Frameworx Specification

Service Ordering Management API REST Specification

TMF641
Release 18.0.0
February 2018

Latest Update: TM Forum Release 18.0	**Member Evaluation**
Version 3.0.4	**IPR Mode: RAND**

NOTICE

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. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

LIST OF FIGURES

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 describe a group of operations on service – one service order item per service. An action at the level of the service order item describe the operation to be done on a service (add, changer, terminate for example). The service order is triggered from the BSS system in charge of the product order management to the SOM (Service order Management) system that will manage the service fulfillment.

Resource model

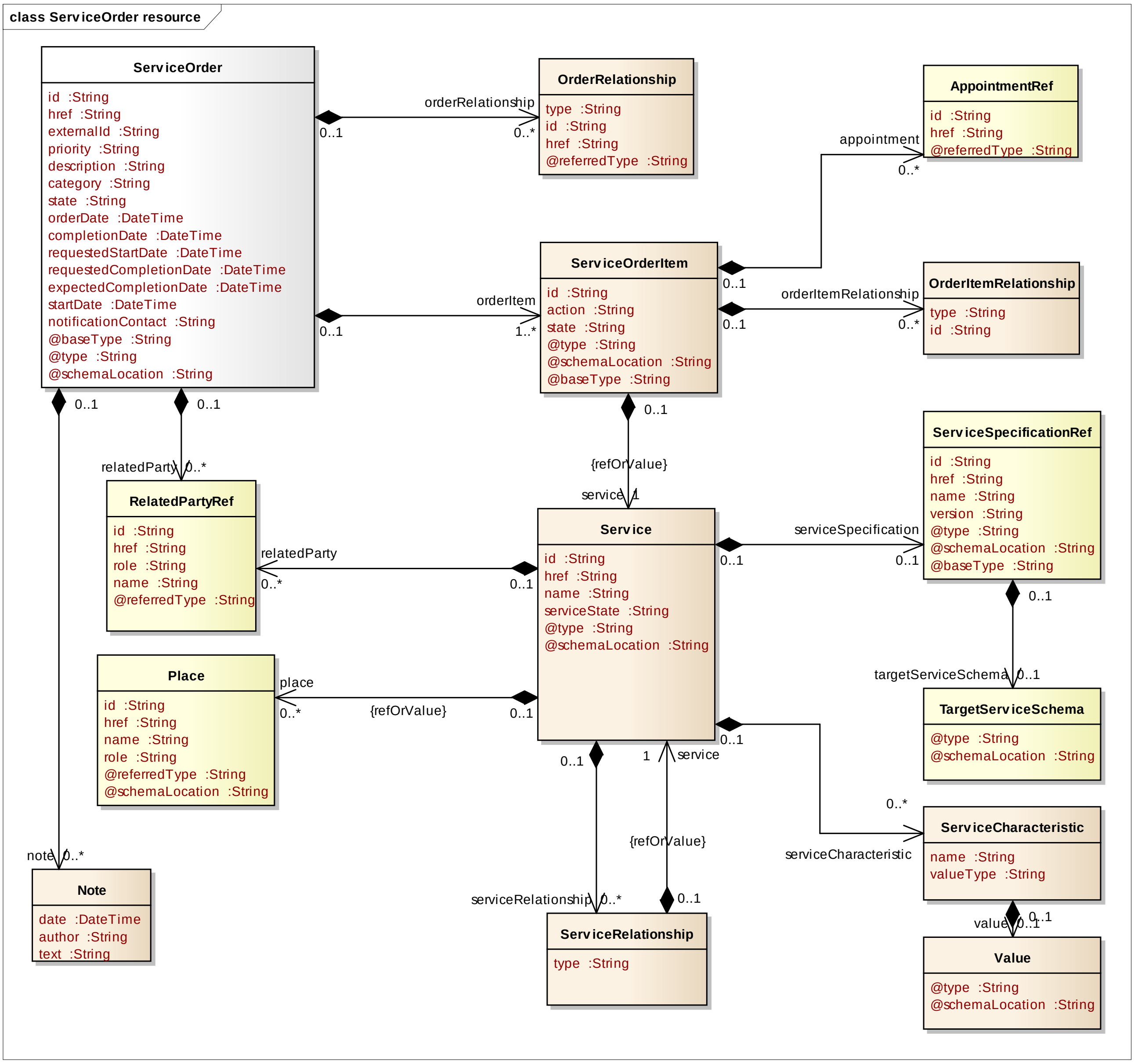


Figure 1 - ServiceOrder resource model

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 except 'Partial' status which is not available for service order item.

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

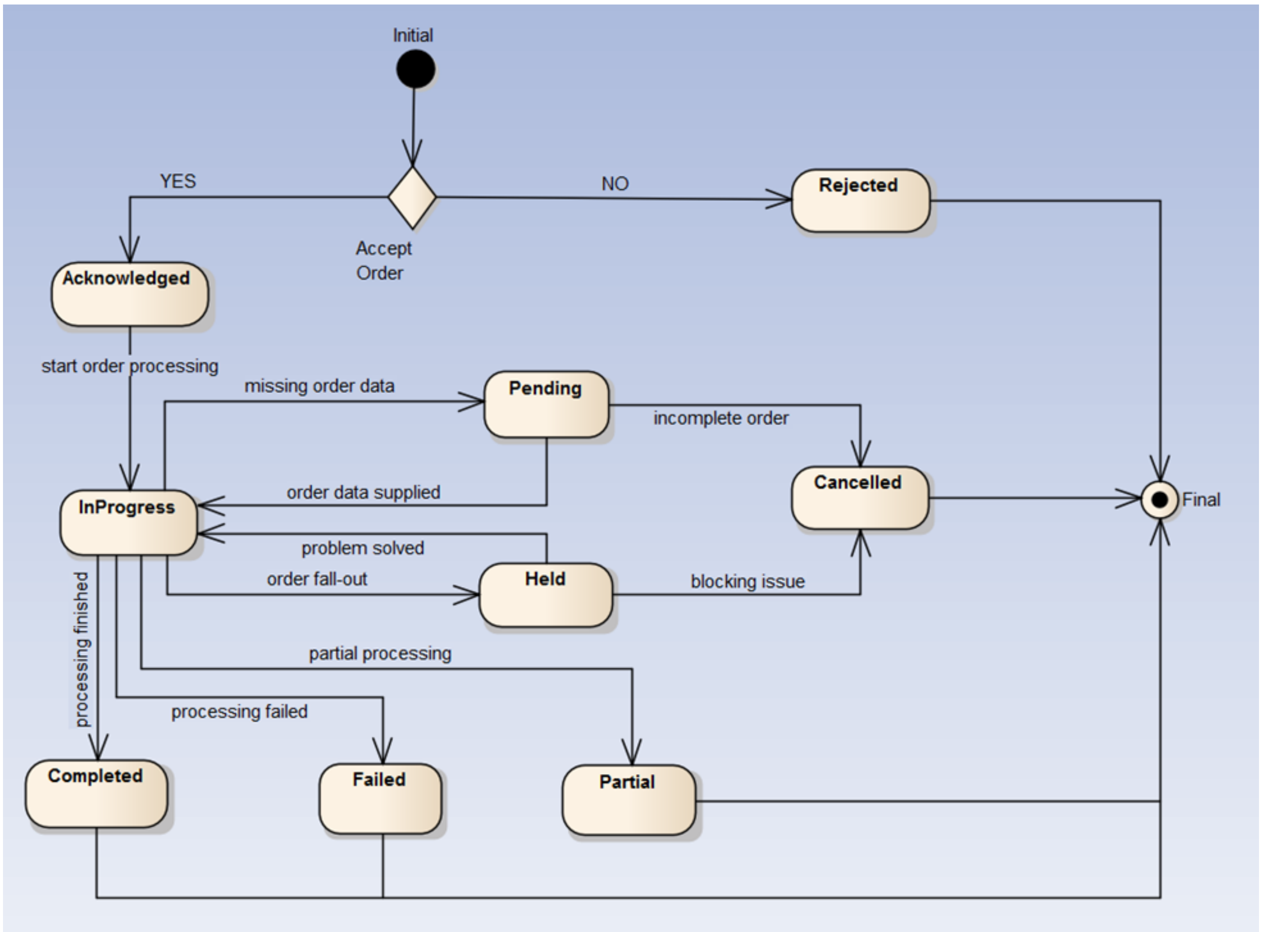


Figure 2 - Order and Order Item State Model

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	The Rejected state is where:

	- 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	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. 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.
Held	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.
Failed	All Order items have failed which results in the entire Order has Failed.
Partial	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

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
Pending	All ‘Pending state
Held	All ‘Held’ state
Cancelled	All ‘Cancelled’
Partial	All Order item are in final state (either ‘Failed’ or ‘Completed’ or ‘Cancelled’) At least one item has ‘Failed’ state At least one item has ‘Completed’ state
Failed	All ‘Failed’ or ‘Cancelled’ At least one item has ‘Failed’ state
Completed	All ‘Completed’ or ‘Cancelled’ At least one item has ‘Completed’ state

Service state model:

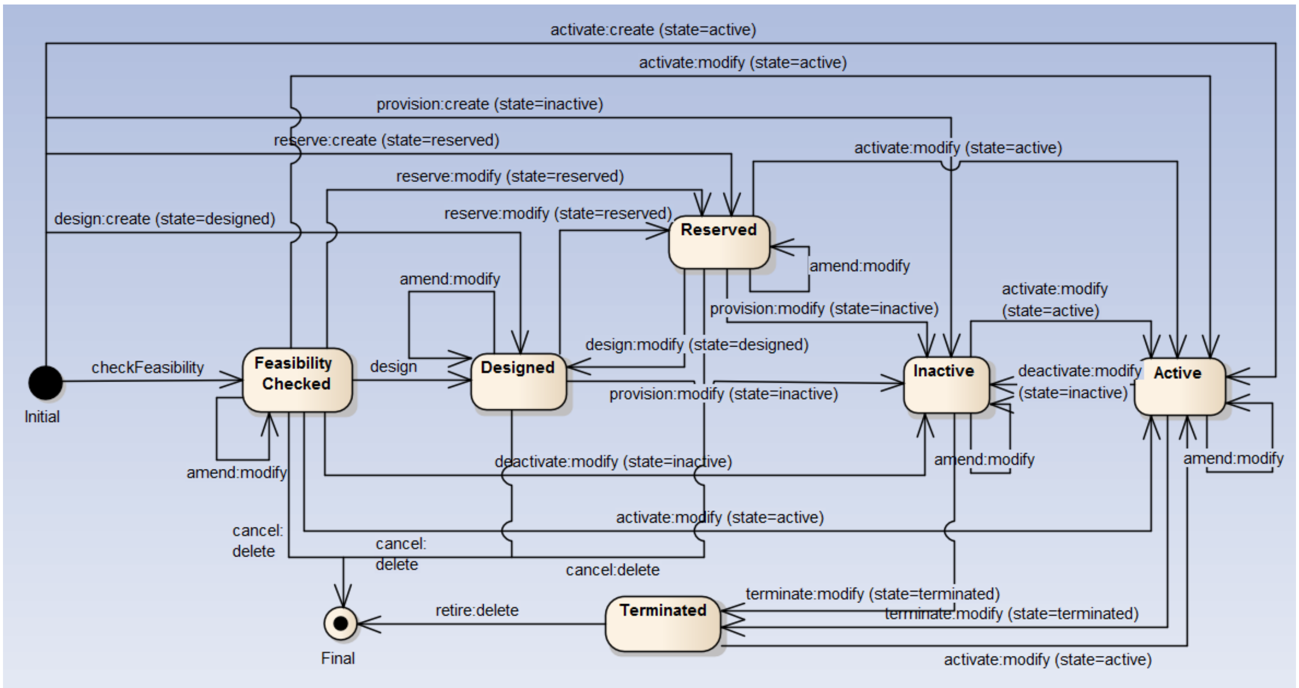

activate:create (state=active)
activate:modify (state=active)
provision:create (state=inactive)
reserve:create (state=reserved)
activate:modify (state=active)
reserve:modify (state=reserved)
design:create (state=designed)
reserve:modify (state=reserved)
Reserved
amend:modify
amend:modify
provision:modify (state=inactive)
activate:modify
(state=active)
design:modify (state=designed)
checkFeasibility
Feasibility
Checked
design
Designed
Inactive
deactivate:modify
(state=inactive)
Active
provision:modify (state=inactive)
Initial
amend:modify
deactivate:modify (state=inactive)
amend:modify
amend:modify
activate:modify (state=active)
cancel:
delete
cancel:
delete
cancel:delete
terminate:modify (state=terminated)
terminate:modify (state=terminated)
retire:delete
Terminated
Final
activate:modify (state=active)

Figure 3 - Service State Model

Note: ‘Feasability 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	*not relevant…service removed*
Reserved	delete	*not relevant…service removed*
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	*not relevant…service removed*

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

Field descriptions

ServiceOrder

Field	Description
id	A string. ID created on repository side
href	A string. Hyperlink to access the order
externalId	A string. ID given by the order requester and only understandable by him (to facilitate his searches afterwards). It could an order number reference from the triggering system (OM)
priority	A string. A way that can be used by order requester to prioritize orders in Service Order Management system (from 0 to 4: 0 is the highest priority, and 4 the lowest). It could be for example valued by BSS based on customer order requested priority.
description	A string. A free-text description of the service order.
category	A string. Used to categorize the order - Requester valued it and this category can be useful for the Service Order Management system and/or requester (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 (DateTime). Order start date wished by the requestor
requestedCompletionDate	A date (DateTime). Requested delivery date from the requestor perspective.
expectedCompletionDate	A date (DateTime). Expected delivery date amended by the provider
startDate	A date (DateTime). Date when the order fulfillment start – valued by server side (not requester)
notificationContact	A string. Contact attached to the order to send back information regarding this order
@baseType	A string. Indicates the base type of the resource. Here it should be valued to 'ServiceOrder'.
@type	A string. Indicates the type of resource.
@schemaLocation	A string. A link to the schema describing this REST resource.
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).

Field	Description
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.

Field	Description
href	A string. An hyperlink to the related order.
id	A string. The id of the related order.
type	A string. The type of the relationship to the related order, as examples, it 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)
@referredType	A string. The type of the referred order

ServiceOrderItem

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

Field	Description
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
state	A string. State of the order item: described in the state machine diagram. This is the requested state.
@baseType	A string. Indicates the base type of the sub resource. Here it should be valued to 'ServiceOrderItem'.
@type	A string. Indicates the type of the sub resource.
@schemaLocation	A string. A link to the schema describing this REST subresource.
appointment	An appointment references (AppointmentRef). Used to precise that an appointment was set up with a related party for this order item.
orderItemRelationship	A list of order items relationships (ServiceOrderItemRelationship[*]). Linked order items to the one containing this attribute.
service	A service reference (Service). The Service to be acted on by the order item.

OrderItemRelationship

Linked order item to the one containing this attribute.

Field	Description
type	A string. The type of relation to another order item, as example it can be: o “dependency” if the order item needs to be “not started” until another order item is complete o "reliesOn" if the service needs another service to rely on o ...
id	A service order item id - It could be any order item from this order.

Appointment

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

Field	Description
id	A string. The id of the appointment.
href	A string. An hyperlink to the appointment
@referredType	A string. The type of the referred appointment

Place

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

Field	Description
id	A string. ID of an existing 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).
@referredType	A string. The type of the referred place
@schemaLocation	A string. The URL to get the resource schema

Service

Service attributes description. The service could referred or valued(described) in this structure.

Field	Description
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).
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 define places useful for the service (for example a delivery geographical place).
@type	A string. The type of the service
@schemaLocation	A string. The URL to get the resource schema
serviceSpecification	A service specification (ServiceSpecificationRef). The service specification (default values, etc. are fetched from the catalogue).
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 :

	• “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 CSP internal Administrator or Owner for example).

RelatedPartyRef

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

Field	Description
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.
@referredType	A string. The type of the referred party

ServiceRelationship

Service relationship describe relationship with the service described in this service order item with existing or described service

Field	Description
type	A string. The type of related order item, can be: o “dependency” if the order item needs to be “not started” until another order item is complete o - "reliesOn" if the service needs another already owned service to rely on o …
service	A service reference (Service).

ServiceCharacteristic

Characteristics of the service that is instantiated or to modified.

Field	Description
name	A string. Name of the characteristic.
valueType	A string. ValueType of the characteristic. Could a simple attribute or a complex one as a table or an array.
value	An object. A link to a value description structure.

Value

Characteristic value of the service that is instantiated or modified by the service order

Field	Description
@type	A string. Indicates the (class) type of resource.

@schemaLocation	A string. This field provided a link to the schema describing this REST resource.

ServiceSpecificationRef relationship

Service specification reference is a class that offers characteristics to describe a type of service (CFS or RFS).

Field	Description
id	A string. Unique identifier of the service specification.
href	A string. Reference of the service specification.
version	A string. Version of the service specification.
name	A string. Name of the service specification.
@type	A string. Indicates the (class) type of resource (here service specification).
@schemaLocation	A string. URL to get to the schema description file.
@baseType	A string. Indicates the base type of the sub resource. Here it should be valued to 'ServiceSpecification'.

TargetServiceSchema sub-resource

Target to the schema describing the service spec resource (and type)

Field	Description
@type	A string. Indicates the (class) type of resource.
@schemaLocation	A string. This field provided a link to the schema describing this REST resource.

Json representation sample

We provide below the json representation of a sample of a ServiceOrder resource composed of 4 order lines (orderItems):

- Line “1”: Ordering of a vCPE (this item illustrates targetServiceSchema use by bringing ‘invariantUUID’ and ‘toscalModelURL’ attributes from the service schema).
- Line “2” : Generic example of modification of a characteristic value of an already owned Service,
- Line “3” : Generic example of ordering of a new simple Service that needs (is supported by) another already owned Service ()
- Line “4”: Generic example of modification of an already owned Service to switch its state to ‘inactive’.

```
{
  "id":"42",
  "href":"http://serverlocation:port/orderManagement/serviceOrder/42",
  "externalId":"OrangeBSS747",
  "priority":"1",
  "description":"Service order description",
  "category":"TMF resource illustration",
  "state":"acknowledged",
  "orderDate":"2018-01-12T09:37:40.508Z",
  "completionDate":"",
  "requestedStartDate":"2018-01-15T09:37:40.508Z",
  "requestedCompletionDate":"2018-01-15T09:37:40.508Z",
  "expectedCompletionDate":"2018-01-15T09:37:40.508Z",
  "startDate":"2018-01-12T09:37:40.508Z",
  "@type":"standard",
  "note":{
    "date":"2018-01-15T09:37:40.508Z",
    "author":"Jean Pontus",
    "text":"bla bla bla"
  },
  "relatedParty":[
    {
      "id":"456",
      "href":"http://serverlocation:port/partyManagement/party/456",
      "role":"requester"
    }
  ],
  "orderItem":[
    {
      "id":"1",
      "action":"add",
      "state":"acknowledged",
      "@type":"standard",
      "service":{
        "serviceState":"Active",
        "serviceCharacteristic":[
          {
            "name":"vCPE_IP",
            "valueType":"String",
            "value":{
              "@type":"IPAddress",
              "@schemaLocation":"http....",
              "vCPE_IP":"193.218.236.21"
            }
          }
        ],
        "serviceSpecification":{
          "id":"12",
          "href":"http://...:serviceSpecification/12",
          "name":"vCPE",
          "version":"1",
          "targetServiceSchema":{
            "@type":"ONAPServiceSpec",
            "@schemaLocation":"http//..."
          },
          "invariantUUID":"456-852-357",
          "toscaModelURL":"http//...",
          "@type":"vCPE",
          "@schemaLocation":"http..."
        }
      }
    },
    {
      "id":"2",
      "action":"modify",
      "state":"acknowledged",
      "@type":"standard",
      "service":{
        "id":"456",
        "href":"http://serverlocation:port/inventoryManagement/service/456",
        "serviceState":"Active",
        "serviceCharacteristic":[
          {
            "name":"Characteristic1",
            "valueType":"string",
            "value":{
              "@type":"string",
              "value":"newValue"
            }
          }
        ]
      }
    },
    {
      "id":"3",
      "action":"add",
      "state":"acknowledged",
      "@type":"standard",
      "service":{
        "serviceState":"Active",
        "serviceRelationship":[
          {
            "type":"reliesOn",
            "id":"45"
          }
        ],
        "serviceSpecification":{
          "id":"48",
          "href":"http//..../48",
          "name":"genericService48",
          "version":"2"
        }
      }
    },
    {
      "id":"4",
      "action":"modify",
      "state":"acknowledged",
      "@type":"standard",
      "service":{
        "id":"12",
        "href":"http://serverlocation:port/inventoryManagement/service/12",
        "serviceState":"inactive"
      }
    }
  ]
}
```

Notification Resource Models

Five notifications are defined for this API.

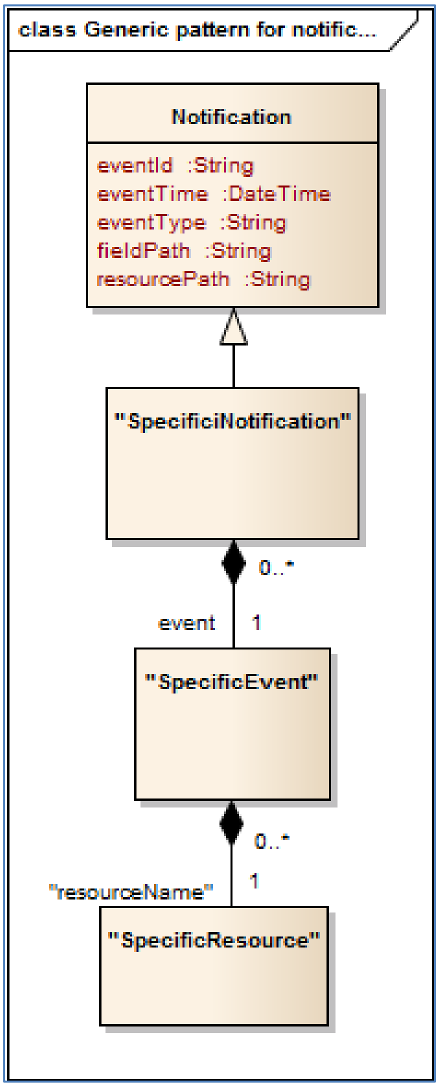
Notifications related to ServiceOrder:

- ServiceOrderCreationNotification
- ServiceOrderAttributeValueChangeNotification
- ServiceOrderStateChangeNotification
- ServiceOrderInformationRequiredNotification
- ServiceOrderRemoveNotification

The notification structure for all notifications in this API follows 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 occurence (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 :
    - "missing=" points at the missing field
    - "&<criteria>" can be used to identify a specific element in lists

<u>Simple example</u> : 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.

Providing filtering criteria is mandatory to avoid too many answers retrieved. See example below:

- externalId
- description
- orderDate
- state + orderDate interval (orderDate > 11/4/2018 & orderDate > 11/9/2017)
- priority + orderDate interval
- category + orderDate interval
- ...

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

```
GET /serviceOrderingManagement/v1/serviceOrder?state=completed&category= TMFresourceillustration
&completionDate.gt=2018-01-14T00:00:00.000Z
Accept: application/json
```

Response

```
200
Content-Type : application/json
[
{
  "id": "42",
  "href": "http://serverlocation:port/serviceOrderingManagement/serviceOrder/42",
  "externalId": "OrangeBSS747",
  "priority": "1",
  "description": "Service order description",
  "category": "TMFresourceillustration",
  "state": "completed",
  "orderDate": "2018-01-12T09:37:40.508Z",
  "completionDate": "2018-01-15T09:37:40.508Z",
  "requestedStartDate": "2018-01-15T09:37:40.508Z",
  "requestedCompletionDate": "2018-01-15T09:37:40.508Z",
  "expectedCompletionDate": "2018-01-15T09:37:40.508Z",
  "startDate": "2018-01-12T09:37:40.508Z",
  "@type": "standard",
  "note": {
   "date": "2018-01-15T09:37:40.508Z",
   "author": "Jean Pontus",
   "text": "bla bla bla"
  },
  "relatedParty": [
   {
    "id": "456",
    "href": "http://serverlocation:port/partyManagement/party/456",
    "role": "requester"
   }
  ],

  "orderItem": [
   {
    "id": "1",
    "action": "add",
    "state": "completed",
    "@type": "standard",
    "service": {
     "serviceState": "Active",
     "serviceCharacteristic": [
      {
       "name": "vCPE_IP",
       "valueType": "String",
       "value": {
        "@type": "IPAddress",
        "@schemaLocation": "http....",
                          "vCPE_IP": "193.218.236.21"
       }
      }
     ],
     "serviceSpecification": {
      "id": "12",
      "href": "http://...:serviceSpecification/12",
      "name": "vCPE",
      "version": "1",
      "@type": "vCPE",
      "@schemaLocation": "http..."
     }
    }
   }
  ]
}]
```

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

```
GET /serviceOrderingManagement/v1/serviceOrder/43
Accept: application/json
```

Response

```
200
Content-Type : application/json
{
  "id": "43",
  "href": "http://serverlocation:port/serviceOrderingManagement/serviceOrder/43",
  "externalId": "OrangeBSS748",
  "priority": "1",
  "description": "Service order description",
  "category": "TMF resource illustration",
  "state": "completed",
  "orderDate": "2018-01-12T09:37:40.508Z",
  "completionDate": "2018-01-15T09:37:40.508Z",
  "requestedStartDate": "2018-01-15T09:37:40.508Z",
  "requestedCompletionDate": "2018-01-15T09:37:40.508Z",
  "expectedCompletionDate": "2018-01-15T09:37:40.508Z",
  "startDate": "2018-01-12T09:37:40.508Z",
  "@type": "standard",
  "note": {
    "date": "2018-01-15T09:37:40.508Z",
    "author": "Jean Pontus",
    "text": "bla bla bla"
  },
  "relatedParty": [
    {
      "id": "456",
      "href": "http://serverlocation:port/partyManagement/party/456",
      "role": "requester"
    }
  ],
  "orderItem": [
    {
      "id": "1",
      "action": "add",
      "state": "completed",
      "@type": "standard",
      "service": {
       "serviceState": "Active",
       "serviceCharacteristic": [
        {
         "name": "vCPE_IP",
         "valueType": "String",
         "value": {
          "@type": "IPAddress",
          "@schemaLocation": "http....",
                             "vCPE_IP": "193.218.236.21"
         }
        }
       ],
       "serviceSpecification": {
        "id": "12",
        "href": "http://...:serviceSpecification/12",
        "name": "vCPE",
        "version": "1",
        "@type": "vCPE",
        "@schemaLocation": "http..."
       }
      }
    }
  ]
}
```

CREATE SERVICE ORDER

POST /serviceOrder

Description

This operation creates a service order entity.

Mandatory and Non Mandatory Attributes

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

POST request should be used without specifying following attributes (these attributes will be defined by server side):

- id
- href
- state
- orderDate

- completionDate (once service order completed)
- expectedCompletionDate
- startDate
- orderItem.state

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

Mandatory Attributes	Rule
orderItem	At least one orderItem must be provided
orderItem.id	
orderItem.action	
orderItem.service	

Following table describes additional rules to conditional mandatory attributes:

if OrderRelationship is provided	type and (id AND/OR href) are required
if AppointmentRef is provided	(id AND/OR href) is required
if OrderItemRelationship is provided	type and id are required
if RelatedPartyRef is provided	(id AND/OR href AND/OR name) and role are required
if Place is provided	(id AND/OR href) and role are required
if ServiceSpecificationRef is provided	(id AND/OR href) is required
if Note is provided	date, author and text are required
if ServiceRelationship is provided	type and service (ref or value) are required
if ServiceCharacteristic is provided	name, valueType and value are required
ServiceOrderItem.Service.id AND/OR ServiceOrderItem.Service.href	Mandatory if serviceOrderItem.action different from ‘add’

Non Mandatory Attributes	Default Value	Rule
externalId	No specified default value	
priority	4 (lowest)	
description	No specified default value	
category	No specified default value	
requestedStartDate	No specified default value	
requestedCompletionDate	No specified default value	

notificationContact	No specified default value	
note	Structure could be not present	
relatedParty	Structure could be not present	
orderItem.ServiceSpecification		The serviceSpecification may not be useful when doing a “modify” or “delete” on an owned Service
orderItem.place	Structure could be not present	
orderItem.appointment	Structure could be not present	
orderItem.orderItemRelationship	Structure could be not present	
orderRelationship	Structure could be not present	
ServiceOrderItem.Service.serviceState	active	

Usage Samples

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

Request

```
POST /serviceOrderingManagement/v1/serviceOrder
Content-Type: application/json
{
  "externalId": "OrangeBSS748",
  "priority": "1",
  "description": "Service order description",
  "category": "TMF resource illustration",
  "requestedStartDate": "2018-01-15T09:37:40.508Z",
  "requestedCompletionDate": "2018-01-15T09:37:40.508Z",
  "@type": "standard",
  "orderItem": [
    {
      "id": "1",
      "action": "add",
      "@type": "standard",
      "service": {
        "serviceState": "Active",
        "serviceCharacteristic": [
          {
            "name": "vCPE_IP",
            "valueType": "String",
            "value": {
              "@type": "IPAddress",
              "@schemaLocation": "http....",
                                        "vCPE_IP": "193.218.236.21"
            }
          }
        ],
        "serviceSpecification": {
          "id": "12",
          "href": "http://...:serviceSpecification/12",
          "name": "vCPE",
          "version": "1",
          "@type": "vCPE",
          "@schemaLocation": "http..."
        }
      }
    }
  ]
}
```

Response

```
201
Content-Type: application/json

{
  "id": "11",
  "href": https://host:port/serviceOrder/11
}
```

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
description	
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 serviceOrderItem.service.relatedParty	Only when order is in “Acknowledged” state – delivery process not started
serviceOrderItem.service	Only when order is in “Acknowledged” state – delivery process not started
serviceOrderItem.appointment	Only when order is in “Acknowledged” state – delivery process not started or suspended
serviceOrderItem.orderItemRelationship	Only when order is in “Acknowledged” state – delivery process not started or suspended
serviceOrderItem.service.serviceSpecification	Only when order is in “Acknowledged” state – delivery process not started or suspended
serviceOrderItem.service.place	Only when order is in “Acknowledged” state – delivery process not started or suspended
serviceOrderItem.service.serviceRelationship	Only when order is in “Acknowledged” state – delivery process not started or suspended
serviceOrderItem.service.serviceCharacteristic	Only when order is in “Acknowledged” state – delivery process not started or suspended

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

Usage Samples

Here's an example of a request for patching a ServiceOrder resource with change on following attributes:

- description
- requestedStartDate
- requestedCompletionDate
- vCPE IP (characteristic) value

Request

```
PATCH /serviceOrderingManagement/v1/serviceOrder/42
Content-Type: application/merge-patch+json
{
  "description": "Service order new description",
  "requestedStartDate": "2018-01-16T09:37:40.508Z",
  "requestedCompletionDate": "2018-01-17T09:37:40.508Z",
  "orderItem": [
  {
   "id": "1",
   "service": {
    "serviceCharacteristic": [
     {
      "name": "vCPE_IP",
      "valueType": "String",
      "value": {
       "@type": "IPAddress",
       "@schemaLocation": "http....",
                         "vCPE_IP": "193.218.236.26"
      }
     }
    ]
   }
  }
 ]
}
```

Response

```
200

{
  "id": "42",
  "href": https://host:port/serviceOrder/42
}
```

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/v1/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.

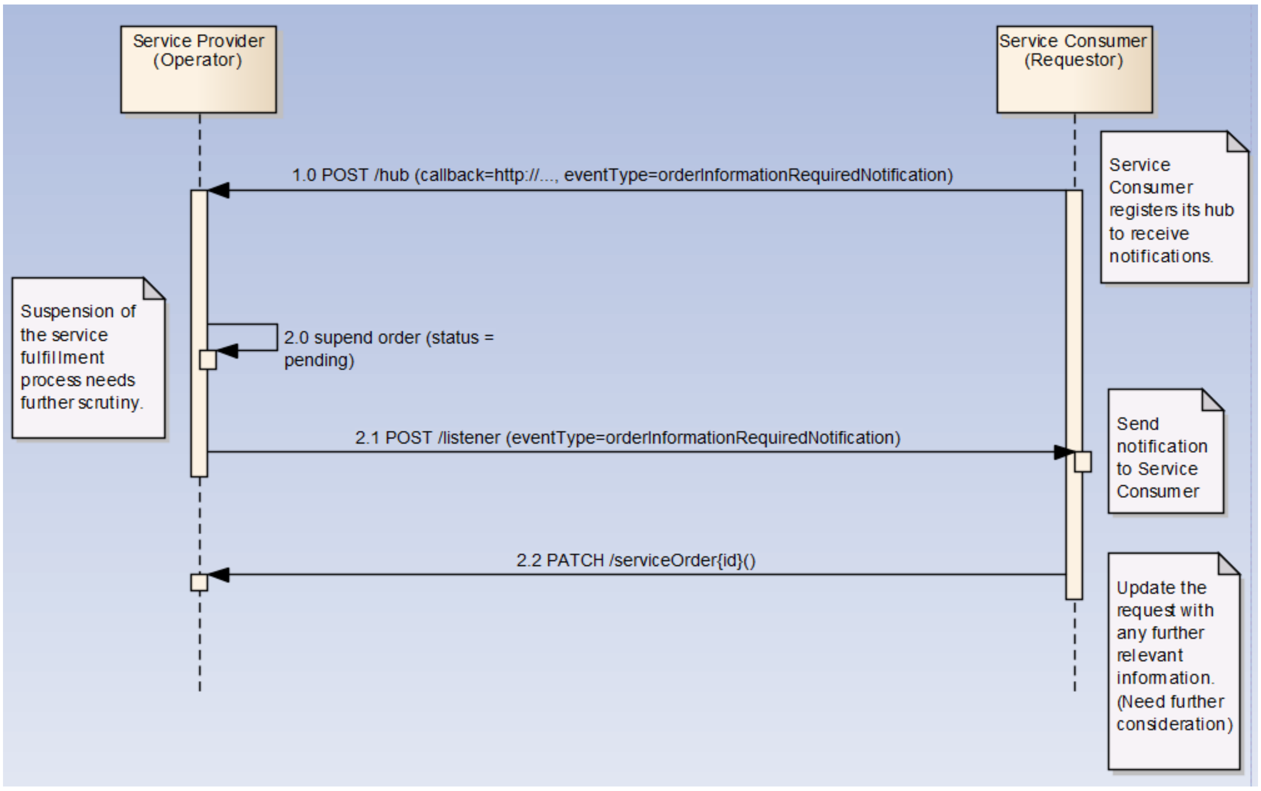

Service Provider (Operator)
Service Consumer (Requestor)
1.0 POST /hub (callback=http://..., eventType=orderInformationRequiredNotification)
Service Consumer registers its hub to receive notifications.
Suspension of the service fulfillment process needs further scrutiny.
2.0 supend order (status = pending)
2.1 POST /listener (eventType=orderInformationRequiredNotification)
Send notification to Service Consumer
2.2 PATCH /serviceOrder{id}()
Update the request with any further relevant information. (Need further consideration)

Figure 4 - Notification flow

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

Delete 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 /event

Description

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 /event
Accept: application/json

{
  "eventId": "12365",
  "eventTime": "2018-01-12T13:23:43.704Z",
  "eventType": " ServiceOrderStateChangeNotification ",
  "event": {
    "resource": "48"
  }
}
```

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		Pierre Gauthier TM Forum pgauthier@tmforum.org Andrew Forth aforth@amdocs.com August-Wilhelm Jagau August-wilhelm.jagau@ericsson.com	Final release of specification with additional comments on Resource usage
Release 2.0	24-Oct-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	
Release 18.0.0 Version 3.0.4	12-Jan-2018	Pierre Gauthier TM Forum pgauthier@tmforum.org Ludovic Robert Orange ludovic.robert@orange.com	Alignment with TMF open API guidelines 3.0