



Frameworkx Specification

Service Inventory Management API REST Specification

TMF638
Release 16.5.1
April 2017

Latest Update: Frameworkx Release 16.5	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:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 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	6
Introduction	7
SAMPLE USE CASES	8
Service inventory query for a customer.....	8
Service inventory update as part of service provisioning	8
RESOURCE MODEL.....	9
Managed Entity and Task Resource Models	9
Service resource.....	9
Notification Resource Models	14
Service Creation Notification	15
Service Attribute Value Change Notification	15
Service State Change Notification	16
Service Batch Notification	16
Service Remove Notification.....	17
API OPERATIONS.....	18
Operations on Service	18
List services.....	19
Retrieve service.....	20
Create service	21
Patch service.....	23
Delete service.....	25
API NOTIFICATIONS.....	26
Register listener	26
Unregister listener	26

Publish Event to listener	27
Acknowledgements.....	29
Release History	29

LIST OF TABLES

N/A

INTRODUCTION

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

SAMPLE USE CASES

Service inventory query for a customer

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

Note: Only the CustomerFacingServices instances will be presented to the customer.

Service inventory update as part of service provisioning

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

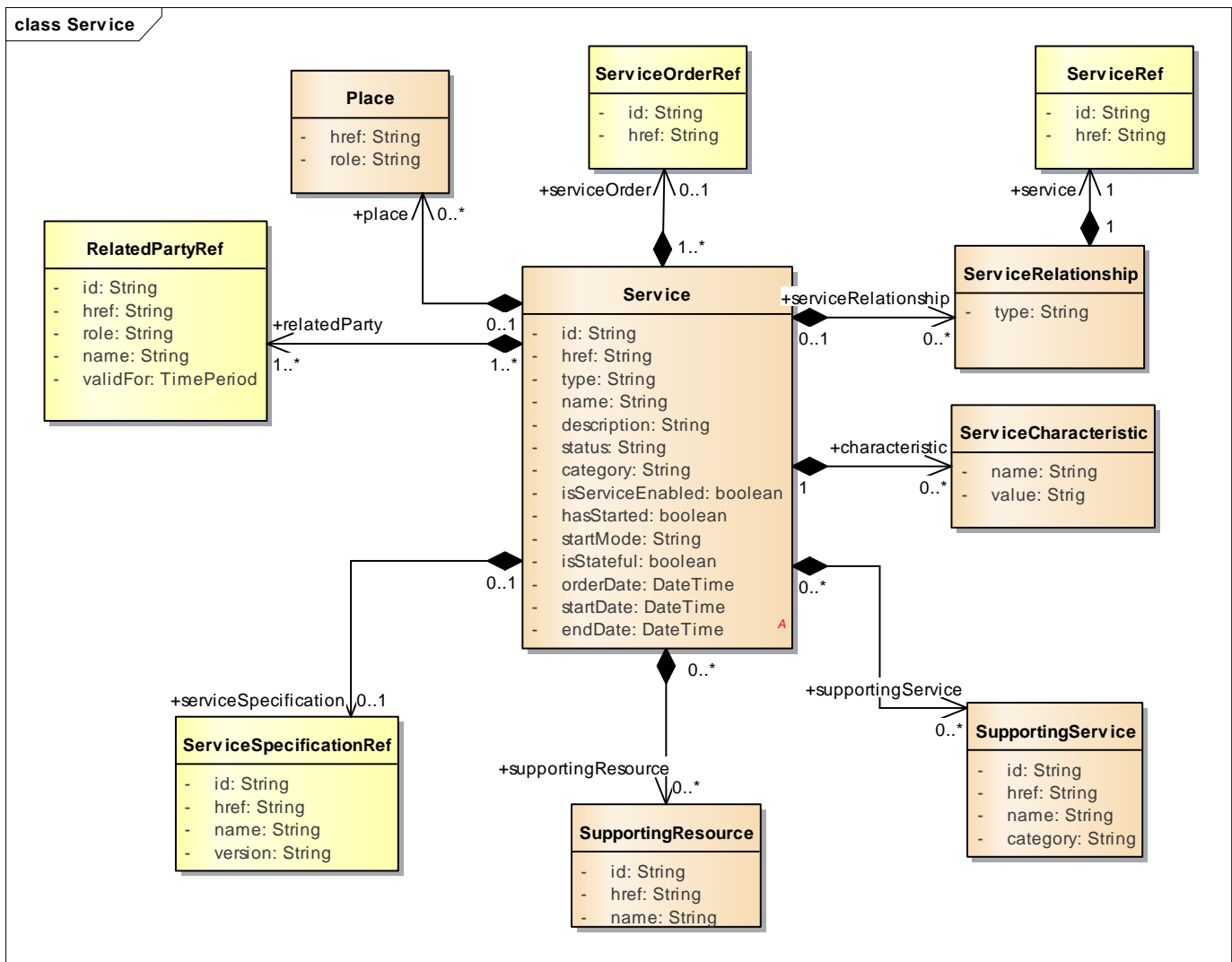
RESOURCE MODEL

Managed Entity and Task Resource Models

SERVICE RESOURCE

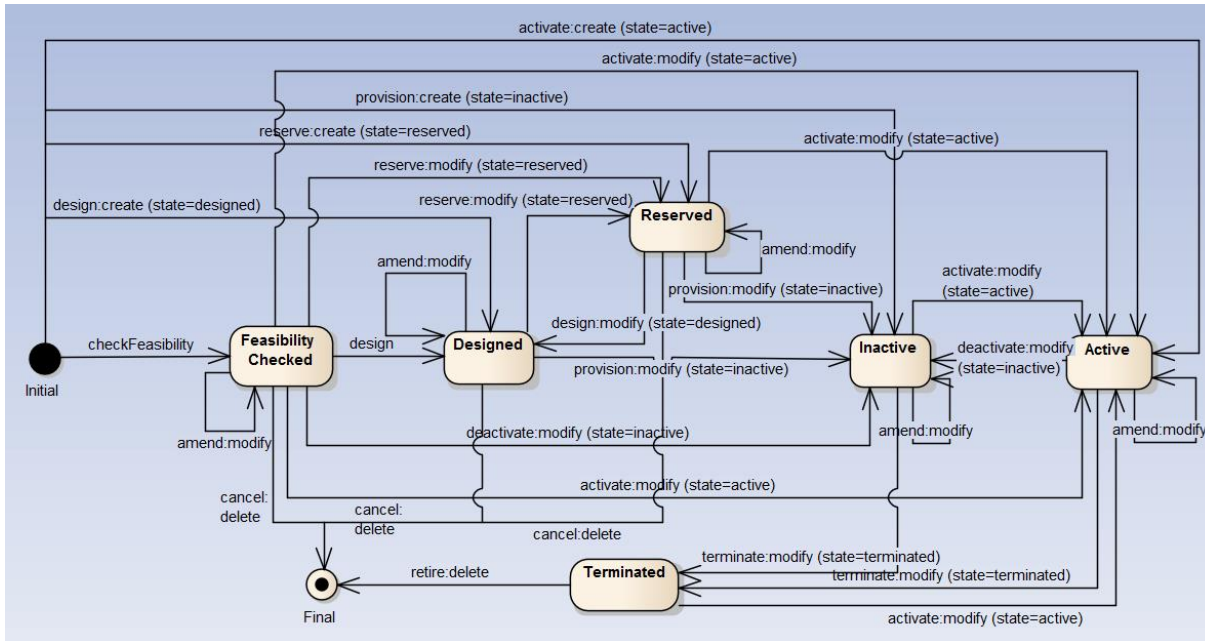
Service is an abstract base class for defining the Service hierarchy. All Services are characterized as either being possibly visible and usable by a Customer or not. This gives rise to the two subclasses of Service: CustomerFacingService and ResourceFacingService.

Resource model



Lifecycle

Here is the state machine diagram for a Service.



Field descriptions

Service fields

category	A string. Is it a customer facing or resource facing service.
description	A string. free-text description of the service.
endDate	A date time (DateTime). endDate is the date when the service ends.
hasStarted	A boolean. This is a Boolean attribute that, if TRUE, signifies that this Service has already been started. If the value of this attribute is FALSE, then this signifies that this Service has NOT been Started.
href	A string. Reference of the service.
id	A string. "id" is the ID created for the service.
isServiceEnabled	A boolean. For use. If the value of this attribute is FALSE, then this means that this particular Service has NOT been enabled for use.
isStateful	A boolean. This is a Boolean attribute that, if TRUE, means that this Service can be changed without affecting any other services.
name	A string. "name" is the name of the service.
orderDate	A date time (DateTime). "orderDate" is the date when the service was ordered.
startDate	A date time (DateTime). stardDate is the date when the service starts.

startMode	A string. This attribute is an enumerated integer that indicates how the Service is started. Values include: 0: Unknown 1: Automatically by the managed environment 2: Automatically by the owning device 3: Manually by the Provider of the Service 4: Manually by a Customer of the Provider 5: Any of the above.
status	A string. The status of the service: - feasibilityChecked - designed - reserved - active - inactive - terminated
type	A string. Name of the resource type.
serviceOrder	A service order reference (ServiceOrderRef). A Service Order is a request to perform an action on a specific Service and its contained services.
supportingResource	A list of supporting resources (SupportingResource [*]).
serviceRelationship	A list of service relationships (ServiceRelationship [*]). Describes links with services of the same category (useful for bundled services).
place	A list of places (Place [*]). Used to define a place useful for the service (for example a delivery geographical place).
supportingService	A list of supporting services (SupportingService [*]). A collection of services that support this service (links between CFS; RFS).
serviceSpecification	A service specification reference (ServiceSpecificationRef). ServiceSpecification(s) required to realize a ProductSpecification.
relatedParty	A list of related party references (RelatedPartyRef [1..*]). A related party defines party or party role linked to a specific entity.
characteristic	A list of service characteristics (ServiceCharacteristic [*]). is a list of name value pairs that define the service characteristics.

Place sub-resource

Used to define a place useful for the service (for example a delivery geographical place).

href A string. Reference of a place (for instance in google map).

role A string. Role of the place (for instance delivery geographical place).

ServiceCharacteristic sub-resource

is a list of name value pairs that define the service characteristics.

name A string. Name of the characteristic.

value A string (String). Value of the characteristic.

ServiceRelationship sub-resource

Describes links with services of the same category (useful for bundled services).

type A string. Describes links with services of the same category (useful for bundled services).

serviceRef A service reference (ServiceRef). Useful to link services of the same category.

SupportingResource sub-resource

href A string. Reference of the supporting resource.

id A string. Unique identifier of the supporting resource.

name A string. Name of the resource supporting the service.

SupportingService sub-resource

A collection of services that support this service (links between CFS; RFS).

category A string. Category of the supporting service.

href A string. Reference of the supporting service.

id A string. Unique identifier of the supporting service.

name A string. Name of the supporting service.

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.

ServiceOrderRef relationship

A Service Order is a request to perform an action on a specific Service and its contained services.

href A string. The Hyperlink to access the related Service Order.

id A string. Unique identifier of the related Service Order.

ServiceRef relationship

Service reference. Useful to link services of the same category.

href A string. reference of the service.

id A string. Id of the service.

ServiceSpecificationRef relationship

Service specification reference: ServiceSpecification(s) required to realize a ProductSpecification.

href A string. Reference of the ServiceSpecification.

id A string. Unique identifier of the service specification.

name A string. Name of the required ServiceSpecification.

version A string. Service specification version.

Json representation sample

We provide below the json representation of an example of a 'Service' resource object

```
{
  "id": "42",
  "href": "http://server:port/serviceInventory/service/42",
  "category": "CFS",
  "name": "Broadband",
  "description": "Description of the Broadband service",
  "isServiceEnabled": true,
  "hasStarted": true,
  "startMode": 0,
  "isStateful": false,
  "status": "active",
  "serviceSpecification": {
    "id": "4",
    "href": "http://server:port/serviceCatalogManagement/serviceSpecification/4"
  },
  "serviceCharacteristic": [
    {
      "name": "speed",
      "value": "16M"
    }
  ],
  "serviceRelationship": [
```

```

    {
      "type": "contains",
      "service": {
        "id": "44",
        "href": " http://server:port/serviceInventoryManagement/service/44"
      }
    }
  ],
  "supportingService": [
    {
      "id": "59",
      "href": " http://server:port/ serviceInventoryManagement /service/59"
    }
  ],
  "supportingResource": [
    {
      "id": "46779",
      "href": "http://server:port/resourceInventoryManagement/logicalResource/46779"
    }
  ],
  "relatedParty": [
    {
      "role": "partner",
      "id": "42",
      "href": "http://serverlocation:port/partyManagement/organisation/42"
    }
  ]
}

```

Notification Resource Models

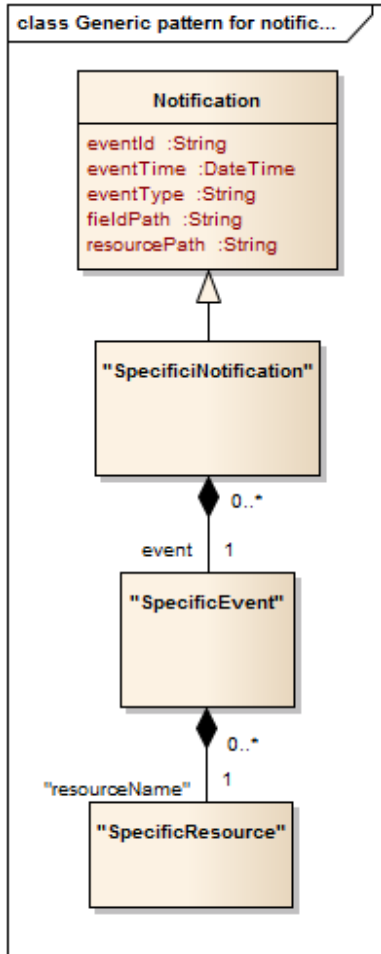
5 notifications are defined for this API

Notifications related to Service:

- ServiceCreationNotification
- ServiceAttributeValueChangeNotification
- ServiceStateChangeNotification
- ServiceBatchNotification
- ServiceRemoveNotification

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).



SERVICE CREATION NOTIFICATION

Notification sent when a new Service resource is created.

Json representation sample

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

```

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

```

SERVICE ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when changing an attribute of a Service resource.

Json representation sample

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

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

SERVICE STATE CHANGE NOTIFICATION

Notification sent when changing the status of a Service resource.

Json representation sample

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

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

SERVICE BATCH NOTIFICATION

Notification sent when a batch job on resource Service changes

Json representation sample

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

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

SERVICE REMOVE NOTIFICATION

Notification sent when removing a Service resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceRemoveNotification",
  "event": {
    "service" :
      {-- SEE Service 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 SERVICE

LIST SERVICES

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

Description

This operation list service 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 Service resources.

Get all RFS supported by a specific resource

Request
GET /serviceInventoryManagement/service?fields=category,name,status&category=RFS&supportingResources.id=33 Accept: application/json
Response
200 <pre>[{ "category": "RFS", "status": "active", "href": " http://server:port/ serviceInventoryManagement /service/42", "id": "42", "name": "My service 1" }, { "category": "RFS", "status": "active", "href": "http://server:port/ serviceInventoryManagement /service/43", "id": "43", "name": "My service 2" }, { "category": "RFS", "status": "active", "href": "http://server:port/ serviceInventoryManagement /service/44", "id": "44", "name": "My service 3" }]</pre>

RETRIEVE SERVICE

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

Description

This operation retrieves a service 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 Service resource.

Request
GET /serviceInventoryManagement/service/42 Accept: application/json
Response
200 <pre>{ "id": "42", "href": "http://server:port/ serviceInventoryManagement /service/42", "category": "CFS", "name": "Broadband" "description": "Description of the Broadband service", "isServiceEnabled": true, "hasStarted": true, "startMode": 0, "isStateful": false, "status": "active", "serviceSpecification": { "id": "4", "href": " http://server:port/ serviceCatalogManagement /serviceSpecification/4" }, "serviceCharacteristic": [{ "name": "speed", "value": "16M" }], "serviceRelationship": [{ "type": "contains", "service": {</pre>

```

        "id": "44",
        "href": " http://server:port/ serviceInventoryManagement /service/44"
    }
    },
    ],
    "supportingService": [
        {
            "id": "59",
            "href": " http://server:port/ serviceInventoryManagement /service/59"
        }
    ],
    "supportingResource": [
        {
            "id": "46779",
            "href": "http://server:port/resourceInventoryManagement/logicalResource/46779"
        }
    ],
    "relatedParty": [
        {
            "role": "partner",
            "id": "42",
            "href": "http://serverlocation:port/partyManagement/organisation/42"
        }
    ]
}
    
```

CREATE SERVICE

POST /service

Note: this operation is available only to ADMIN API users

Description

This operation creates a service entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	
relatedParty	

Non Mandatory Attributes	Default Value	Rule
category		
description		
endDate		
hasStarted		
isServiceEnabled		
isStateful		

orderDate		
startDate		
startMode		
status		
type		
serviceOrder		
supportingResource		
serviceRelationship		
place		
supportingService		
serviceSpecification		
relatedParty		
characteristic		

Additional Rules

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

Context	Mandatory Sub-Attributes
serviceOrder	id, href
serviceSpecification	id, href
relatedParty	role, id or href
serviceRelationship	type, Id OR href
supportingService	id, href
supportingResource	id, href
place	role, id or href

Usage Samples

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

Create a CFS service in the inventory

<p>Request</p> <pre> POST /serviceInventoryManagement/service Content-Type: application/json { "isServiceEnabled": true, "description": "Description of the Broadband service", "serviceRelationship": [{ "type": "contains", "service": { "href": " http://server:port/ serviceInventoryManagement /service/44", "id": "44" } }] } </pre>
--

```

    }
  ],
  "serviceSpecification": {
    "href": " http://server:port/ serviceCatalogManagement /serviceSpecification/4",
    "id": "4"
  },
  "relatedParty": [
    {
      "href": "http://serverlocation:port/partyManagement/organisation/42",
      "role": "partner",
      "id": "42"
    }
  ],
  "serviceCharacteristic": [
    {
      "name": "speed",
      "value": "16M"
    }
  ],
  "category": "CFS",
  "supportingService": [
    {
      "href": " http://server:port/ serviceInventoryManagement /service/59",
      "id": "59"
    }
  ],
  "name": "Broadband",
  "status": "active",
  "isStateful": false,
  "startMode": 0,
  "hasStarted": true
}

```

Response

201

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

PATCH SERVICE

PATCH /service/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a service 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.

Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
category	
description	
endDate	
hasStarted	
isServiceEnabled	
isStateful	
name	
startDate	
startMode	
status	
type	
serviceOrder	
supportingResource	
serviceRelationship	
place	
supportingService	
serviceSpecification	
reatedParty	
characteristic	

Non Patchable Attributes	Rule
id	
href	
orderDate	

Usage Samples

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

Changing the service status (using json-patch)

Request
PATCH /serviceInventoryManagement/service/42 Content-Type: application/json-patch+json <pre>{ "path": "/status", "value": "reserved",</pre>

<pre>"op": "replace" }</pre>
Response
201 { Similar JSON as in GET response with status changed}

DELETE SERVICE

DELETE /service/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service entity.

Usage Samples

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

Request
DELETE /serviceInventoryManagement/service/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 204 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 1.0	04/15/2016	Pierre Gauthier TM Forum pgauthier@tmforum.org Nicoleta Stoica Vodafone Nicoleta.stoica@vodafone.com Mariano Belaunde Orange Labs	Generated using the API data model