



Framework Specification

Service Ordering Management API REST Specification

TMF641
Release 16.5
October 2016

Latest Update: Framework Release 16.5

Member Evaluation

Version 1.0.0	IPR Mode: RAND
----------------------	-----------------------

NOTICE

Copyright © TM Forum 2016. 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:

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	3
Table of Contents	4
List of Tables	6
Introduction	7
SAMPLE USE CASES	8
RESOURCE MODEL	9
Managed Entity and Task Resource Models.....	9
Service Order resource	9
Notification Resource Models.....	19
Service Order Creation Notification.....	20
Service Order Attribute Value Change Notification	21
Service Order State Change Notification.....	21
Service Order Remove Notification	21
Service Order Information Required Notification.....	22
API OPERATIONS.....	24
Operations on Service Order.....	25
List service orders	25
Retrieve service order	28
Create service order.....	31
Patch service order	35
Delete service order	37
API NOTIFICATIONS	39
Register listener	39
Unregister listener	40
Publish Event to listener.....	40

Acknowledgements.....42

 Release History 42

SERVICE ORDER EXAMPLE43

 BROADBAND SERVICE EXAMPLE 43

LIST OF TABLES

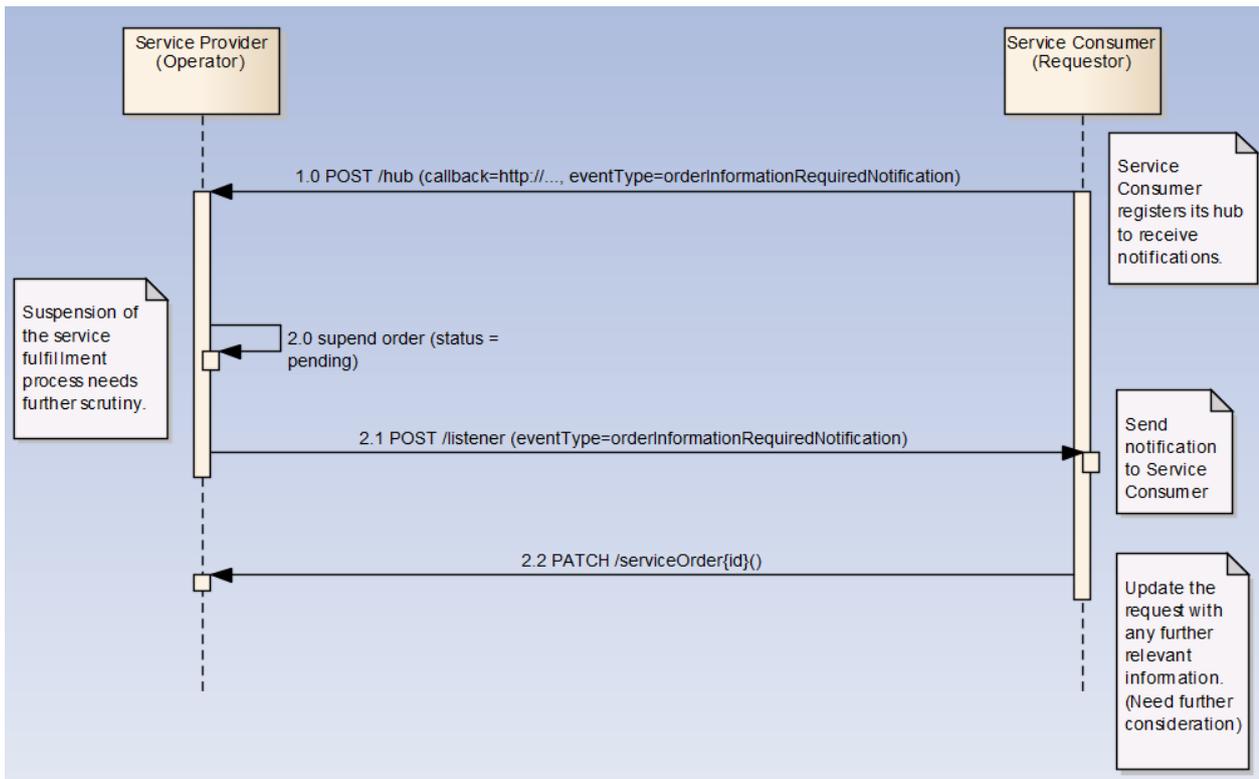
N/A

INTRODUCTION

The following document is the specification of the REST API for Service Order Management. It includes the model definition as well as all available operations. Possible actions are creating, updating and retrieving Service Orders (including filtering).

The following Assumptions were considered in the development of this document :

- The Order Management system has access to a catalog system



A service order will describe a list of service order items. A service order item references an action on an existing or future service. By service we designed Customer-Facing Service (CFS) as well as Resource Facing Service (RFS).

From a component perspective, a service order should be available

- from a Service Orchestration Component (and it could mix CFS and RFS)
- from an Infrastructure Control & Management component (and it would have only RFS)

SAMPLE USE CASES

No use cases

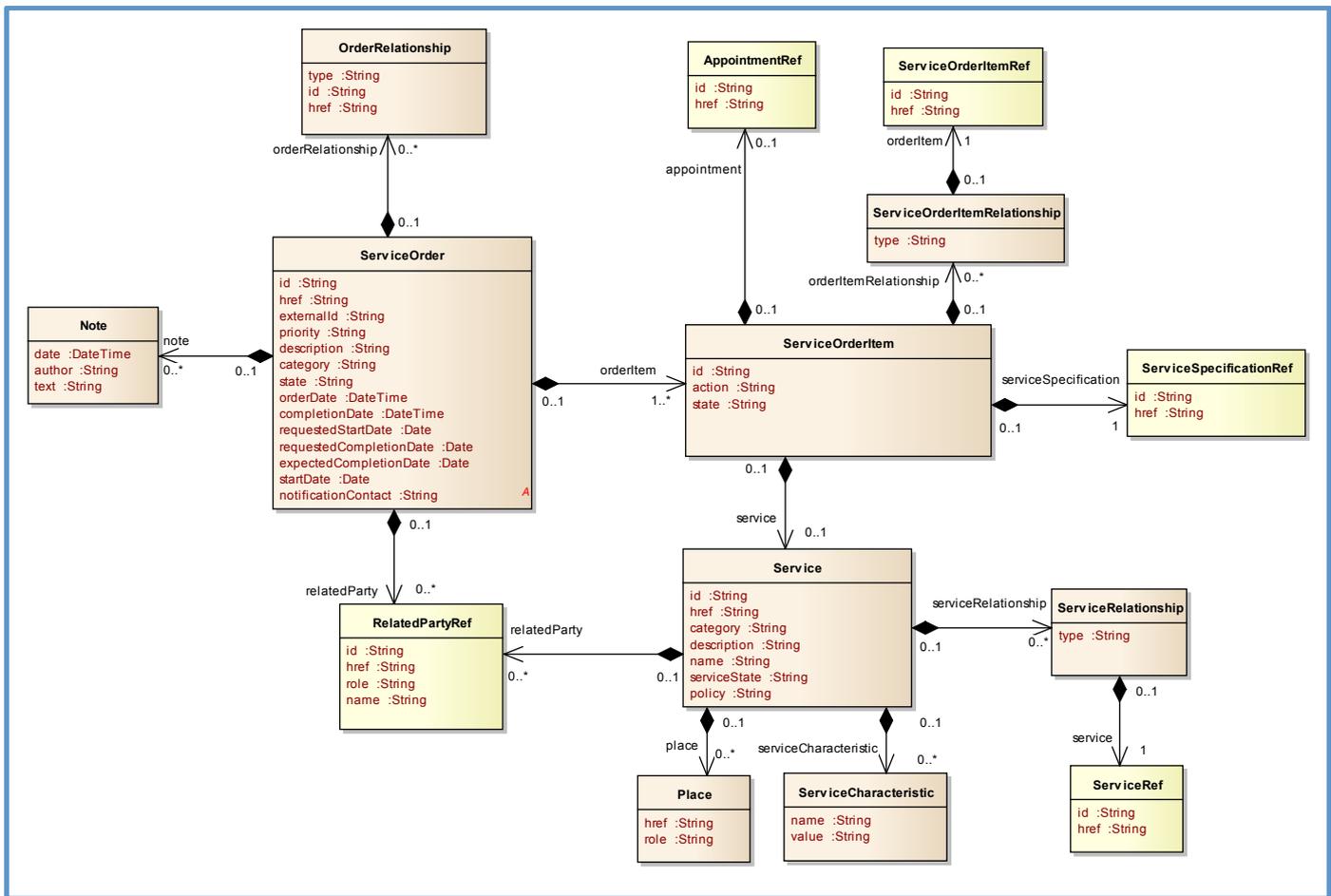
RESOURCE MODEL

Managed Entity and Task Resource Models

SERVICE ORDER RESOURCE

A Service Order is a type of order which can be used to place an order between a customer and a service provider or between a service provider and a partner and vice versa,.

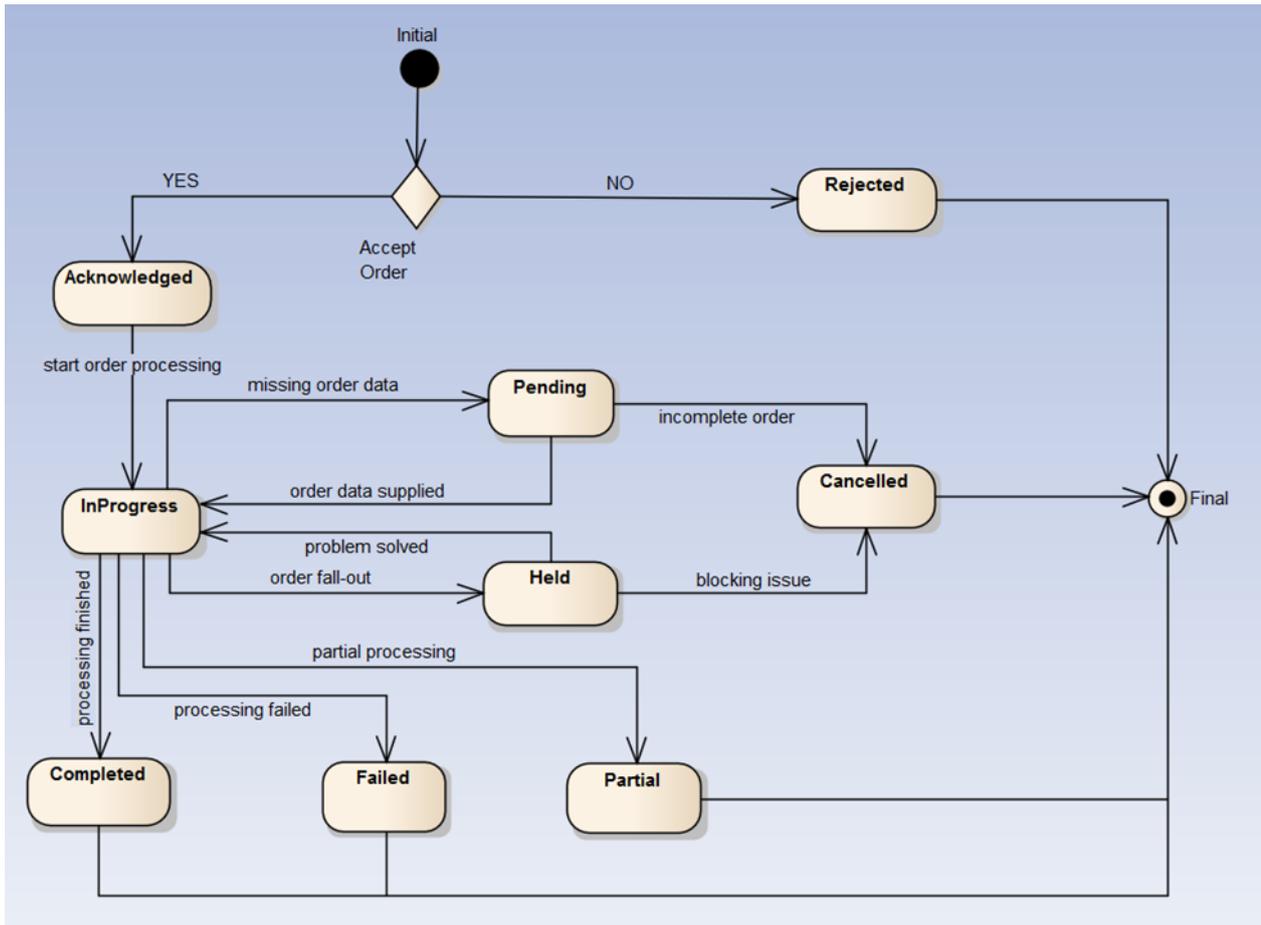
Resource model



Lifecycle

Here is the state machine diagram for a Service order. Each order state is described in the table below.

The order item states are the same as the order ones. Note that the order and order item states are tightly linked and need to be consistent (see table below):



The following table provides service order state and service order item state description :

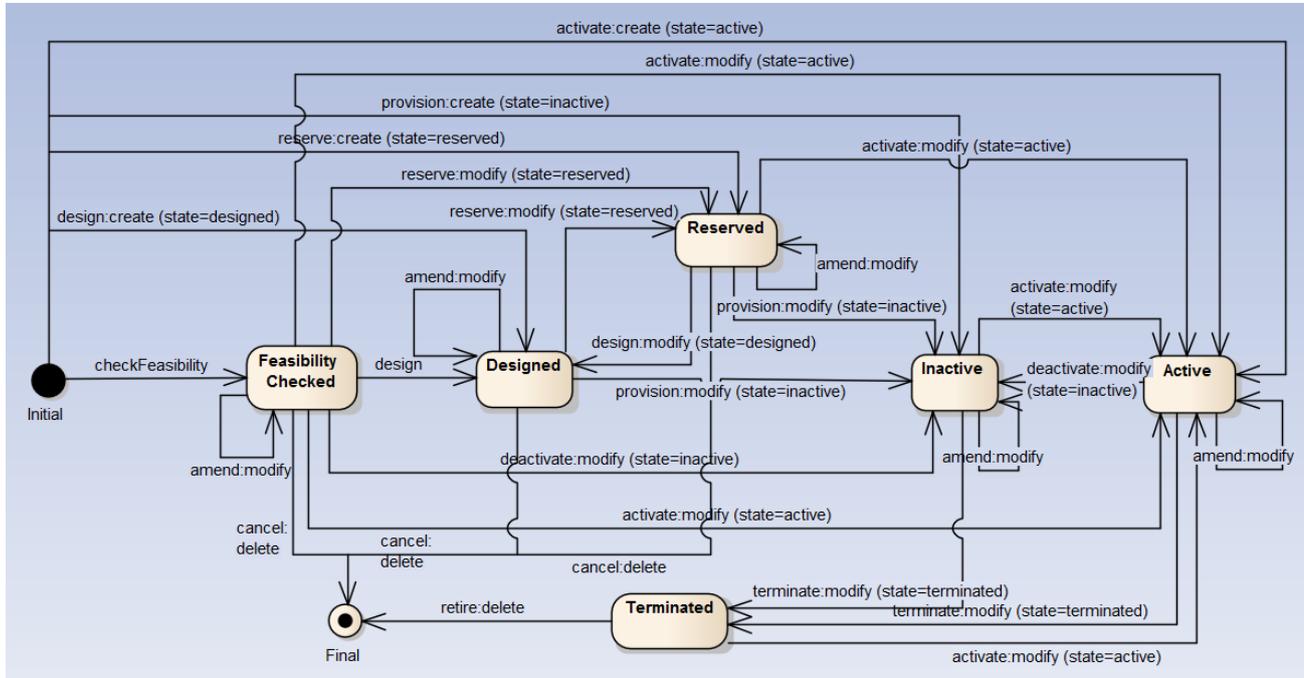
Acknowledged	The Acknowledged state is where an order has been received and has passed message and basic business validations.
In Progress	The In Progress state is when service delivery has started.
Cancelled	The Cancelled state is where an In-Flight Order has been successfully cancelled.
Completed	The Completed state is where an order has complete provision and the service is now active.
Rejected	<p>The Rejected state is where:</p> <ul style="list-style-type: none"> - An order failed the Order Feasibility check (but service technical eligibility is not done though service order API but with dedicated serviceQualification API (from preOrdering domain) - Invalid information is provided through the order request - The order request fails to meet business rules for ordering.

Pending	<p>The Pending state is used when an order is currently in a waiting stage for an action/activity to be completed before the order can progress further, pending order amend or cancel assessment. In situations where Access Seeker action is required, an “information required” notification will be issued on transition into this state.</p> <p>A pending stage can lead into auto cancellation of an order, if no action is taken within the defined timeframes to be described under the Agreement.</p>
Held	<p>The Held state is used when an order cannot be progressed due to an issue. SP has temporarily delayed completing an order to resolve an infrastructure shortfall to facilitate supply of order. Upon resolution of the issue, the order will continue to progress.</p>
Failed	<p>All Order items have failed which results in the entire Order has Failed.</p>
Partial	<p>Some Order items have failed and some have succeeded so the entire Order is in a Partial state. This provides support for partial Failure of an Order</p>

Consistency between Service Order state and Service Order Item state table:

If service order state has state...	the service order items state should be...
Rejected	All ‘Rejected’
Acknowledged	All ‘Acknowledged’ note: once delivery begins for at least an item the SO state shifts to ‘In Progress’
in Progress	At least one SO item has ‘In Progress’ state All items should be ‘Acknowledged’
Pending	All ‘Pending state’
Held	All ‘Held’ state
Cancelled	All ‘Cancelled’
Partial	All Order item are either ‘Failed’ or ‘Completed’ At least one item has ‘Failed’ state At least one item has ‘Completed’ state
Failed	All ‘Failed’
Completed	All ‘Completed’

Service state model:



Note: 'Feasibility Checked' should not be managed through service order. A dedicated API provides service qualification; This API is part of the pre-ordering domain.

Consistency between Service Order Item state and service state:

source service state	action	target service state
blank	add	Designed (valued in order item service state) Reserved (valued in order item service state) Inactive (valued in order item service state) Active – by default
Designed	modify	Reserved (valued in order item service state) Inactive (valued in order item service state) Designed– by default
Designed	delete	<i>not relevant...service removed</i>
Reserved	delete	<i>not relevant...service removed</i>
Inactive	modify	Terminated (valued in order item service state) Active (valued in order item service state) Inactive– by default
Active		Active – by default Terminated (valued in order item service state) Inactive (valued in order item service state)
Terminated	delete	<i>not relevant...service removed</i>

Note: when action 'no change' is used we did not expect any state change.

Field descriptions

ServiceOrder

id	A string. ID created on repository side
href	A string. Hyperlink to access the order
externalId	A string. ID given by the consumer and only understandable by him (to facilitate his searches afterwards)
priority	A string. A way that can be used by consumers to prioritize orders in Service Order Management system (from 0 to 4 : 0 is the highest priority, and 4 the lowest)
description	A string. A free-text description of the service order.
category	A string. Used to categorize the order from a business perspective that can be useful for the Service Order Management system (e.g. "broadband", "TV option", ...).
state	A string. State of the order : described in the state-machine diagram.
orderDate	A date time (DateTime). Date when the order was created
completionDate	A date time (DateTime). Date when the order was completed.
requestedStartDate	A date (Date). Order start date wished by the requestor
requestedCompletionDate	A date (Date). Requested delivery date from the requestor perspective.
expectedCompletionDate	A date (Date). Expected delivery date amended by the provider
startDate	A date (Date). Order start date wished by the requestor
notificationContact	A string. Contact attached to the order to send back information regarding this order
note	A list of notes (Note [*]). Extra-information about the order (e.g. useful to add extra delivery information that could be useful for a human process)
orderItem	A list of order items (ServiceOrderItem [*]). Order items that have to be processed.
orderRelationship	A list of related order references (OrderRelationship [*]). Linked order to the one containing this attribute

relatedParty A list of related party references (RelatedPartyRef [*]). Parties which are involved in this order and the role they are playing.

Note

Extra-information about the order (e.g. useful to add extra delivery information that could be useful for a human process).

author A string. Author of the note.

date A date time (DateTime). Date of the note.

text A string. Text of the note.

OrderRelationship

Linked order to the one containing this attribute.

href A string. An hyperlink to the related order.

id A string. The id of the related order.

type A string. The type of related order, can be:

- “dependency” if the order needs to be “not started” until another order item is complete (a service order in this case)
- “cross-ref” to keep track of the source order (a productOrder)

OrderItem

An identified part of the order. A service order is decomposed into one or more order items.

id A string. Identifier of the line item (generally it is a sequence number 01, 02, 03, ...).

action A string. The action to be carried out on the Service. Can be:

- add
- modify
- delete
- noChange

These align to the Service Activation operations POST, PATCH, DELETE and this shows how Service Activation request information can be included in service order items

state A string. State of the order item : described in the state machine diagram.

appointment	An appointment references (AppointmentRef). Used to precise that an appointment was set up with a related party for this order item.
serviceSpecification	A service specification (ServiceSpecificationRef). The service specification (default values, etc. are fetched from the catalogue).
orderItemRelationship	A list of order items relationships (ServiceOrderItemRelationship[*]). Linked order items to the one containing this attribute.
service	A service references (Service). The Service to be acted on by the order item.

OrderItemRelationship

Linked order item to the one containing this attribute.

type	A string. The type of related order item, can be: <ul style="list-style-type: none"> “dependency” if the order item needs to be “not started” until another order item is complete
------	---

orderItem	A service order item references (ServiceOrderItemRef). An id and an hyperlink to the related order item.
-----------	--

Appointment

Used to precise that an appointment was set-up with a related party for this order item.

id	A string. The id of the appointment.
href	A string. An hyperlink to the appointment

Place

Used to defined 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. The role of the place (e.g. delivery address, install site etc).

Service

Service attributes description (these are as per the Service ODE model as used in the Service Inventory specification).

id	A string. Identifier of a service instance. Required to be unique. Used in URIs as the identifier of the service (for modify or delete use cases).
href	A string. Reference to the owned Service (useful for delete or modify command).
category	A string. The category of the service (e.g. CFS, RFS).

description	A string. A description of the service (what it provides).
name	A string. Name of the service.
serviceState	A string. The lifecycle state of the service (as per state diagram below).
place	A list of places (Place [*]). Used to defined places useful for the service (for example a delivery geographical place).
serviceCharacteristic	A list of service characteristics (ServiceCharacteristic[*]).A name/value pair list used to store instance specific values of attributes. The behavior is equivalent to a MAP data structure where only one entry for any given value of "name" can exist.
serviceRelationship	A list or service relationships (ServiceRelationship[*]). Linked Services to the one instantiate, it can be : <ul style="list-style-type: none"> • “reliesOn” if the Service needs another already owned Service to rely on (e.g. an option on an already owned mobile access Service) • “targets” or “isTargeted” (depending on the way of expressing the link) for any other kind of links that may be useful
relatedParty	A list of related party references (RelatedPartyRef[*]). Parties linked at the Service level (it may be a User for example).

[RelatedPartyRef](#)

Related party references. A related party defines party which are involved in this order and the role they are playing.

id	A string. Unique identifier of a related party.
href	A string. An hyperlink to the party.
role	A string. The role of the related party (e.g. Owner, requester, fulfiller etc).
name	A string. Name of the related party.

Json representation sample

We provide below the json representation of a sample of a ServiceOrder resource composed of 3 order lines (orderItems) :

- Line “1” : Ordering of a new simple Service that needs a physical delivery place and an appointment to be delivered
- Line “2” : Modification of a characteristic value of an already owned Service, and change the user associated to this Service
- Line “3” : Ordering of a new simple Service that needs (is supported by) another already owned Service ()
- Line “4” : Modification of an already owned Service to switch its state to ‘inactive’.

```

{
  "id": "42",
  "href": "http://serverlocation:port/orderManagement/serviceOrder/42",
  "externalId": "NiceNameForTheConsumer_42",
  "priority": "1",
  "description": "A wonderful 42 order for brand new Services",
  "category": "Broadband",
  "state": "InProgress",
  "orderDate": "2013-04-12T16:42:23-04:00",
  "completionDate": "2013-04-19T16:42:23-04:00",
  "requestedStartDate": "2013-04-12",
  "requestedCompletionDate": "2013-04-19",
  "expectedCompletionDate": "2013-04-19",
  "notificationContact": "emailAddr-myAddress@hotmail.com",
  "note": [
    {
      "text": "A free text detailing the note",
      "date": "2013-04-12T16:42:23-04:00",
      "author": "name"
    }
  ],
  "relatedParty": [
    {
      "role": "requester",
      "id": "345231",
      "name": "John Doe"
    },
    {
      "role": "fulfiller",
      "href": "http://serverlocation:port/provider/42"
    }
  ],
  "orderItem": [
    {
      "id": "1",
      "action": "add",
      "state": "Acknowledged",
      "appointment": {
        "id": "89",
        "href": "http://www.doodle.com/1WCV5647438"
      }
    },
    {
      "serviceSpecification": {
        "id": "42",
        "href": "http://serverlocation:port/catalogManagement/serviceSpecification/42"
      },
      "service": {
        "place": [
          {
            "href": "http://map.google.com/.../1234112GDE",
            "role": "DeliveryPlace"
          }
        ]
      },
      "serviceCharacteristic": [

```

```
    {
      "name": "Colour",
      "value": "White"
    },
    {
      "name": "Memory",
      "value": "16"
    }
  ]
}
},
{
  "id": "2",
  "action": "modify",
  "state": "InProgress",
  "serviceSpecification": {
    "id": "43",
    "href": "http://serverlocation:port/catalogManagement/serviceSpecification/43"
  },
  "service": {
    "id": "456",
    "href": "http://serverlocation:port/inventoryManagement/service/456",
    "serviceCharacteristic": [
      {
        "name": "anotherCharacteristic",
        "value": "itsNewValue"
      }
    ],
    "relatedParty": [
      {
        "role": "user",
        "id": "5667443",
        "name": "Jimmy Doe"
      }
    ]
  }
},
{
  "id": "3",
  "action": "add",
  "state": "InProgress",
  "serviceSpecification": {
    "id": "51",
    "href": "http://serverlocation:port/catalogManagement/serviceSpecification/51"
  },
  "service": {
    "serviceRelationship": [
      {
        "type": "reliesOn",
        "service": {
          "id": "511",
          "href": "http://serverlocation:port/inventoryManagement/service/511"
        }
      }
    ]
  }
}
```

```
    ]
  }
},
{
  "id": "4",
  "action": "modify",
  "state": "InProgress",
  "serviceSpecification": {
    "id": "89",
    "href": "http://serverlocation:port/catalogManagement/serviceSpecification/89"
  },
  "service": {
    "id": "120",
    "href": "http://serverlocation:port/inventoryManagement/service/120",
    "serviceState": "Inactive"
  }
}
]
}
```

Notification Resource Models

Five notifications are defined for this API.

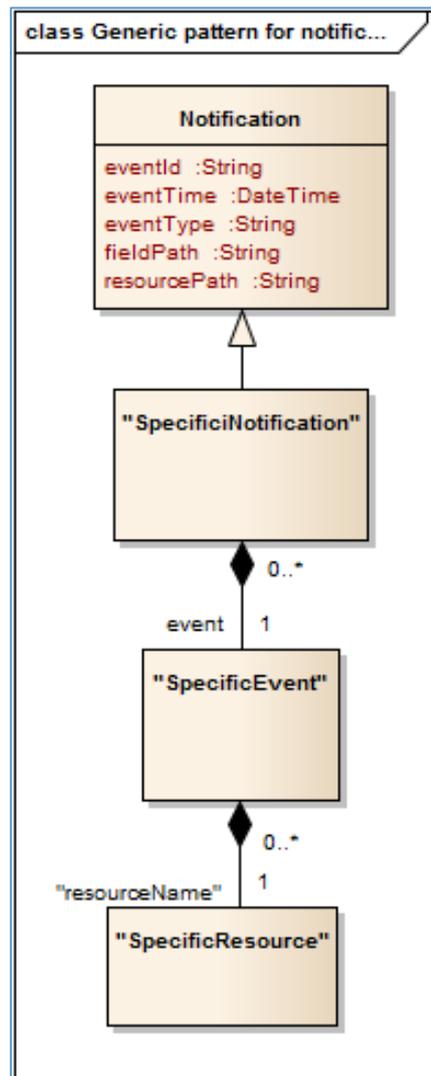
Notifications related to ServiceOrder:

- ServiceOrderCreationNotification
- ServiceOrderAttributeValueChangeNotification
- ServiceOrderStateChangeNotification
- ServiceOrderInformationRequiredNotification
- ServiceOrderRemoveNotification

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 ORDER CREATION NOTIFICATION

Notification sent when a new ServiceOrder resource is created.

Json representation sample

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

```

{
  "eventId":"00001",
  "eventTime":"2016-11-16T16:42:25-04:00",
  "eventType":"ServiceOrderCreationNotification",
  "event": {
    "ServiceOrder" :
      {-- SEE ServiceOrder RESOURCE SAMPLE --}
  }
}

```

SERVICE ORDER ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when changing an attribute of a ServiceOrder resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2016-11-16T16:42:25-04:00",
  "eventType":"ServiceOrderAttributeValueChangeNotification",
  "event": {
    "ServiceOrder" :
      [-- SEE ServiceOrder RESOURCE SAMPLE --]
  }
}
```

SERVICE ORDER STATE CHANGE NOTIFICATION

Notification sent when changing the state of a ServiceOrder resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2016-11-16T16:42:25-04:00",
  "eventType":"ServiceOrderStateChangeNotification",
  "event": {
    "ServiceOrder" :
      [-- SEE ServiceOrder RESOURCE SAMPLE --]
  }
}
```

SERVICE ORDER REMOVE NOTIFICATION

Notification sent when removing a ServiceOrder resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2016-11-16T16:42:25-04:00",
  "eventType":"ServiceOrderRemoveNotification",
  "event": {
    "ServiceOrder" :
      [-- SEE ServiceOrder RESOURCE SAMPLE --]
  }
}
```

SERVICE ORDER INFORMATION REQUIRED NOTIFICATION

Used to notify that some data in the order needs to be filled / is missing.

- “resourcePath” allows to precise if it is a data at order level or at orderItem level (and which one of them) that is missing
- “fieldPath” details which field is missing. Its structure is quite similar to GET filter criteria :
 - o “missing=” points at the missing field
 - o “&<criteria>” can be used to identify a specific element in lists

Simple example : notification contact is missing

```
{
  "eventId":"00005",
  "eventTime":"2013-04-19T16:42:25-30:00",
  "eventType":"orderInformationRequiredNotification",
  "resourcePath":"/order/42 ",
  "fieldPath":"missing=notificationContact",
  "serviceOrder":{
    "id":" 42",
    "href":"http://serverlocation:port/orderManagement/serviceOrder/42",
    "externalId":"NiceNameForTheConsumer_42"
  }
}
```

Complex example: in the order item “1”, the IMEI characteristic value is missing to instantiate the Service “465665”

```
{
  "eventId":"00006",
  "eventTime":"2013-04-19T16:42:25-30:00",
  "eventType":"orderInformationRequiredNotification",
  "resourcePath":"/order/42/orderItem/1",
  "fieldPath":"missing=service.serviceCharacteristic.value&service.id=465665&service.serviceCharacteristic.name=IMEI",
  "serviceOrder":{
    "id":" 42",
```

```
"href":"http://serverlocation:port/serviceOrderingManagement/serviceOrder/42",  
"externalId":"NiceNameForTheSubscriber_42"  
}  
}
```

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 ORDER

LIST SERVICE ORDERS

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

Description

This operation list service order 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 all the service orders for a given customer that were completed before a specified date.

Request
<pre>GET /serviceOrderingManagement/serviceOrder?relatedParty.role=subscriber&relatedParty.id=345221&completionDate.lt=2013-09-10 00:00:00ZGMT+1 Accept: application/json</pre>
Response
<pre>200 Content-Type : application/json [{ "id":"42", "href":"http://serverlocation:port/orderManagement/serviceOrder/42", "externalId":"NiceNameForTheConsumer_42", "priority":"1", "description":"A wonderful 42 order for brand new Services", "category":"Broadband", "state":"InProgress", "orderDate":"2013-04-12T16:42:23-04:00", "completionDate":"2013-04-19T16:42:23-04:00", "requestedStartDate":"2013-04-12", "requestedCompletionDate":"2013-04-19", "expectedCompletionDate":"2013-04-19", "notificationContact":"emailAddr-myAddress@hotmail.com", "note":[{ "text":"A free text detailing the note", "date":"2013-04-12T16:42:23-04:00", "author":"name"</pre>

```

    }
  ],
  "relatedParty":[
    {
      "role":"requester",
      "id":"345231",
      "name":"John Doe"
    },
    {
      "role":"fulfiller",
      "href":"http://serverlocation:port/provider/42"
    }
  ],
  "orderItem":[
    {
      "id":"1",
      "action":"add",
      "state":"Acknowledged",
      "appointment":{
        "id":"89",
        "href":"http://www.doodle.com/1WCV5647438"
      },
      "serviceSpecification": {
        "id":"42",
        "href":"http://serverlocation:port/catalogManagement/serviceSpecification/42"
      },
      "service": {
        "place": [
          {
            "href":"http://map.google.com/.../1234112GDE",
            "role":"DeliveryPlace"
          }
        ],
        "serviceCharacteristic":[
          {
            "name":"Colour",
            "value":"White"
          },
          {
            "name":"Memory",
            "value":"16"
          }
        ]
      }
    }
  ],
  {
    "id":"2",
    "action":"modify",
    "state":"InProgress",
    "serviceSpecification":{
      "id":"43",
      "href":"http://serverlocation:port/catalogManagement/serviceSpecification/43"
    },
    "service":{

```

```

    "id": "456",
    "href": "http://serverlocation:port/inventoryManagement/service/456",
    "serviceCharacteristic": [
      {
        "name": "anotherCharacteristic",
        "value": "itsNewValue"
      }
    ],
    "relatedParty": [
      {
        "role": "user",
        "id": "5667443",
        "name": "Jimmy Doe"
      }
    ]
  },
  {
    "id": "3",
    "action": "add",
    "state": "InProgress",
    "serviceSpecification": {
      "id": "51",
      "href": "http://serverlocation:port/catalogManagement/serviceSpecification/51"
    },
    "service": {
      "serviceRelationship": [
        {
          "type": "reliesOn",
          "service": {
            "id": "511",
            "href": "http://serverlocation:port/inventoryManagement/service/511"
          }
        }
      ]
    }
  },
  {
    "id": "4",
    "action": "modify",
    "state": "InProgress",
    "serviceSpecification": {
      "id": "89",
      "href": "http://serverlocation:port/catalogManagement/serviceSpecification/89"
    },
    "service": {
      "id": "120",
      "href": "http://serverlocation:port/inventoryManagement/service/120",
      "serviceState": "Inactive"
    }
  }
]
}
]

```

RETRIEVE SERVICE ORDER

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

Description

This operation retrieves a service order 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 a sample of a request for retrieving a ServiceOrder resource based on its id.

Request
<pre>GET /serviceOrderingManagement/serviceOrder/42 Accept: application/json</pre>
Response
<pre>200 Content-Type : application/json { "id":"42", "href":"http://serverlocation:port/orderManagement/serviceOrder/42", "externalId":"NiceNameForTheConsumer_42", "priority":"1", "description":"A wonderful 42 order for brand new Services", "category":"Broadband", "state":"InProgress", "orderDate":"2013-04-12T16:42:23-04:00", "completionDate":"2013-04-19T16:42:23-04:00", "requestedStartDate":"2013-04-12", "requestedCompletionDate":"2013-04-19", "expectedCompletionDate":"2013-04-19", "notificationContact":"emailAddr-myAddress@hotmail.com", "note":[{ "text":"A free text detailing the note", "date":"2013-04-12T16:42:23-04:00", "author":"name" }], }</pre>

```

"relatedParty":[
  {
    "role":"requester",
    "id":"345231",
    "name":"John Doe"
  },
  {
    "role":"fulfiller",
    "href":"http://serverlocation:port/provider/42"
  }
],
"orderItem":[
  {
    "id":"1",
    "action":"add",
    "state":"Acknowledged",
    "appointment":{
      "id":"89",
      "href":"http://www.doodle.com/1WCV5647438"
    },
    "serviceSpecification": {
      "id":"42",
      "href":"http://serverlocation:port/catalogManagement/serviceSpecification/42"
    },
    "service": {
      "place": [
        {
          "href":"http://map.google.com/.../1234112GDE",
          "role":"DeliveryPlace"
        }
      ],
      "serviceCharacteristic":[
        {
          "name":"Colour",
          "value":"White"
        },
        {
          "name":"Memory",
          "value":"16"
        }
      ]
    }
  },
  {
    "id":"2",
    "action":"modify",
    "state":"InProgress",
    "serviceSpecification":{
      "id":"43",
      "href":"http://serverlocation:port/catalogManagement/serviceSpecification/43"
    },
    "service":{
      "id":"456",
      "href":"http://serverlocation:port/inventoryManagement/service/456",

```

```
"serviceCharacteristic":[
  {
    "name":"anotherCharacteristic",
    "value":"itsNewValue"
  }
],
"relatedParty":[
  {
    "role":"user",
    "id":"5667443",
    "name":"Jimmy Doe"
  }
]
},
{
  "id":"3",
  "action":"add",
  "state":"InProgress",
  "serviceSpecification":{
    "id":"51",
    "href":"http://serverlocation:port/catalogManagement/serviceSpecification/51"
  },
  "service":{
    "serviceRelationship":[
      {
        "type":"reliesOn",
        "service":{
          "id":"511",
          "href":"http://serverlocation:port/inventoryManagement/service/511"
        }
      }
    ]
  }
},
{
  "id":"4",
  "action":"modify",
  "state":"InProgress",
  "serviceSpecification": {
    "id":"89",
    "href":"http://serverlocation:port/catalogManagement/serviceSpecification/89"
  },
  "service":{
    "id":"120",
    "href":"http://serverlocation:port/inventoryManagement/service/120",
    "serviceState":"Inactive"
  }
}
]
```

CREATE SERVICE ORDER

POST /serviceOrder

Description

This operation creates a service order entity.

Mandatory and Non Mandatory Attributes

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

POST should be used without specifying the id and the href, the Service Order Management system is in charge of generating the id + href for the ServiceOrder.

Mandatory Attributes	Rule
relatedParty	
orderItem.id	
orderItem.action	
orderItem.service	

Non Mandatory Attributes	Default Value	Rule
externalId	N/A	
state	Acknowledged	
priority	4 (lowest)	
description	N/A	
category	"uncategorized"	
requestedStartDate	No specified default value	
requestedCompletionDate	No specified default value	
notificationContact	No specified default value	
note	No specified default value	
relatedParty		They must be at least a defined party linked to the order (Customer, ...) Related party could have following role: owner, requester, fulfiller
orderItem.state	Acknowledged	
orderItem.ServiceSpecification		The serviceSpecification may not be useful when doing a "modify" or "delete" on an owned Service
orderItem.place		
orderItem.appointment		

Additional Rules

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

Context	Mandatory Sub-Attributes
POST Mandatory attributes within Service Order	<ul style="list-style-type: none"> ○ Within “note” <ul style="list-style-type: none"> ▪ text ○ Within “relatedParty” <ul style="list-style-type: none"> ▪ id AND/OR href AND/OR name ▪ role
POST Mandatory attributes within Service Order item	<ul style="list-style-type: none"> ○ Within “place” <ul style="list-style-type: none"> ▪ role ▪ id AND/OR href ○ Within “serviceSpecification” <ul style="list-style-type: none"> ▪ id AND/OR href ○ Within “service” <ul style="list-style-type: none"> ▪ Depends on the order item “action”, cf. rule below
POST Mandatory attributes within Service (in each order item)	<ul style="list-style-type: none"> ○ If orderItem.action = “add” : <ul style="list-style-type: none"> ▪ serviceCharacteristics ○ If orderItem.action = “modify” <ul style="list-style-type: none"> ▪ id AND/OR href ○ If orderItem.action = “delete” <ul style="list-style-type: none"> ▪ id AND/OR href

Default Values Summary

When creating the resource, the following table summarizes the default values applicable to optional attributes of the resource (or sub-resources).

Attributes	Default Value
state	Acknowledged
priority	4 (lowest)
category	Uncategorized
orderItem.state	Acknowledged

Usage Samples

Here's a sample of a request for creating a ServiceOrder resource.

Request
POST /serviceOrderingManagement/serviceOrder

```
Content-Type: application/json
{
  "note":[
    {
      "text":"A free text detailing the note",
      "date":"2013-04-12T16:42:23-04:00",
      "author":"name"
    }
  ],
  "relatedParty":[
    {
      "role":"subscriber",
      "id":"345231",
      "name":"John Doe"
    },
    {
      "role":"provider",
      "href":"http://serverlocation:port/provider/4563"
    }
  ],
  "orderItem":[
    {
      "id":"1",
      "action":"add",
      "serviceSpecification": {
        "id":"42",
        "href":"http://serverlocation:port/catalogManagement/serviceSpecification/42"
      },
      "service": {
        "serviceCharacteristic":[
          {
            "name":"Colour",
            "value":"White"
          },
          {
            "name":"Memory",
            "value":"16"
          }
        ]
      }
    }
  ],
  {
    "id":"2",
    "action":"modify",
    "service":{
      "id":"456",
      "href":"http://serverlocation:port/inventoryManagement/service/456",
      "relatedParty":[
        {
          "role":"user",
          "id":"5667443",
          "name":"Jimmy Doe"
        }
      ]
    }
  ]
}
```

```

}
},
{
  "id":"3",
  "action":"delete",
  "serviceSpecification":{
    "id":"456",
    "href":"http://serverlocation:port/catalogManagement/serviceSpecification/456"
  }
}
]
}

```

Response

201

Content-Type: application/json

```

{
  "id":"42",
  "href":"http://serverlocation:port/orderManagement/serviceOrder/42",
  "priority":"4",
  "description":"A wonderful 42 order for brand new Services",
  "category":"Uncategorized",
  "state":"Acknowledged",
  "orderDate":"2013-04-12T16:42:23-04:00",
  "expectedCompletionDate":"2013-04-19",
  "note":[
    {
      "text":"A free text detailing the note",
      "date":"2013-04-12T16:42:23-04:00",
      "author":"name"
    }
  ],
  "relatedParty":[
    {
      "role":"subscriber",
      "id":"345231",
      "name":"John Doe"
    },
    {
      "role":"provider",
      "href":"http://serverlocation:port/provider/4563"
    }
  ],
  "orderItem":[
    {
      "id":"1",
      "action":"add",
      "state":"Acknowledged",
      "serviceSpecification": {
        "id":"42",
        "href":"http://serverlocation:port/catalogManagement/serviceSpecification/42"
      }
    }
  ]
}

```

```

    },
    "service": {
      "serviceCharacteristic":[
        {
          "name":"Colour",
          "value":"White"
        },
        {
          "name":"Memory",
          "value":"16"
        }
      ]
    }
  },
  {
    "id":"2",
    "action":"modify",
    "state":"Acknowledged",
    "service":{
      "id":"456",
      "href":"http://serverlocation:port/inventoryManagement/service/456",
      "relatedParty":[
        {
          "role":"user",
          "id":"5667443",
          "name":"Jimmy Doe"
        }
      ]
    }
  },
  {
    "id":"3",
    "action":"delete",
    "state":"Acknowledged",
    "serviceSpecification":{
      "id":"456",
      "href":"http://serverlocation:port/catalogManagement/serviceSpecification/456"
    }
  }
]
}

```

PATCH SERVICE ORDER

PATCH /serviceOrder/{id}

Description

This operation allows partial updates of a service order 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
state	To manage the order delivery process : Acknowledged, InProgress (start process), Held / Pending (suspend process), Cancelled Influence the orderItem states
priority	
category	
requestedStartDate	Only when order is in "Acknowledged" state – delivery process not started
requestedCompletionDate	Only when order is in "Acknowledged" state – delivery process not started
expectedCompletionDate	
notificationContact	
note	
relatedParty	Only when order is in "Acknowledged" state – delivery process not started
orderItem.state	To manage the order delivery process : Acknowledged, InProgress (start process), Held / Pending (suspend process), Cancelled Influence the orderItem states
orderItem.serviceSpecification	Only when order is in "Acknowledged" state – delivery process not started
orderItem.service	Only when order is in "Acknowledged" state – delivery process not started or suspended
orderItem.place	Only when order is in "Acknowledged" state – delivery process not started or suspended
orderItem.appointment	Only when order is in "Acknowledged" state – delivery process not started or suspended

Non Patchable Attributes	Rule
id	
href	
externalId	
orderDate	
completionDate	
orderItem.id	
orderItem.action	

Additional Rules

The following pre-conditions apply for this operation.

Pre-conditions
<p>When patching an order state to :</p> <ul style="list-style-type: none"> ○ “Pending” / “Held” : all “InProgress” order items SHOULD be set to “Pending” / “Held” ○ “InProgress” : <ul style="list-style-type: none"> ○ all “Acknowledged” order items MUST be set to “InProgress” ○ all “Pending”/“Held” order items MAY be set to “InProgress” <p>When patching an order item state to :</p> <ul style="list-style-type: none"> ○ “Pending” / “Held” : the order state itself SHOULD be set to “Pending” / “Held”

Usage Samples

Here's an example of a request for patching a ServiceOrder resource : resume order (Suspend would be the same request using the “Pending” state).

Request
<pre>PATCH /serviceOrderingManagement/serviceOrder/42 Content-Type: application/merge-patch+json { "state": "InProgress" }</pre>
Response
<pre>201 { Similar JSON as in GET response with state changed }</pre>

DELETE SERVICE ORDER

DELETE /serviceOrder/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service order entity.

Usage Samples

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

Request
DELETE /serviceOrderingManagement/serviceOrder/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
<pre>POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}</pre>
Response
<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.

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.

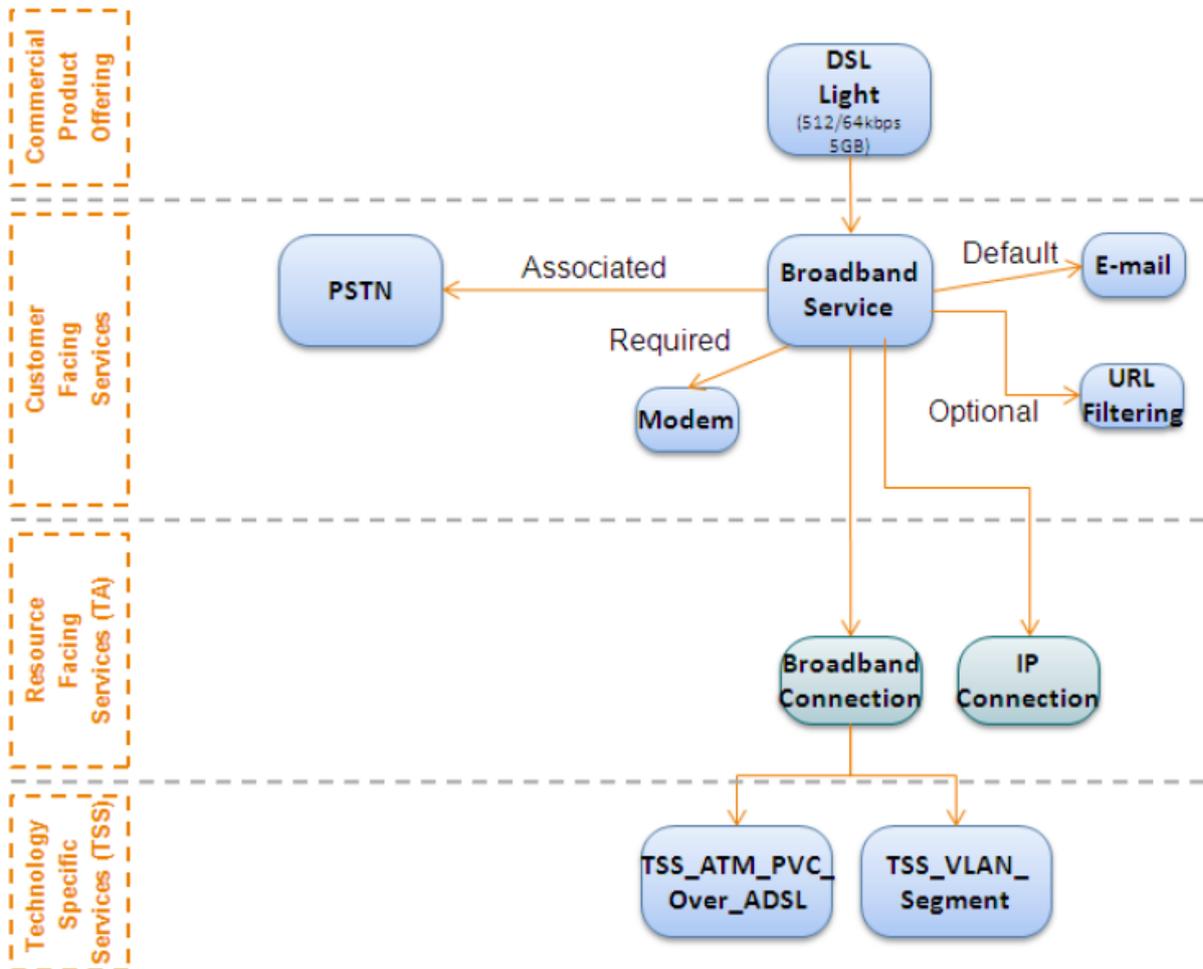
ACKNOWLEDGEMENTS

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	24/10/2016	Pierre Gauthier TM Forum pgauthier@tmforum.org Ludovic Robert Orange ludovic.robert@orange.com Nicoleta Stoica Vofadone nicoleta.stoica@vodafone.com Jean-Luc Tymen Orange jeanluc.tymen@orange.com	

SERVICE ORDER EXAMPLE

BROADBAND SERVICE EXAMPLE



CREATE BROADBAND SERVICE (CFS)

REQUEST - An example service order composed of 1 order line (orderItem).

This request would be used for the subsequent creation of the following CFSs:

- PSTN
- EMAIL
- URL Filtering

```

POST /serviceOrderingManagement/serviceOrder
Content-type: application/json
{
  "priority": "1",
  "description": "An order for a Broadband service",
  "category": "CFS",
  "requestedStartDate": "2016-10-12",
  "requestedCompletionDate": "2016-10-19",
  "orderRelationship": [
    {
      "role": "cross-ref",
      "id": "producOrderxx",
      "href": "http://serverlocation:port/productOrderingManagement/productOrder/productOrderxx"
    }
  ],
  "relatedParty": [
    {
      "role": "Service Provider",
      "href": "http://serverlocation:port/partyManagement/party/42"
    }
  ],
  "orderItem": [
    {
      "id": "1",
      "action": "add",
      "serviceSpecification": {
        "id": "BBserviceSpecID",
        "href": "http://serverlocation:port/catalogManagement/serviceSpecification/BBServiceSpecID"
      },
      "service": {
        "place": {
          "role": "installation address",
          "href": "http://serverlocation:port/addressManagement/address/addressID"
        },
        "serviceCharacteristic": [{
          "name": "bandwith",
          "value": "10"
        }]
      }
    }
  ]
}

```

RESPONSE

```

201
Content-type: application/json
{
  "id": "42",
  "href": "http://serverlocation:port/serviceOrderingManagement/serviceOrder/42",
  "state": "Acknowledged",
  "description": "An order for a Broadband service",
  "category": "CFS",
  "requestedStartDate": "2016-10-12T16:42:23-04:00",
  "requestedCompletionDate": "2016-10-19T16:42:23-04:00",
  "orderDate": "2016-10-19T16:42:23-04:10",
  "orderRelationship": [
    {
      "role": "cross-ref",
      "id": "productOrderxx",
      "href": "http://serverlocation:port/productOrderingManagement/productOrder/productOrderxx"
    }
  ],
  "relatedParty": [
    {
      "role": "Service Provider",
      "href": "http://serverlocation:port/partyManagement/party/42"
    }
  ],
  "orderItem": [
    {
      "id": "1",
      "action": "add",
      "state": "Acknowledged",
      "serviceSpecification": {
        "id": "BBserviceSpecID",
        "href": "http://serverlocation:port/catalogManagement/serviceSpecification/BBServiceSpecID"
      },
      "service": {
        "place": {
          "role": "installation address",
          "href": "http://serverlocation:port/addressManagement/address/addressID"
        },
        "serviceCharacteristic": [{
          "name": "bandwidth",
          "value": "10"
        }
      ]
    }
  ]
}

```



```
}  
}  
}  
}
```