-Type} Resource.

Json representation sample

We provide below the json representation of an example of a '{Resource-Type}StateChangeNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"{Resource-Type}StateChangeNotification",
  "event": {
    "resource" :
      {-- SEE Resource RESOURCE SAMPLE --}
  }
}
```

{RESOURCE-TYPE} REMOVE NOTIFICATION

Notification sent when removing a Resource resource.

Json representation sample

We provide below the json representation of an example of a 'ResourceRemoveNotification' notification object

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"{Resource-Type}RemoveNotification",
  "event": {
    "resource" :
      {-- SEE {resource-type} RESOURCE SAMPLE --}
  }
}
```


API OPERATIONS

Remember the following Uniform Contract:

Operation on Entities	Uniform API Operation	Description
Query Entities	GET Resource	GET must be used to retrieve a representation of a resource.
Create Entity	POST Resource	POST must be used to create a new resource
Partial Update of an Entity	PATCH Resource	PATCH must be used to partially update a resource
Complete Update of an Entity	PUT Resource	PUT must be used to completely update a resource identified by its resource URI
Remove an Entity	DELETE Resource	DELETE must be used to remove a resource
Execute an Action on an Entity	POST on TASK Resource	POST must be used to execute Task Resources
Other Request Methods	POST on TASK Resource	GET and POST must not be used to tunnel other request methods.

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

OPERATIONS ON RESOURCE

LIST RESOURCES

GET /resource?fields=...&{filtering}

Description

This operation list resource entities.

Attribute selection is enabled for all first level attributes.

Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving resource instances. For example we can retrieve both logical and physical resources.

Get all resources of type equipment or MSISDN that belongs to party with id =42.

Request
<pre>GET /resourceInventoryManagement/resource?@type="MSISDN","Equipment" & relatedParty.id =42 Accept: application/json</pre>
Response
<pre>200 [[{ "id": "42", "href": "http://server:port/resourceInventoryManagement/resource/42", "@type": "MSISDN", "@basetype": "LogicalResource", "value": "07376456789", "lifecycleState": "Available", "resourceSpecification": { "id": "4", "href": "http://server:port/resourceCatalogManagement/resourceSpecification/4" }, "resourceCharacteristic": { "name": "Equipment", "value": { "@type": "Equipment", "serialNumber": "12444545544", "versionNumber": "1.22", "manufactureDate": "05-04-2017" } }, "@schemaLocation": "http://server:port/resourceInventoryManagement/schema/Equipment.yml" }]]</pre>

```

    "relatedParty": [{
      "role": "owner",
      "id": "42",
      "href": "http://serverlocation:port/customerManagement/individual/42"
    }],
    "resourceAttachment": [{
      "id": "http://server:port/documentManagement/document/123"
    }],
    "note": [{
      "text": "something about this resource"
    }],
    "place": {
      "id": "1979",
      "href": "https://host:port/genericCommon/place/1979",
      "name": "Main Office",
      "role": "default delivery"
    }
  },
  {
    "id": "45",
    "href": "http://server:port/resourceInventoryManagement/resource/45",
    "publicIdentifier": "07467223333",
    "@type": "Equipment",
    "@baseType": "PhysicalResource",
    "lifecycleState": "Active",
    "manufactureDate": "2007-04-12",
    "serialNumber": "123456745644",
    "versionNumber": "11",
    "resourceSpecification": {
      "id": "6",
      "href": "http://server:port/resourceCatalogManagement/resourceSpecification/6",
      "@type": "EquipmentSpecification"
    },
    "resourceCharacteristic": [{
      "name": "Equipment",
      "value": {
        "@type": "Equipment",
        "serialNumber": "12444545544",
        "versionNumber": "1.22",
        "manufactureDate": "05-04-2017"
      }
    }],
    "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
  },
  "resourceRelationship": [{
    "type": "requires",
    "resource": {
      "id": "46",
      "href": "http://server:port/resourceInventoryManagement/logicalResource/46"
    }
  }],
  "relatedParty": [{

```

```

        "role": "Manufacturer",
        "id": "43",
        "href": "http://serverlocation:port/PartyManagement/individual/43"
      },
      {
        "role": "owner",
        "id": "42",
        "href": "http://serverlocation:port/customerManagement/individual/42"
      }
    ],
    "resourceAttachment": [{
      "id": "http://server:port/documentManagement/document/123"
    }],
    "note": {
      "text": "something about this resource"
    },
    "place": {
      "id": "1979",
      "href": "https://host:port/genericCommon/place/1979",
      "name": "Main Office",
      "role": "default delivery"
    }
  }
]

```

RETRIEVE RESOURCE

GET /resource/{id}?fields=...&{filtering}

Description

This operation retrieves a resource entity.

Attribute selection is enabled for all first level attributes.

Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a single Resource from the Inventory providing as input publicIdentifier for a resource

Request

```
GET /resourceInventoryManagement/resource/publicIdentifier = "07455559833"
Accept: application/json
```

Response

200

```
{
  "id": "42",
  "href": "http://server:port/resourceInventoryManagement/resource/42",
  "publicIdentifier": "07455559833",
  "@type": "EQUIPMENT",
  "name": "1 - MAIN",
  "lifecycleState": "Active",
  "resourceSpecification": {
    "id": "4",
    "href": "http://server:port/resourceCatalogManagement/resourceSpecification/4",
    "@type": "EquipmentSpecification"
  },
  "resourceCharacteristic": [
    {
      "name": "Equipment",
      "value": {
        "@type": "Equipment",
        "serialNumber": "12444545544",
        "versionNumber": "1.22",
        "manufactureDate": "05-04-2017"
      },
      "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
    }
  ],
  "resourceRelationship": [{
    "type": "contains",
    "resource": {
      "id": "44",
      "href": "http://server:port/resourceInventoryManagement/resource/44"
    }
  }],
  "relatedParty": [{
    "role": "customer",
    "id": "42",
    "href": "http://serverlocation:port/customerManagement/individual/42"
  }],
  "resourceAttachment": [{
    "id": "http://server:port/documentManagement/document/123"
  }],
  "note": [{
```

```

        "text": "something about this resource"
    }
  },
  "place": {
    "id": "1979",
    "href": "https://host:port/genericCommon/place/1979",
    "name": "Main Office",
    "role": "default delivery"
  }
}

```

Here's an example of a request for retrieving a single Resource from the Inventory providing as input resource id.

Request

```

GET /resourceInventoryManagement/resource/42
Accept: application/json

```

Response

```

200
{
  "id": "42",
  "href": "http://server:port/resourceInventoryManagement/resource/42",
  "@type": "EQUIPMENT",
  "name": "1 - MAIN",
  "lifecycleState": "Active",
  "resourceSpecification": {
    "id": "4",
    "href": "http://server:port/resourceCatalogManagement/resourceSpecification/4",
    "@type": "EquipmentSpecification"
  },
  "resourceCharacteristic": [
    {
      "name": "Equipment",
      "value": {
        "@type": "Equipment",
        "serialNumber": "12444545544",
        "versionNumber": "1.22",
        "manufactureDate": "05-04-2017"
      },
      "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
    }
  ]
}

```

```

    }

    ],
    "resourceRelationship": [{
      "type": "contains",
      "resource": {
        "id": "44",
        "href": " http://server:port/resourceInventoryManagement/resource/44"
      }
    }
  ]},
  "relatedParty": [{
    "role": "customer",
    "id": "42",
    "href": "http://serverlocation:port/customerManagement/individual/42"
  }
]}
}

```

CREATE RESOURCE

POST /resource

Description

This operation creates a resource entity. E.,g Equipment, ResourceFunction, MSISDN, IPAddress, TopologicalLink

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a Resource, including any possible rule conditions and applicable default values.

Mandatory Attributes	Rule
name	
@type	

Non Mandatory Attributes	Default Value	Rule
description		
version		
@baseType		
@schemaLocation		
validFor		
lifecycleState		
resourceRelationship		
place		
note		
resourceAttachment		

resourceSpecification		
relatedParty		
resourceCharacteristic		
resourceRelationshipCharacteristic		

Additional Rules

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating a Resource resource.

Context	Mandatory Sub-Attributes
resourceSpecification	Id or href
relatedParty	role, id or href
resourceRelationship	type, Id OR href
note	text
place	role, id or href
resourceAttachment	Id or href

Usage Samples

Here's an example of a request for creating a resource of type Equipment. In this example the request only passes mandatory attributes.

Create a Resource of type Equipment in the inventory

Request
POST /resourceInventoryManagement/resource Content-Type: application/json <pre>{ "name": "1-MAIN", "@type": "Equipment", "@baseType": "PhysicalResource", "@schemaLocation": "http://server:port/resourceInventoryManagement/schema/Equipment.yml", "resourceSpecification": { "id": "6", "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6" }, "resourceCharacteristic": { "name": "Equipment", "value": { "@type": "Equipment", "serialNumber": "12444545544", "versionNumber": "1.22", "manufactureDate": "05-04-2017" }, "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml" } }</pre>

<pre> }} }</pre>
Response
201 "{ JSON Resource Representation with every provided and default attribute}"

PATCH RESOURCE

PATCH /resource/{id}

Description

This operation allows partial updates of a resource entity. Support of json/merge (<https://tools.ietf.org/html/rfc7386>) is mandatory, support of json/patch (<http://tools.ietf.org/html/rfc5789>) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

Patchable Attributes	Rule
description	
validFor	
name	
lifecycleState	
resourceRelationship	
place	
note	
resourceSpecification	
relatedParty	
resourceCharacteristic	
@schemaLocation	

Non Patchable Attributes	Rule
id	
href	

@type	
@baseType	

Usage Samples

Here's an example of a request for patching a Resource resource.

Changing the resource state (using json-patch)

Request
PATCH /resourceInventoryManagement/resource/42 Content-Type: application/json-patch+json <pre>{ "path": "/lifecycleState", "value": "Active", "op": "replace" }</pre>
Response
200 { Similar JSON as in GET response with state changed }

PUT RESOURCE

This Uniform Contract operation is used to completely replace a RESOURCE.

Description :

- Used to replace one RESOURCE with another.

Behavior :

- Returns HTTP/1.1 status code 200 if the request was successful.

Request
PUT /resourceInventoryManagement/resource/42 Accept: application/json <pre>{ "id": "42",</pre>

```

    "href": "http://server:port/resourceInventoryManagement/resource/42",
    "@type": "EQUIPMENT",
    "name": "1 - MAIN",
    "lifecycleState": "Active",
    "resourceSpecification": {
      "id": "4",
      "href": " http://server:port/resourceCatalogManagement/resourceSpecification/4",
      "@type": "EquipmentSpecification"
    },
    "resourceCharacteristic": [
      {
        "name": "Equipment",
        "value": {
          "@type": "Equipment",
          "serialNumber": "12444545544",
          "versionNumber": "1.22",
          "manufactureDate": "05-04-2017"
        },
        "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
      }
    ],
    "resourceRelationship": [{
      "type": "contains",
      "resource": {
        "id": "44",
        "href": " http://server:port/resourceInventoryManagement/resource/44"
      }
    }],
    "relatedParty": [{
      "role": "customer",
      "id": "42",
      "href": "http://serverlocation:port/customerManagement/individual/42"
    }
  ]
}

```

Response

201

Content-Type: application/json

Location: <http://server:port/resourceInventoryManagement/resource/42>

```

{
  "id": "42",
  "href": "http://server:port/resourceInventoryManagement/resource/42",
  "@type": "EQUIPMENT",
  "name": "1 - MAIN",
  "lifecycleState": "Active",
  "resourceSpecification": {

```

```

    "id": "4",
    "href": " http://server:port/resourceCatalogManagement/resourceSpecification/4",
    "@type": "EquipmentSpecification"
  },
  "resourceCharacteristic": [
    {
      "name": "Equipment",
      "value": {
        "@type": "Equipment",
        "serialNumber": "12444545544",
        "versionNumber": "1.22",
        "manufactureDate": "05-04-2017"
      },
      "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
    }
  ],
  "resourceRelationship": [{
    "type": "contains",
    "resource": {
      "id": "44",
      "href": " http://server:port/resourceInventoryManagement/resource/44"
    }
  }],
  "relatedParty": [{
    "role": "customer",
    "id": "42",
    "href": "http://serverlocation:port/customerManagement/individual/42"
  }
]}

```

DELETE RESOURCE

DELETE /resource/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a resource entity.

Usage Samples

Here's an example of a request for deleting a resource.

Request
DELETE /resourceInventoryManagement/resource/42
Response
204

OPERATIONS ON LOGICAL RESOURCE

LIST LOGICAL RESOURCES

GET /logicalResource?fields=...&{filtering}

Description

This operation list logical resource entities.

Attribute selection is enabled for all first level attributes.

Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving logical resource instances.

Get all available resources of type "MSISDN" of category 'Regular'.

Request
GET /resourceInventoryManagement/logicalResource?@type="MSISDN" & lifecycleStatus ="Available" & category ="Regular" & relatedParty.id =42 Accept: application/json
Response
200 [["id": "42", "href": "http://server:port/resourceInventoryManagement/logicalResource/42",

```

"category": "Regular",
"@type": "MSISDN",
"@basetype": "LogicalResource",
"value": "07376456789",
"lifecycleState": "Available",
"resourceSpecification": {
  "id": "4",
  "href": " http://server:port/resourceCatalogManagement/resourceSpecification/4"
},
"relatedParty": [{
  "role": "owner",
  "id": "42",
  "href": "http://serverlocation:port/customerManagement/individual/42"
}]
}
}

```

RETRIEVE LOGICAL RESOURCE

GET /logicalResource/{id}?fields=...&{filtering}

Description

This operation retrieves a logical resource entity.

Attribute selection is enabled for all first level attributes.

Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a single Logical Resource from the Inventory providing as input resource id.

Request
GET /resourceInventoryManagement/logicalResource/42 Accept: application/json
Response
200 { "id": "42", "href": "http://server:port/resourceInventoryManagement/logicalResource/42", "category": "Regular", "@type": "MSISDN",

```

    "@basetype": "LogicalResource",
    "value": "07376456789",
    "lifecycleState": "Active",
    "resourceSpecification": {
      "id": "4",
      "href": " http://server:port/resourceCatalogManagement/resourceSpecification/4"
    },
    "relatedParty": [{
      "role": "owner",
      "id": "42",
      "href": "http://serverlocation:port/customerManagement/individual/42"
    }]
  }

```

CREATE LOGICAL RESOURCE

POST /logicalResource

Note: this operation is available only to ADMIN API users

Description

This operation creates a Logical resource entity. E.g MSISDN, IPAddress, TopologicalLink

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a logical Resource, including any possible rule conditions and applicable default values.

Mandatory Attributes	Rule
name	
@type	
value	

Non Mandatory Attributes	Default Value	Rule
category		
description		
version		
@baseType		
@schemaLocation		
validFor		
lifecycleState		
resourceRelationship		
Place		
Note		
resourceSpecification		
relatedParty		
resourceCharacteristic		
resourceAttachment		

Additional Rules

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating a Resource resource.

Context	Mandatory Sub-Attributes
resourceSpecification	id, href
relatedParty	role, id or href
resourceRelationship	type, Id OR href
note	text
place	role, id or href
resourceAttachment	Id or href

Usage Samples

Here's an example of a request for creating a resource of type MSISDN. In this example the request only passes mandatory attributes.

Create a Resource of type MSISDN in the inventory

Request
POST /resourceInventoryManagement/logicalResource Content-Type: application/json <pre>{ "value": "07865443255 ", "@type": "MSISDN", }</pre>
Response
201 "{ JSON Resource Representation with every provided and default attribute}"

PATCH LOGICAL RESOURCE

PATCH /logicalResource/{id}

Description

This operation allows partial updates of a logical resource entity. Support of json/merge (<https://tools.ietf.org/html/rfc7386>) is mandatory, support of json/patch (<http://tools.ietf.org/html/rfc5789>) is

optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

Patchable Attributes	Rule
category	
description	
endDate	
name	
value	
startDate	
lifecycleState	
resourceRelationship	
place	
note	
resourceSpecification	
relatedParty	
characteristic	
@schemaLocation	
resourceAttachment	

Non Patchable Attributes	Rule
id	
href	
@type	
@baseType	

Usage Samples

Here's an example of a request for patching a LogicalResource resource.

Changing the resource state (using json-patch)

Request
PATCH /resourceInventoryManagement/logicalResource/42 Content-Type: application/json-patch+json <pre>{ "path": "/lifecycleState",</pre>

<pre>"value": "Reserved", "op": "replace" }</pre>
<p>Response</p>
<p>200</p> <p>{ Similar JSON as in GET response with state changed }</p>

PUT LOGICAL RESOURCE

This Uniform Contract operation is used to completely replace a Logical resource.

Description :

- Used to replace one Logical Resource with another.

Behavior :

- Returns HTTP/1.1 status code 200 if the request was successful.

<p>Request</p>
<pre>PUT /resourceInventoryManagement/logicalResource/42 Accept: application/json { "id": "42", "href": "http://server:port/resourceInventoryManagement/logicalResource/42", "@type": "MSISDN", "@basetype": "LogicalResource", "category": "Regular", "value": "07376456789", "lifecycleState": "Active", "resourceSpecification": { "id": "4", "href": " http://server:port/resourceCatalogManagement/resourceSpecification/4" }, "relatedParty": [{ "role": "owner", "id": "42", "href": "http://serverlocation:port/customerManagement/individual/42" }] }</pre>

Response

```

200
Content-Type: application/json
Location: http://server:port/resourceInventoryManagement/logicalResource/42

{
  "id": "42",
  "href": "http://server:port/resourceInventoryManagement/logicalResource/42",
  "@type": "MSISDN",
  "@basetype": "LogicalResource",
  "category": "Regular",
  "value": "07376456789",
  "lifecycleState": "Active",
  "resourceSpecification": {
    "id": "4",
    "href": "http://server:port/resourceCatalogManagement/resourceSpecification/4"
  },
  "relatedParty": [{
    "role": "owner",
    "id": "42",
    "href": "http://serverlocation:port/customerManagement/individual/42"
  }]
}

```

DELETE LOGICAL RESOURCE

DELETE /logicalResource/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a logical resource entity.

Usage Samples

Here's an example of a request for deleting a logical resource.

Request

```
DELETE /resourceInventoryManagement/logicalResource/42
```

Response

204

OPERATIONS ON PHYSICAL RESOURCE

LIST PHYSICAL RESOURCES

GET /physicalResource?fields=...&{filtering}

Description

This operation list physical resource entities.

Attribute selection is enabled for all first level attributes.

Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving physical resource instances.

Get all active resources of type "EQUIPMENT" and manufactured by a party

Request

GET /resourceInventoryManagement/physicalResource?@type=Equipment&lifecycleState=Active & relatedParty.id =43
Accept: application/json

Response

200

```
[{
  "id": "45",
  "href": "http://server:port/resourceInventoryManagement/physicalResource/45",
  "@type": "Equipment",
  "@baseType": "PhysicalResource",
  "lifecycleState": "Active",
  "manufactureDate": "2007-04-12",
  "serialNumber": "123456745644",
  "versionNumber": "11",
  "resourceSpecification": {
    "id": "6",
```

```

        "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6",
        "@type": "PhysicalResourceSpecification"
    },
    "resourceCharacteristic": {
        "name": "Equipment",
        "value": {
            "@type": "Equipment",
            "serialNumber": "12444545544",
            "versionNumber": "1.22",
            "manufactureDate": "05-04-2017"
        },
        "@schemaLocation": "http:
//serverlocation:port/partyManagement/schema/Equipment.yml"
    },
    },
    "resourceRelationship": {
        "type": "requires",
        "resource": {
            "id": "46",
            "href": " http://server:port/resourceInventoryManagement/logicalResource/46"
        }
    },
    },
    "relatedParty": {
        "role": "Manufacturer",
        "id": "43",
        "href": "http://serverlocation:port/PartyManagement/individual/43"
    }
}
]

```

RETRIEVE PHYSICAL RESOURCE

GET /physicalResource/{id}?fields=...&{filtering}

Description

This operation retrieves a resource entity.

Attribute selection is enabled for all first level attributes.

Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a single Resource from the Inventory providing as input resource id.

Request

GET /resourceInventoryManagement/physicalResource/45
 Accept: application/json

Response

```
{
  "id": "45",
  "href": "http://server:port/resourceInventoryManagement/physicalResource/45",
  "@type": "Equipment",
  "@baseType": "PhysicalResource",
  "lifecycleState": "Active",
  "manufactureDate": "2007-04-12",
  "serialNumber": "123456745644",
  "versionNumber": "11",
  "resourceSpecification": {
    "id": "6",
    "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6",
    "@type": "PhysicalResourceSpecification"
  },
  "resourceCharacteristic": [
    {
      "name": "Equipment",
      "value": {
        "@type": "Equipment",
        "serialNumber": "12444545544",
        "versionNumber": "1.22",
        "manufactureDate": "05-04-2017"
      },
      "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
    }
  ],
  "resourceRelationship": [{
    "type": "requires",
    "resource": {
      "id": "46",
      "href": " http://server:port/resourceInventoryManagement/logicalResource/46"
    }
  }
  ],
  "relatedParty": [{
    "role": "Manufacturer",
    "id": "43",
    "href": "http://serverlocation:port/PartyManagement/individual/43"
  }
  ]
}
```

CREATE PHYSICAL RESOURCE

POST /resource

Note: this operation is available only to ADMIN API users

Description

This operation creates a physical resource entity. E.,g Equipment, ResourceFunction, MSISDN, IPAddress, TopologicalLink

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a Resource, including any possible rule conditions and applicable default values.

Mandatory Attributes	Rule
name	
@type	
serialNumber	

Non Mandatory Attributes	Default Value	Rule
Category		
Description		
version		
manufactureDate		
versionNumber		
powerState		
@baseType		
@schemaLocation		
validFor		
lifecycleState		
resourceRelationship		
Place		
Note		
resourceSpecification		
relatedParty		
resourceCharacteristic		
resourceAttachment		

Additional Rules

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating a Resource resource.

Context	Mandatory Sub-Attributes
resourceSpecification	id, href
relatedParty	role, id or href
resourceRelationship	type, Id OR href

note	text
place	role, id or href
resourceAttachment	Id or href

Usage Samples

Here's an example of a request for creating a resource of type Equipment. In this example the request only passes mandatory attributes.

Create a Resource of type Equipment in the inventory

Request
<pre>POST /resourceInventoryManagement/physicalResource Content-Type: application/json { "@type": "Equipment", "manufactureDate": "2007-04-12", "serialNumber": "123456745644", "resourceSpecification": { "id": "6", "href": "http://server:port/resourceCatalogManagement/resourceSpecification/6", "@type": "PhysicalResourceSpecification" } }</pre>
Response
<pre>201 "{ JSON Resource Representation with every provided and default attribute}"</pre>

PATCH PHYSICAL RESOURCE

PATCH /physicalResource/{id}

Description

This operation allows partial updates of a physical resource entity. Support of json/merge (<https://tools.ietf.org/html/rfc7386>) is mandatory, support of json/patch (<http://tools.ietf.org/html/rfc5789>) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules

concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

Patchable Attributes	Rule
category	
description	
validFor	
name	
@schemaLocation	
manufactureDate	
powerState	
serialNumber	
versionNumber	
lifecycleState	
resourceRelationship	
place	
note	
resourceSpecification	
relatedParty	
resourceCharacteristic	
resourceAttachment	

Non Patchable Attributes	Rule
id	
href	
@type	
@baseType	

Usage Samples

Here's an example of a request for patching a physical resource.

Changing the resource state (using json-patch)

Request
PATCH /resourceInventoryManagement/physicalResource/45 Content-Type: application/json-patch+json <pre>{ "path": "/versionNumber", "value": "11", "op": "replace" }</pre>

}
Response
200 { Similar JSON as in GET response with version number changed}

PUT PHYSICAL RESOURCE

This Uniform Contract operation is used to completely replace a Physical resource.

Description :

- Used to replace one Physical Resource with another.

Behavior :

- Returns HTTP/1.1 status code 200 if the request was successful.

Request
<pre> PUT /resourceInventoryManagement/physicalResource/45 Accept: application/json { "id": "45", "href": "http://server:port/resourceInventoryManagement/physicalResource/45", "@type": "Equipment", "@baseType": "PhysicalResource", "lifecycleState": "Active", "manufactureDate": "2007-04-12", "serialNumber": "123456745644", "versionNumber": "11", "resourceSpecification": { "id": "6", "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6", "@type": "PhysicalResourceSpecification" }, "resourceCharacteristic": [{ "name": "Equipment", "value": { "@type": "Equipment", "serialNumber": "12444545544", "versionNumber": "1.22", "manufactureDate": "05-04-2017" } } } </pre>

```

        "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
    }
},
"resourceRelationship": [{
    "type": "requires",
    "resource": {
        "id": "46",
        "href": " http://server:port/resourceInventoryManagement/logicalResource/46"
    }
}],
"relatedParty": [{
    "role": "Manufacturer",
    "id": "43",
    "href": "http://serverlocation:port/PartyManagement/individual/43"
}]
}

```

Response

200

Content-Type: application/json

Location: <http://server:port/resourceInventoryManagement/physicalResource/45>

```

{
    "id": "45",
    "href": "http://server:port/resourceInventoryManagement/physicalResource/45",
    "@type": "Equipment",
    "@baseType": "PhysicalResource",
    "lifecycleState": "Active",
    "manufactureDate": "2007-04-12",
    "serialNumber": "123456745644",
    "versionNumber": "11",
    "resourceSpecification": {
        "id": "6",
        "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6",
        "@type": "PhysicalResourceSpecification"
    },
    "resourceCharacteristic": [{
        "name": "Equipment",
        "value": {
            "@type": "Equipment",
            "serialNumber": "12444545544",
            "versionNumber": "1.22",
            "manufactureDate": "05-04-2017"
        }
    },
    "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
}
},
"resourceRelationship": [{
    "type": "requires",
    "resource": {
        "id": "46",
        "href": " http://server:port/resourceInventoryManagement/logicalResource/46"
    }
}

```

```
    }  
  },  
  "relatedParty": [{  
    "role": "Manufacturer",  
    "id": "43",  
    "href": "http://serverlocation:port/PartyManagement/individual/43"  
  }  
}
```

DELETE PHYSICAL RESOURCE

DELETE /physicalResource/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a physical resource entity.

Usage Samples

Here's an example of a request for deleting a physical resource.

Request
DELETE /resourceInventoryManagement/physicalResource/45
Response
204

OPERATIONS ON {RESOURCE-TYPE}

LIST {RESOURCE-TYPE}

Description

This operation list of {resource-type} entities. A resource type can be an MSISDN, Equipment, IP address etc.....

Attribute selection is enabled for all first level attributes.

Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving {resource-type} instances.

Get all resources of type equipment

Request
GET /resourceInventoryManagement/equipment?fields=id, href, name Accept: application/json
Response
200 [{ "id": "42", "href": "http://server:port/resourceInventoryManagement/equipment/42", "name": "equipment -1" }, { "id": "43", "href": "http://server:port/resourceInventoryManagement/equipment/43", "name": "equipment -3" }]

RETRIEVE {RESOURCE-TYPE}

GET /{resource-type}/{id}?fields=...&{filtering}

Description

This operation retrieves a resource entity.

Attribute selection is enabled for all first level attributes.

Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a single Resource of type equipment from the Inventory providing as input resource id.

Request
GET /resourceInventoryManagement/equipment/45?fields=id,href,name &lifecycleStatus="Active" Accept: application/json
Response
{ "id": "42", "href": "http://server:port/resourceInventoryManagement/equipment/42", "name": "equipment -1" }

CREATE {RESOURCE-TYPE}

POST /{resource-type}

Note: this operation is available only to ADMIN API users

Description

This operation creates a {resource-type} entity. E.g Equipment, ResourceFunction, MSISDN, IPAddress, TopologicalLink

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a Resource, including any possible rule conditions and applicable default values.

Mandatory Attributes	Rule
name	
@type	

Non Mandatory Attributes	Default Value	Rule
description		
publicIdentifier		
version		
manufactureDate		
versionNumber		
powerState		
@baseType		
@schemaLocation		
validFor		
lifecycleState		
resourceRelationship		
Place		
Note		
resourceSpecification		
relatedParty		
resourceCharacteristic		
resourceAttachment		

Additional Rules

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating a Resource resource.

Context	Mandatory Sub-Attributes
resourceSpecification	id, href
relatedParty	role, id or href
resourceRelationship	type, Id OR href
note	text
place	role, id or href
resourceAttachment	Id or href

Usage Samples

Here's an example of a request for creating a resource of type Equipment. In this example the request only passes mandatory attributes.

Create a Resource of type Equipment in the inventory

Request

```

POST /resourceInventoryManagement/equipment
Content-Type: application/json
{
    "@type": "Equipment",
    "manufactureDate": "2007-04-12",
    "serialNumber": "123456745644",
    "resourceSpecification": {
        "id": "6",
        "href": " http://server:port/resourceCatalogManagement/resourceSpecification/6",
        "@type": "PhysicalResourceSpecification"
    }
}

```

Response

201

```
"{ JSON Resource Representation with every provided and default attribute}"
```

PATCH {RESOURCE-TYPE}

PATCH /{resource-type}/{id}

Description

This operation allows partial updates of a {resource-type} entity. Support of json/merge (<https://tools.ietf.org/html/rfc7386>) is mandatory, support of json/patch (<http://tools.ietf.org/html/rfc5789>) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

Patchable Attributes	Rule
description	
validFor	
name	
value	
lifecycleState	

resourceRelationship	
place	
note	
resourceSpecification	
relatedParty	
characteristic	
@schemaLocation	

Non Patchable Attributes	Rule
id	
href	
@type	
@baseType	

Usage Samples

Here's an example of a request for patching a {resource-type} resource.

Changing the resource state (using json-patch) for an equipment

Request
PATCH /resourceInventoryManagement/equipment/42 Content-Type: application/json-patch+json { "path": "/lifecycleState", "value": "Reserved", "op": "replace" }
Response
200 { Similar JSON as in GET response with state changed }

Here's an example of a request for patching a {resource-type} resource.

Changing the resource state (using json-patch) providing as input publicIdentifier

Request
PATCH /resourceInventoryManagement/equipment/07467223333 Content-Type: application/json-patch+json

```
{
  "path": "/lifecycleState",
  "value": "Reserved",
  "op": "replace"
}
```

Response

201

{ Similar JSON as in GET response with state changed }

PUT {RESOURCE-TYPE}

This Uniform Contract operation is used to completely replace a {resource –type}

Description :

- Used to replace one {resource –type} with another.

Behavior :

- Returns HTTP/1.1 status code 200 if the request was successful.
Completely update an equipment

Request

PUT /resourceInventoryManagement/equipment/45
Accept: application/json

```
{
  "id": "45",
  "href": "http://server:port/resourceInventoryManagement/equipment/45",
  "@type": "Equipment",
  "@baseType": "PhysicalResource",
  "lifecycleState": "Active",
  "manufactureDate": "2007-04-12",
  "serialNumber": "123456745644",
  "versionNumber": "11",
  "resourceSpecification": {
    "id": "6",
    "href": "http://server:port/resourceCatalogManagement/resourceSpecification/6",
    "@type": "PhysicalResourceSpecification"
  }
},
```

```

"resourceCharacteristic": [{
  "name": "Equipment",
  "value": {
    "@type": "Equipment",
    "serialNumber": "12444545544",
    "versionNumber": "1.22",
    "manufactureDate": "05-04-2017"
  },
  "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
}],
"resourceRelationship": [{
  "type": "requires",
  "resource": {
    "id": "46",
    "href": "http://server:port/resourceInventoryManagement/logicalResource/46"
  }
}],
"relatedParty": [{
  "role": "Manufacturer",
  "id": "43",
  "href": "http://serverlocation:port/PartyManagement/individual/43"
}]
}

```

Response

200

Content-Type: application/json

Location: <http://server:port/resourceInventoryManagement/equipment/45>

```

{
  "id": "45",
  "href": "http://server:port/resourceInventoryManagement/equipment/45",
  "@type": "Equipment",
  "@baseType": "PhysicalResource",
  "lifecycleState": "Active",
  "manufactureDate": "2007-04-12",
  "serialNumber": "123456745644",
  "versionNumber": "11",
  "resourceSpecification": {
    "id": "6",
    "href": "http://server:port/resourceCatalogManagement/resourceSpecification/6",
    "@type": "PhysicalResourceSpecification"
  },
  "resourceCharacteristic": [{
    "name": "Equipment",
    "value": {
      "@type": "Equipment",
      "serialNumber": "12444545544",
      "versionNumber": "1.22",
      "manufactureDate": "05-04-2017"
    },
    "@schemaLocation": "http://serverlocation:port/partyManagement/schema/Equipment.yml"
  }
}

```

```

    }},
    "resourceRelationship": [{
      "type": "requires",
      "resource": {
        "id": "46",
        "href": "http://server:port/resourceInventoryManagement/logicalResource/46"
      }
    }],
    "relatedParty": [{
      "role": "Manufacturer",
      "id": "43",
      "href": "http://serverlocation:port/PartyManagement/individual/43"
    }
  ]
}

```

DELETE {RESOURCE-TYPE}

DELETE /{resource-type}/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a {resource-type} entity.

Usage Samples

Here's an example of a request for deleting a resource of type equipment.

Request
DELETE /resourceInventoryManagement/equipment/42
Response
204

API NOTIFICATIONS

For every single of operation on the entities use the following templates and provide sample REST notification POST calls.

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

REGISTER LISTENER

POST /hub

Description

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior

Returns HTTP/1.1 status code 201 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

Usage Samples

Here's an example of a request for registering a listener.

Request
POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}
Response
201 Content-Type: application/json Location: /api/hub/42 {"id": "42", "callback": "http://in.listener.com", "query": null}

UNREGISTER LISTENER

DELETE /hub/{id}

Description

Clears the communication endpoint address that was set by creating the Hub..

Behavior

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

Usage Samples

Here's an example of a request for un-registering a listener.

Request
DELETE /api/hub/42 Accept: application/json
Response
204

PUBLISH EVENT TO LISTENER

POST /client/listener

Description

Clears the communication endpoint address that was set by creating the Hub.

Provides to a registered listener the description of the event that was raised. The /client/listener url is the callback url passed when registering the listener.

Behavior

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

Usage Samples

Here's an example of a notification received by the listener. In this example "EVENT TYPE" should be replaced by one of the notification types supported by this API (see Notification resources Models section) and EVENT BODY refers to the data structure of the given notification type.

Request
POST /client/listener Accept: application/json { "event": { EVENT BODY }, "eventType": "EVENT_TYPE" }
Response
201

For detailed examples on the general TM Forum notification mechanism, see the TMF REST Design Guidelines.

ACKNOWLEDGEMENTS**RELEASE HISTORY**

Release Number	Date	Release led by:	Description
Release 17.0.0	20-Sep-2017	Pierre Gauthier TM Forum pgauthier@tmforum.org Nicoleta Stoica Vodafone Nicoleta.stoica@vodafone.com	
Release 17.0.1 Version 1.0.1	04-Dec-2017	Adrienne Walcott TM Forum awalcott@tmforum.org	Updated to reflect TM Forum Approved Status