

# TM Forum Specification

## Service Activation and Configuration API REST Specification

**TMF640**  
**Release 18.5.0**  
**January 2019**

<b>Latest Update: TM Forum Release 18.5.0</b>	<b>Member Evaluation</b>
<b>Version 4.0.0</b>	<b>IPR Mode: RAND</b>

## NOTICE

Copyright © TM Forum 2019. 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.

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100  
Parsippany, NJ 07054, USA  
Tel No. +1 973 944 5100  
Fax No. +1 973 998 7916  
TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org)

# Table of Contents

NOTICE .....	2
Table of Contents.....	3
LIST OF TABLES.....	5
INTRODUCTION.....	6
Lifecycle of Service .....	7
Support of polymorphism and extension patterns .....	8
RESOURCE MODEL.....	9
Managed Entity and Task Resource Models.....	9
Service resource .....	9
Monitor resource .....	15
Notification Resource Models .....	17
Service Create Notification.....	18
Service Attribute Value Change Notification.....	19
Service State Change Notification .....	19
Service Delete Notification.....	19
Monitor Create Notification .....	20
Monitor Attribute Value Change Notification.....	20
Monitor State Change Notification .....	21
Monitor Delete Notification .....	21
API OPERATIONS.....	22
Operations on Service.....	23
List services.....	23
Retrieve service .....	23
Create service.....	24
Patch service.....	27
Delete service .....	28
Operations on Monitor.....	29
List monitors.....	29
Retrieve monitor .....	30
API NOTIFICATIONS.....	32
Register listener .....	32
Unregister listener .....	33

Publish Event to listener .....	33
Acknowledgements .....	35
Version History.....	35
Release History .....	35
Contributors to Document.....	35

# LIST OF TABLES

N/A

## INTRODUCTION

Services may be activated or configured by the Activation and Configuration API.

The following examples use the characteristic specification and the strongly typed patterns.

- 1) Using the characteristic specification pattern. When the characteristic specification pattern is used it is assumed that a service specification with the corresponding characteristic specifications exist.

```

"id" : "id1234567890",
"href" : "http://..",
  "state" : "active",
  "serviceSpecification":{
    "id":"conferenceBridgeEquipment",
    "href":"http://serverlocation:port/catalogManagement/serviceSpecification /conferenceBridgeEquipment"
  },
"characteristic":[
{
  "name":"numberOfVc500Units",
  "valueType": "integer",
  "value":"1"
},
{
  "name":"numberOfVc100Units",
  "valueType": "integer",
  "value":"2"
},
{
  "name":"routerType",
  "valueType": "string",
  "value":"CiscoASR1000"
},
{
  "name":"powerSupply",
  "valueType": "string",
  "value":"UK"
}
]
}

```

- 2) Using the Strongly typed pattern. When the strongly typed pattern is used it is assumed that a corresponding YAML or JSON schema is defined for the Service.

```

{
  "id" : "id1234567890",
  "href" : "http://..",
  "state" : "active",
  "@type" : "conferenceBridgeEquipment",
  "@schemaLocation" : "http://../conferenceBridgeEquipment.json",

  "serviceSpecification":{
    "id":"conferenceBridgeEquipment",
    "href":"http://serverlocation:port/catalogManagement/serviceSpecification /conferenceBridgeEquipment"
  }
}

```

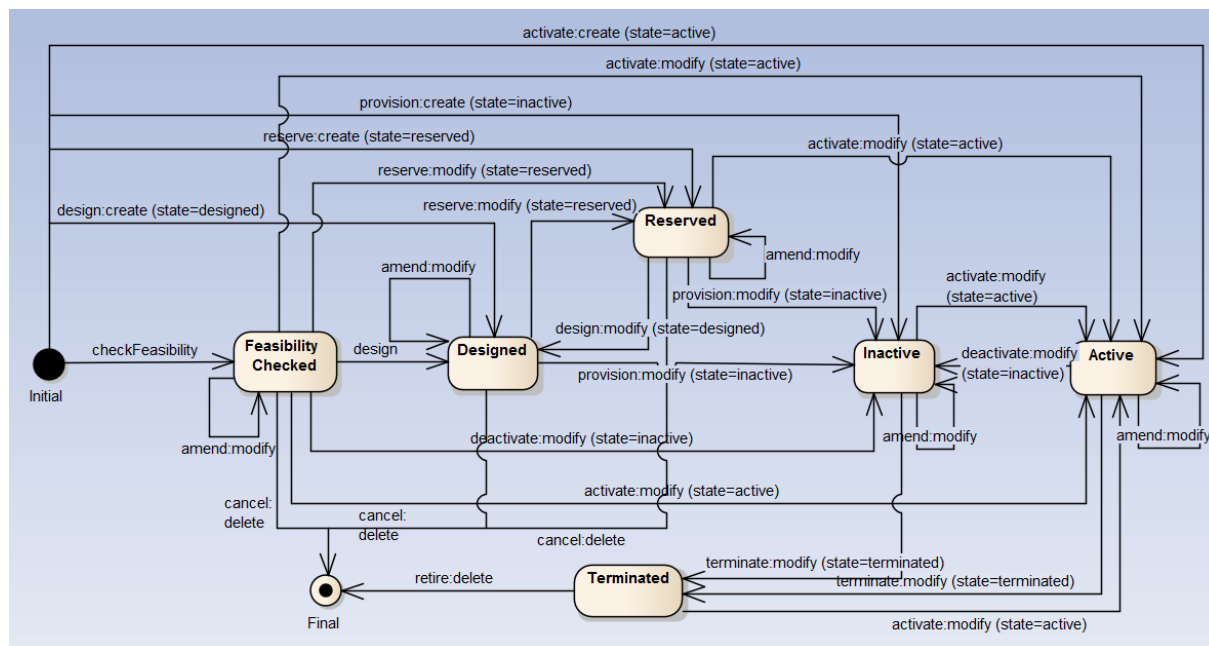
```

},
"numberOfVc500Units": "1",
"numberOfVc100Units": "2",
"routerType": "CiscoASR1000",
"powerSupply": "UK"
}
}

```

## Lifecycle of Service

Here is the state machine diagram for a Service.



## Support of polymorphism and extension patterns

Support of polymorphic collections and types and schema based extension is provided by means of a list of generic meta-attributes that we describe below. Polymorphism in collections occurs when entities inherit from base entities, for instance a TypeAService or TypeBService inheriting properties from the base Service entity.

Generic support of polymorphism and pattern extensions is described in the TMF API Guidelines v3.0 Part 2 document.

The @type attribute provides a way to represent the actual class type of an entity. For example, within a list of Service instances some may be instances of TypeAService where other could be instances of TypeBService. The @type gives this information. All resources and sub-resources of this API have a @type attributes that can be provided when this is useful.

The @referredType can be used within reference entities (like for instance a RelatedParty object) to explicitly denote the actual entity type of the referred class. Notice that in reference entities the @type, when used, denotes the class type of the reference itself, such as RelatedParty, and not the class type of the referred object. However, since reference classes are rarely sub-classed, @type is generally not useful in reference objects.

The @schemaLocation property can be used in resources to allow specifying user-defined properties of an Entity or to specify the expected characteristics of an entity.

The @baseType attribute gives a way to provide explicitly the base of class of a given resource that has been extended.



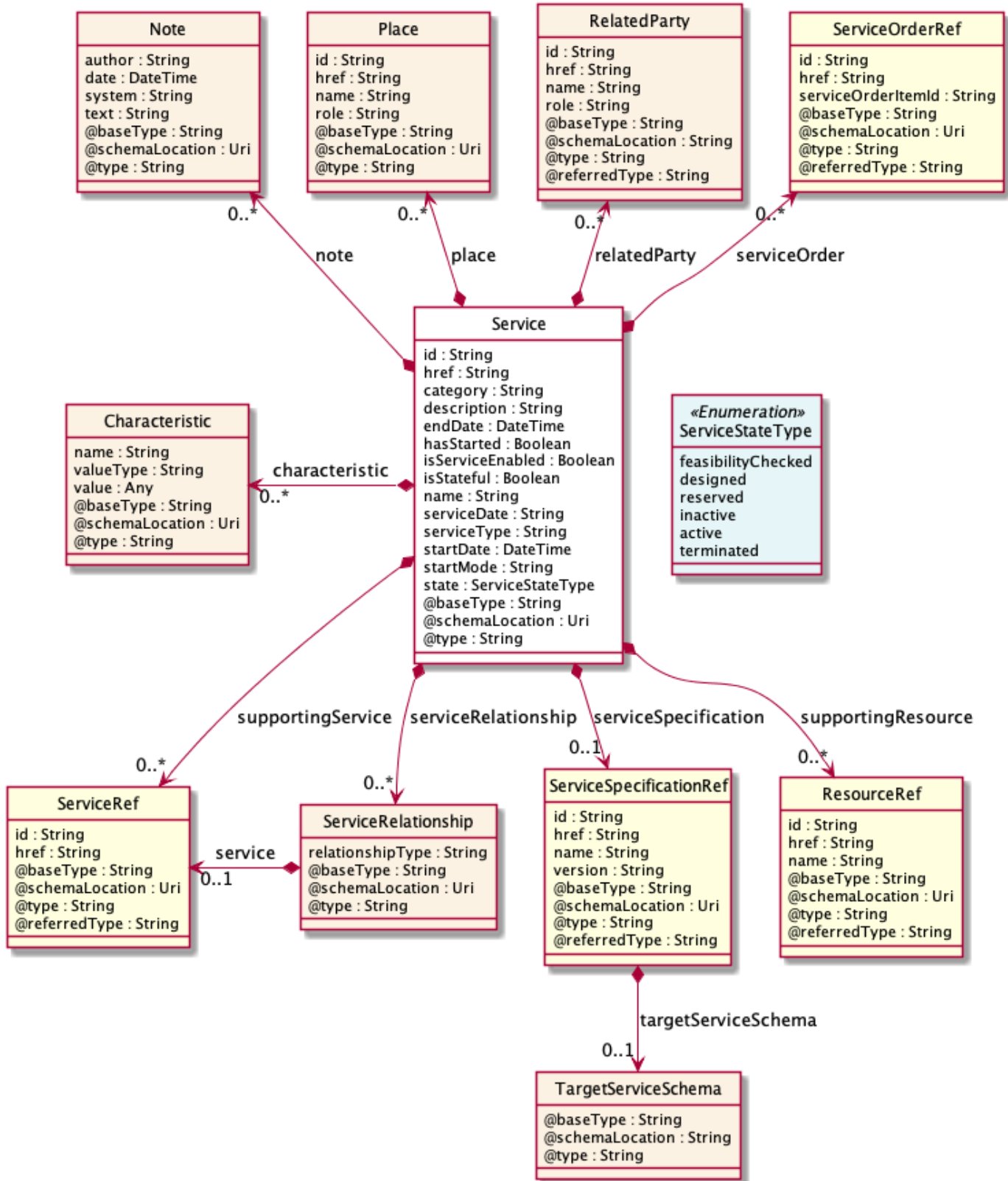
# RESOURCE MODEL

## Managed Entity and Task Resource Models

### Service resource

Service is a 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



**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). Date when the service ends.
hasStarted	A boolean. If TRUE, this Service has already been started.
href	A string. Reference of the service.
id	A string. Unique identifier of the service.
isServiceEnabled	A boolean. If FALSE, this particular Service has NOT been enabled for use.
isStateful	A boolean. If TRUE, this Service can be changed without affecting any other services.
name	A string. Name of the service.
note	A list of notes (Note [*]). A list of notes made on this service.
place	A list of places (Place [*]). A list of places related to this service, e.g. where the service is installed, a delivery address for equipment, etc.
relatedParty	A list of related parties (RelatedParty [*]). A list of related party references (RelatedParty [1..*]). A related party defines party or party role linked to a specific entity.
serviceCharacteristic	A list of characteristics (Characteristic [*]). A list of characteristics that characterize this service (ServiceCharacteristic [*]).
serviceDate	A string. Date when the service was created (whatever its status).
serviceOrder	A list of service order references (ServiceOrderRef [*]). A list of service orders related to this service.
serviceRelationship	A list of service relationships (ServiceRelationship [*]). A list of service relationships (ServiceRelationship [*]). Describes links with other service(s) in the inventory (useful for describing relies-on, relies-from between CFS for example).
serviceSpecification	A service specification reference (ServiceSpecificationRef). The specification from which this service was instantiated.
serviceType	A string. Business type of the service.
startDate	A date time (DateTime). Date when the service starts.

startMode	A string. This attribute is an enumerated integer that indicates how the Service is started, such as: 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.
state	A service state type (ServiceStateType). The life cycle state of the service, such as: feasibilityChecked, designed, reserved, active, inactive, terminated.
supportingResource	A list of resource references (ResourceRef [*]). A list of supporting resources (SupportingResource [*]). Note: only Service of type RFS can be associated with Resources.
supportingService	A list of service references (ServiceRef [*]). A list of supporting services (SupportingService [*]). A collection of services that support this service (bundling, link CFS to RFS).

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

name	A string. Name of the characteristic.
value	An any (Any). The value of the characteristic.
valueType	A string. Data type of the value of the characteristic.

Note sub-resource

Extra information about a given entity.

author	A string. Author of the note.
date	A date time (DateTime). Date of the note.
system	A string. Describes the system from which the action related to this note was done.
text	A string. Text of the note.

Place sub-resource

Place reference. Place defines the places where the products are sold or delivered.

href	A string. Unique reference of the place.
id	A string. Unique identifier of the place.
name	A string. A user-friendly name for the place, such as [Paris Store], [London Store], [Main Home].
role	A string. Role of the place, such as: [home delivery], [shop retrieval]).

RelatedParty sub-resource

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

@baseType	A string. When sub-classing, this defines the super-class.
@referredType	A string. The actual type of the target instance when needed for disambiguation.
@schemaLocation	A string. A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class entity name.
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.

ServiceRelationship sub-resource

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

relationshipType	A string. The type of relationship (e.g. depends on, enables).
service	A service reference (ServiceRef). The service being referred to.

TargetServiceSchema sub-resource

The reference object to the schema and type of target service which is described by service specification.

@schemaLocation	A string. This field provides a link to the schema describing the target service.
@type	A string. Class type of the target service.

ResourceRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
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.

ServiceOrderRef relationship

Service Order reference. Useful to understand the which was the Service order through which the service was instantiated in the service inventory.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the Service Order.
id	A string. Unique identifier of the Service Order.
serviceOrderItemId	A string. Unique identifier of the Service Order Item within a service order, not populated if this is a reference to a service order.

ServiceRef relationship

Service reference, for when Service is used by other entities.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
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.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the serviceSpecification.
id	A string. Unique identifier of the service specification.
name	A string. Name of the requiredServiceSpecification.
targetServiceSchema	A target service schema (TargetServiceSchema). A target service schema reference (TargetServiceSchemaRef). The reference object to the schema and type of target service which is described by service specification.
version	A string. Service specification version.

**Json representation sample**

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

```
{
  "id": "5351",
  "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",
  "serviceType": "Cloud",
  "state": "Active",
  "serviceDate": "2018-01-15T12:26:11.747Z",
  "startDate": "2018-01-15T12:26:11.747Z",
}
```

```

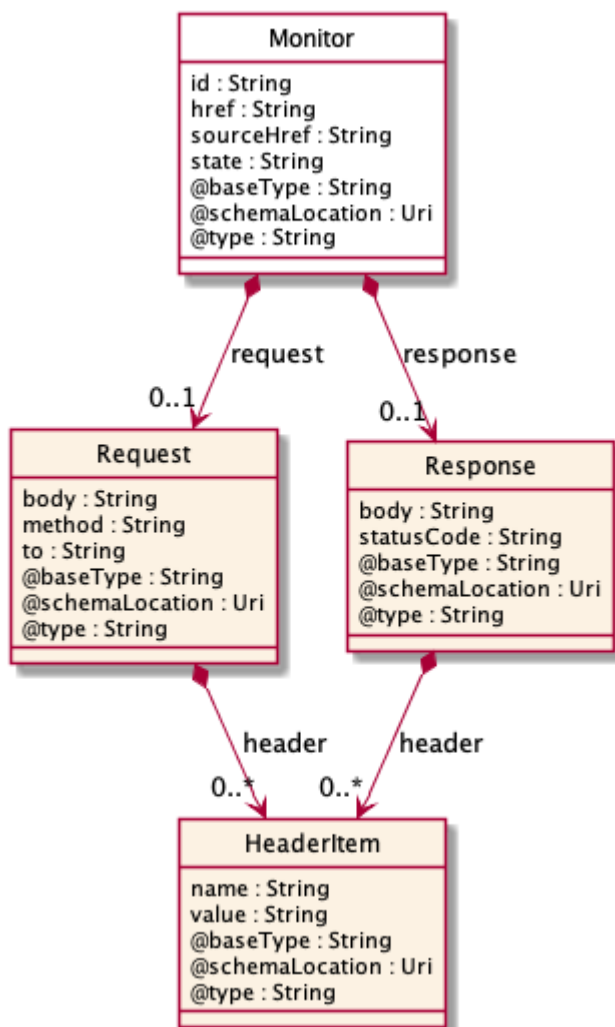
"serviceCharacteristic": [
  {
    "name": "vCPE_IP",
    "valueType": "object",
    "value": {
      "@type": "IPAddress",
      "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",
      "address": "193.218.236.21"
    }
  }
],
"@type": "vCPE",
"@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/vCPE.schema.json",
"@baseType": "Service"
}

```

## Monitor resource

Monitoring of resources.

### Resource model



## Field descriptions

### Monitor fields

href	A string. reference to this monitor.
id	A string. Identifier of an instance of the monitor. Required to be unique within the resource type. Used in URIs as the identifier for specific instances of a type.
request	A request (Request). Represents the request.
response	A response (Response). Represents the response.
sourceHref	A string. The monitored resource href.
state	A string. The Monitor state of the resource. InProgress, InError, Completed.

### HeaderItem sub-resource

An item typically included in a request or response.

name	A string. The name of the header item, e.g. locale.
value	A string. The value of the header item, e.g. en-us.

### Request sub-resource

A response to a request.

body	A string. The body of the request. For example for an HTTP request might contain content of a form .
header	A list of header items (HeaderItem [1..*]). Items included in the header of the request. For example for an HTTP request might contain requested locale, basic authentication.
method	A string. The protocol of the request, e.g. http.
to	A string. The target of the request, e.g. a URL for an HTTP request.

### Response sub-resource

A response to a request.

body	A string. The body of the response. For example for an HTTP response might contain HTML for rendering.
header	A list of header items (HeaderItem [1..*]). Items included in the header of the response. For example for an HTTP response might contain negotiated locale.



**statusCode** A string. The status of the response. For example for an HTTP response would be codes such as 200, 400, etc.

### Json representation sample

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

```
{
  "id": "44555",
  "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/monitor/44555",
  "state": "In Progress",
  "request": {
    "method": "http",
    "to": "https://mycsp.com:5050/netAPIs/activate",
    "body": "ACTIVATE BBAND 144445",
    "header": [
      {
        "name": "locale",
        "value": "en-us"
      }
    ]
  },
  "response": {
    "statusCode": "344",
    "body": "ACTIVATE BBAND 144445 REQUEST RECEIVED PENDING",
    "header": [
      {
        "name": "locale",
        "value": "en-us"
      }
    ]
  },
  "sourceHref": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",
  "@type": "Monitor"
}
```

## Notification Resource Models

8 notifications are defined for this API

Notifications related to Service:

- ServiceCreateNotification
- ServiceAttributeValueChangeNotification
- ServiceStateChangeNotification
- ServiceDeleteNotification

Notifications related to Monitor:

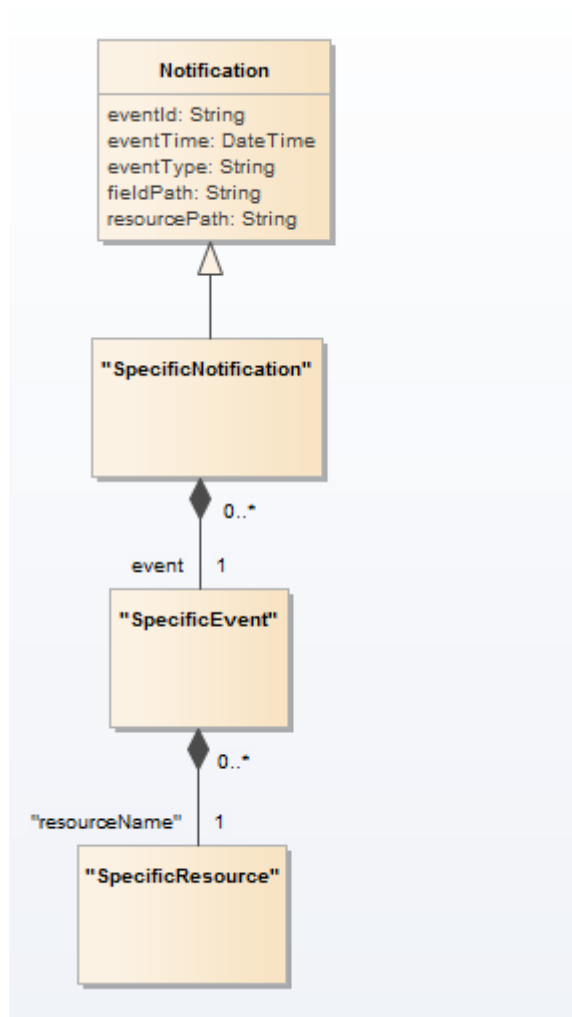
- MonitorCreateNotification
- MonitorAttributeValueChangeNotification

- MonitorStateChangeNotification
- MonitorDeleteNotification

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 Create Notification

Notification ServiceCreateNotification case for resource Service

### Json representation sample

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

```

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

```

```
"eventType": "ServiceCreateNotification",
"event": {
  "service":
    {-- SEE Service RESOURCE SAMPLE --}
}
}
```

### Service Attribute Value Change Notification

Notification ServiceAttributeValueChangeNotification case for resource Service

#### 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 ServiceStateChangeNotification case for resource Service

#### 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 Delete Notification

Notification ServiceDeleteNotification case for resource Service

#### Json representation sample

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

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

### Monitor Create Notification

Notification MonitorCreateNotification case for resource Monitor

#### Json representation sample

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

```
{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "MonitorCreateNotification",
  "event": {
    "monitor": {
      "-- SEE Monitor RESOURCE SAMPLE --"
    }
  }
}
```

### Monitor Attribute Value Change Notification

Notification MonitorAttributeValueChangeNotification case for resource Monitor

#### Json representation sample

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

```
{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "MonitorAttributeValueChangeNotification",
  "event": {
    "monitor": {
      "-- SEE Monitor RESOURCE SAMPLE --"
    }
  }
}
```

## Monitor State Change Notification

Notification MonitorStateChangeNotification case for resource Monitor

### Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"MonitorStateChangeNotification",
  "event": {
    "monitor" :
      {-- SEE Monitor RESOURCE SAMPLE --}
  }
}
```

## Monitor Delete Notification

Notification MonitorDeleteNotification case for resource Monitor

### Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"MonitorDeleteNotification",
  "event": {
    "monitor" :
      {-- SEE Monitor 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 Part 1 document.

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 services from the network.

Request
GET {apiRoot}/service?fields=id,name,state&serviceType=Cloud&state=Active Accept: application/json
Response
200  [ { "id": "5351", "name": "vCPE serial 1355615", "state": "Active" }, { "id": "5352", "name": "vDPI serial 1355445", "state": "Active" } ]

### 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 specific service from the network. The level of detail returned is minimal, according to what might be held in the network.

<b>Request</b>
GET {apiRoot}/service/5351 Accept: application/json
<b>Response</b>
200  <pre>{   "id": "5351",   "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",   "serviceType": "Cloud",   "state": "Active",   "serviceDate": "2018-01-15T12:26:11.747Z",   "startDate": "2018-01-15T12:26:11.747Z",   "serviceCharacteristic": [     {       "name": "vCPE_IP",       "valueType": "object",       "value": {         "@type": "IPAddress",         "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",         "address": "193.218.236.21"       }     }   ],   "@type": "vCPE",   "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/vCPE.schema.json",   "@baseType": "Service" }</pre>

### Create service

#### POST /service

#### 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. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
----------------------	------

Non Mandatory Attributes	Default Value	Rule
category		
description		
endDate		
hasStarted	blank	
isServiceEnabled	false	
isStateful		
name		
note		
place		
relatedParty		
serviceCharacteristic		
serviceDate	<date of POST>	
serviceOrder		
serviceRelationship		
serviceSpecification		
serviceType		
startDate	<date of POST>	
startMode		
state		
supportingResource		
supportingService		

### 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 OR href
serviceSpecification	id OR href
relatedParty	id OR href, role

### Usage Samples

Here's an example of a request for activating a new service on the network.

### Request

POST {apiRoot}/service

Content-Type: application/json

```
{
  "serviceType": "Cloud",
  "serviceDate": "2018-01-15T12:26:11.747Z",
  "startDate": "2018-01-15T12:26:11.747Z",
  "serviceCharacteristic": [
    {
      "name": "vCPE_IP",
      "valueType": "object",
      "value": {
        "@type": "IPAddress",
        "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",
        "address": "193.218.236.21"
      }
    }
  ]
}
```

### Response

201

```
{
  "id": "5351",
  "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",
  "serviceType": "Cloud",
  "state": "Active",
  "serviceDate": "2018-01-15T12:26:11.747Z",
  "startDate": "2018-01-15T12:26:11.747Z",
  "serviceCharacteristic": [
    {
      "name": "vCPE_IP",
      "valueType": "object",
      "value": {
        "@type": "IPAddress",
        "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",
        "address": "193.218.236.21"
      }
    }
  ],
  "@type": "vCPE",
  "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/vCPE.schema.json",
  "@baseType": "Service"
}
```

**Patch service****PATCH /service/{id}****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.

Patchable Attributes	Rule
category	
description	
endDate	
hasStarted	
isServiceEnabled	
isStateful	
name	
note	
place	
relatedParty	
serviceCharacteristic	
serviceDate	
serviceOrder	
serviceRelationship	
serviceSpecification	
serviceType	
startDate	
startMode	
state	
supportingResource	
supportingService	

Non Patchable Attributes	Rule
id	
href	

## Usage Samples

Here's an example of a request for updating a service in the network. In this example, a new state is set.

Request
PATCH {apiRoot}/service/5351 Content-Type: application/merge-patch+json  <pre>{   "state": "Suspended" }</pre>
Response
200  <pre>{   "id": "5351",   "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",   "serviceType": "Cloud",   "state": "Suspended",   "serviceDate": "2018-01-15T12:26:11.747Z",   "startDate": "2018-01-15T12:26:11.747Z",   "serviceCharacteristic": [     {       "name": "vCPE_IP",       "valueType": "object",       "value": {         "@type": "IPAddress",         "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/IPAddress.schema.json",         "address": "193.218.236.21"       }     }   ],   "@type": "vCPE",   "@schemaLocation": "https://mycsp.com:8080/tmf-api/schema/Service/vCPE.schema.json",   "@baseType": "Service" }</pre>

## Delete service

**DELETE** /service/{id}

### Description

This operation deletes a service entity.

## Usage Samples

Here's an example of a request for deactivating a service in the network.

<b>Request</b>
DELETE {apiRoot}/service/5351
<b>Response</b>
204

## Operations on Monitor

### List monitors

**GET /monitor?fields=...&{filtering}**

#### Description

This operation list monitor 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 monitors for service activation.

<b>Request</b>
GET {apiRoot}/monitor?serviceType=Cloud&state=InProgress Accept: application/json
<b>Response</b>
200  [ { "id": "44555", "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/monitor/44555", "state": "In Progress",

```

"request": {
  "method": "http",
  "to": "https://mycsp.com:5050/netAPIs/activate",
  "body": "ACTIVATE BBAND 144445",
  "header": [
    {
      "name": "locale",
      "value": "en-us"
    }
  ]
},
"response": {
  "statusCode": "344",
  "body": "ACTIVATE BBAND 144445 REQUEST RECEIVED PENDING",
  "header": [
    {
      "name": "locale",
      "value": "en-us"
    }
  ]
},
"sourceHref": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",
"@type": "Monitor"
}
]

```

## Retrieve monitor

**GET /monitor/{id}?fields=...&{filtering}**

### Description

This operation retrieves a monitor 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 specific monitor for service activation.

<b>Request</b>
GET {apiRoot}/monitor/44555 Accept: application/json
<b>Response</b>

200

```
{
  "id": "44555",
  "href": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/monitor/44555",
  "state": "In Progress",
  "request": {
    "method": "http",
    "to": "https://mycsp.com:5050/netAPIs/activate",
    "body": "ACTIVATE BBAND 144445",
    "header": [
      {
        "name": "locale",
        "value": "en-us"
      }
    ]
  },
  "response": {
    "statusCode": "344",
    "body": "ACTIVATE BBAND 144445 REQUEST RECEIVED PENDING",
    "header": [
      {
        "name": "locale",
        "value": "en-us"
      }
    ]
  },
  "sourceHref": "https://mycsp.com:8080/tmf-api/ServiceActivationAndConfiguration/v4/service/5351",
  "@type": "Monitor"
}
```

## 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 part 1. Refer to the guidelines for more details.

### 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.

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



## 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.

<b>Request</b>
DELETE /api/hub/42 Accept: application/json
<b>Response</b>
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.

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

For detailed examples on the general TM Forum notification mechanism, see the TMF REST Design Guidelines Part 1 document.

## Acknowledgements

### Version History

Release Number	Date	Release led by:	Description
Version 1.0.0	12 Nov 2015	Pierre Gauthier TM Forum	Submitted for Fx15.5
Version 1.0.1	12 Nov 2015	Alicja Kawecki TM Forum	Updated cover, header; minor formatting/style corrections prior to publishing
Version 1.0.2	02 May 2016	Alicja Kawecki, TM Forum	Updated cover, footer, Notice to reflect TM Forum Approved status
Version 4.0.0	14 Jan 2019	Jonathan Goldberg Amdocs <a href="mailto:Jonathan.Goldberg@amdocs.com">Jonathan.Goldberg@amdocs.com</a>	Schema alignment for NaaS APIs

### Release History

Release Number	Date	Release led by:	Description
15.5.1	02 May 2016	Alicja Kawecki, TM Forum	Updated to reflect TM Forum Approved Status
18.5.0	14 Jan 2019	Jonathan Goldberg Amdocs <a href="mailto:Jonathan.Goldberg@amdocs.com">Jonathan.Goldberg@amdocs.com</a>	Schema alignment for NaaS APIs

### Contributors to Document

Andrew Forth	Amdocs
August-Wilhelm Jagau	Ericsson
Pierre Gauthier	TM Forum
Jonathan Goldberg	Amdocs