**TM Forum Specification**

**Service Ordering Management API REST Specification**

**TMF641**

**Release 18.5.0**

**January 2019**

|  |  |
| --- | --- |
| **Latest Update: TM Forum Release 18.5.0** | **Member Evaluation** |
| **Version 4.0.0** | **IPR Mode: RAND** |

# NOTICE

Copyright © TM Forum 2018. 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](http://www.tmforum.org/IPRPolicy/11525/home.html), 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. +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org/)

# Table of Contents

[NOTICE 2](#_Toc2197152)

[Table of Contents 3](#_Toc2197153)

[List of Tables 4](#_Toc2197154)

[Introduction 5](#_Toc2197155)

[SAMPLE USE CASES 6](#_Toc2197156)

[Support of polymorphism and extension 7](#_Toc2197157)

[RESOURCE MODEL 8](#_Toc2197158)

[Managed Entity and Task Resource Models 8](#_Toc2197159)

[Service Order resource 8](#_Toc2197160)

[Notification Resource Models 19](#_Toc2197161)

[Service Order Create Notification 20](#_Toc2197162)

[Service Order Attribute Value Change Notification 21](#_Toc2197163)

[Service Order State Change Notification 21](#_Toc2197164)

[Service Order Delete Notification 21](#_Toc2197165)

[API OPERATIONS 23](#_Toc2197166)

[Operations on Service Order 24](#_Toc2197167)

[List service orders 24](#_Toc2197168)

[Retrieve service order 27](#_Toc2197169)

[Create service order 30](#_Toc2197170)

[Patch service order 33](#_Toc2197171)

[Delete service order 35](#_Toc2197172)

[API NOTIFICATIONS 36](#_Toc2197173)

[Register listener 36](#_Toc2197174)

[Unregister listener 37](#_Toc2197175)

[Publish Event to listener 38](#_Toc2197176)

[Acknowledgements 39](#_Toc2197177)

[Document History 39](#_Toc2197178)

[Version History 39](#_Toc2197179)

[Release History 39](#_Toc2197180)

[Contributors to Document 39](#_Toc2197181)

# 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

Reader will find example of use cases using Usage API in “Open Digital Business Scenarios and Use Cases” document.

# Support of polymorphism and extension

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 TypeAServiceOrder or TypeBServiceOrder inheriting properties from the base ServiceOrder 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 ServiceOrder instances some may be instances of TypeAServiceOrder where other could be instances of TypeBServiceOrder . 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

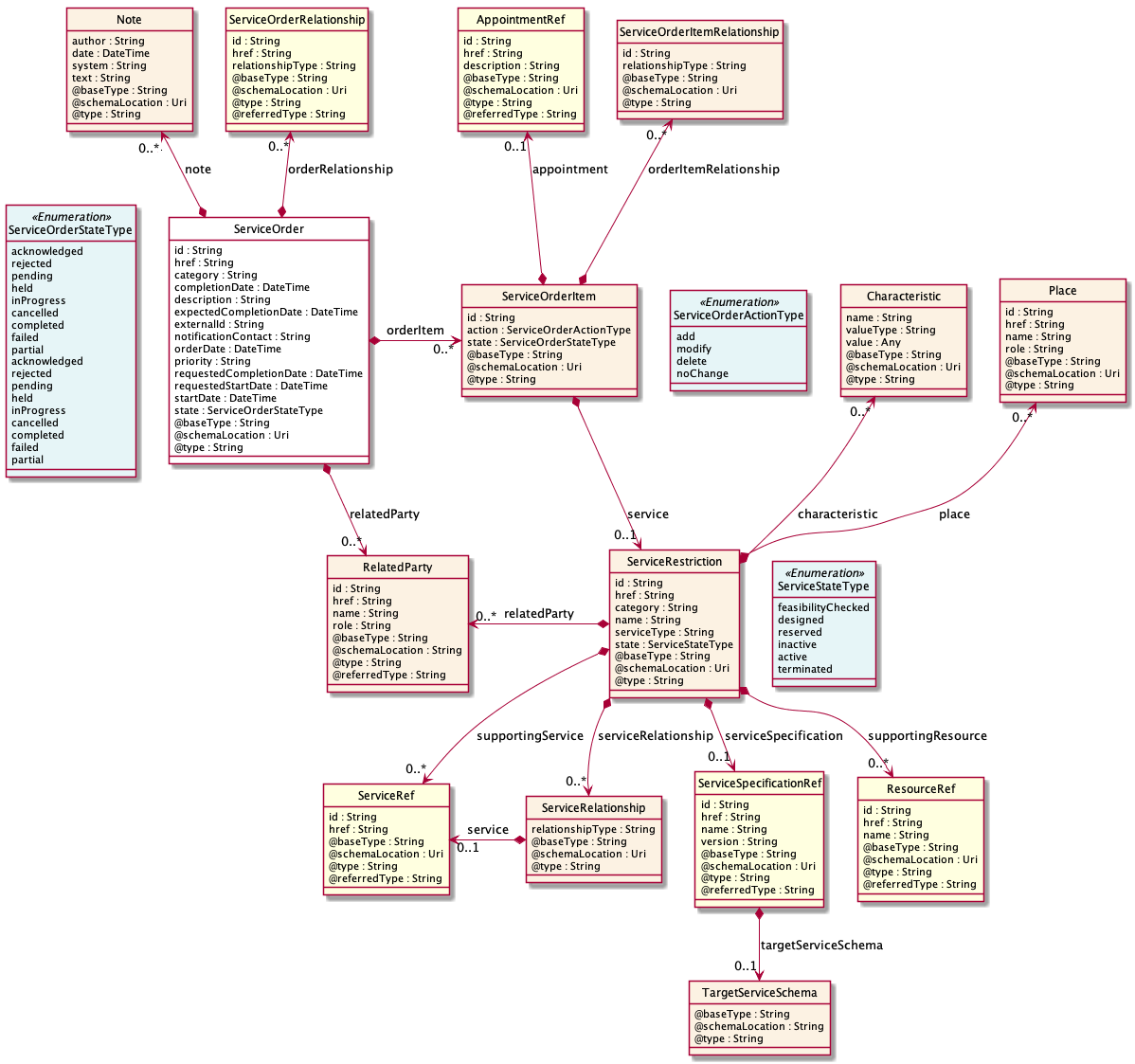
# RESOURCE MODEL

## Managed Entity and Task Resource Models

### Service 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**



**Field descriptions**

*ServiceOrder* fields

|  |  |
| --- | --- |
| category | A string. Used to categorize the order, useful for the OM system, such as: Broadband, TVOption. |
| completionDate | A date time (DateTime). Effective delivery date amended by the provider. |
| description | A string. A free-text description of the service order. |
| expectedCompletionDate | A date time (DateTime). Expected delivery date amended by the provider. |
| externalId | A string. ID given by the consumer to facilitate searches. |
| href | A string. Hyperlink to access the order. |
| id | A string. ID created on repository side. |
| 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. |
| notificationContact | A string. Contact attached to the order to send back information regarding this order. |
| orderDate | A date time (DateTime). |
| orderItem | A list of service order items (ServiceOrderItem [1..\*]). A list of service order items to be processed by this order. |
| orderRelationship | A list of service order relationships (ServiceOrderRelationship [\*]). A list of service orders related to this order (e.g. prerequisite, dependent on). |
| priority | A string. Can be used by consumers to prioritize orders in a Service Order Management system. |
| relatedParty | A list of related parties (RelatedParty [\*]). A list of parties which are involved in this order and the role they are playing. |
| requestedCompletionDate | A date time (DateTime). Requested delivery date from the requestors perspective. |
| requestedStartDate | A date time (DateTime). Order start date wished by the requestor. |
| startDate | A date time (DateTime). Date when the order was started for processing. |
| state | A service order state type (ServiceOrderStateType). State of the order: described in the state-machine diagram. |

*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. |

*ServiceOrderItem* sub-resource

|  |  |
| --- | --- |
| action | A service order action type (ServiceOrderActionType). The action to be carried out on the Service. Can be: add, modify, delete, noChange. |
| appointment | An appointment reference (AppointmentRef). An appointment that was set up with a related party for this order item. |
| id | A string. Identifier of the individual line item. |
| orderItemRelationship | A list of service order item relationships (ServiceOrderItemRelationship [\*]). A list of order items related to this order item. |
| service | A service restriction (ServiceRestriction). The Service to be acted on by the order item. |
| state | A service order state type (ServiceOrderStateType). State of the order item: described in the state machine diagram. This is the requested state. |

*ServiceOrderItemRelationship* sub-resource

Linked service order item to the one containing this attribute.

|  |  |
| --- | --- |
| id | A string. Unique identifier of a service order item. |
| relationshipType | A string. The type of related order item, can be: dependency if the order item needs to be not started until another order item is complete. |

*ServiceOrderRelationship* sub-resource

Linked service order to the one containing this attribute.

|  |  |
| --- | --- |
| @referredType | A string. The entity type of the related order. |
| href | A string. A hyperlink to the related order. |
| id | A string. The id of the related order. |
| relationshipType | A string. The type of related order, such as: [dependency] if the order needs to be [not started] until another order item is complete (a service order in this case) or [cross-ref] to keep track of the source order (a productOrder). |

*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. |

*ServiceRestriction* sub-resource

In the context of a service order, and depending of the action requested (add/modify/delete/noChange) this data structure captures the configuration to apply to an existing subscribed service or to a new one.

|  |  |
| --- | --- |
| category | A string. Is it a customer facing or resource facing service. |
| href | A string. Reference of the service. |
| id | A string. Unique identifier of the service. |
| name | A string. Name of the service. |
| place | A list of places (Place [\*]). A list of places (Place [\*]). Used to define a place useful for the service (for example a delivery geographical place). |
| relatedParty | A list of related parties (RelatedParty [\*]). A list of related party references (RelatedParty [\*]). 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 [\*]). |
| 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. |
| state | A service state type (ServiceStateType). The life cycle state of the service, such as: [feasibilityChecked], [designed]. |
| 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). |

*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. |

*AppointmentRef* relationship

Refers an appointment, such as a Customer presentation or internal meeting or site visit.

|  |  |
| --- | --- |
| @referredType | A string. The actual type of the target instance when needed for disambiguation. |
| description | A string. An explanatory text regarding the appointment made with a party. |
| href | A string. The reference of the appointment. |
| id | A string. The identifier of the referred appointment. |

*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. |

*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.

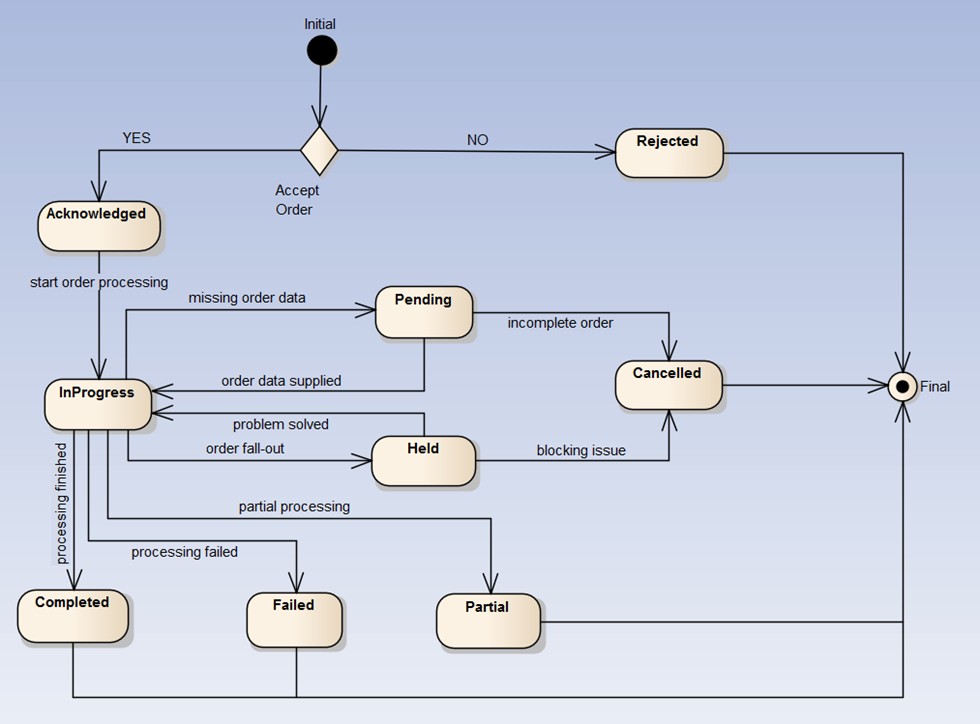
|  |  |
| --- | --- |
| @referredType | A string. The actual type of the target instance when needed for disambiguation. |
| href | A string. Reference of the serviceSpecification. |
| id | A string. Unique identifier of the service specification. |
| name | A string. Name of the requiredServiceSpecification. |
| 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. |

**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):

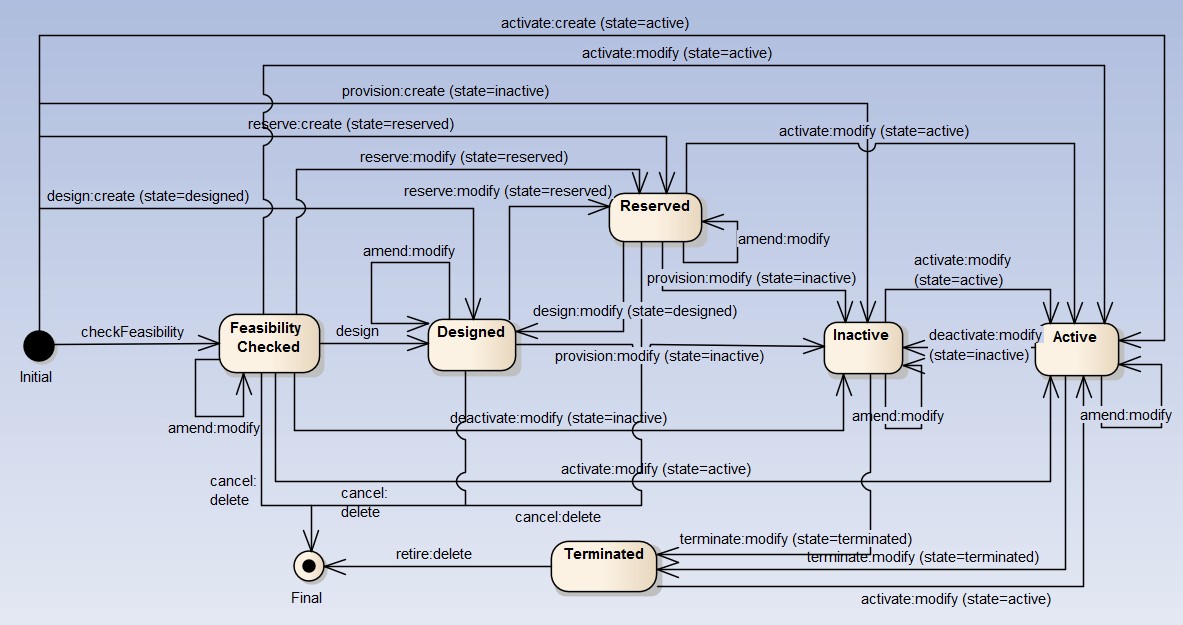


The following table provides service order state and service order item state description:

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: |

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.

|  |  |  |
| --- | --- | --- |
| **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.

**Json representation sample**

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

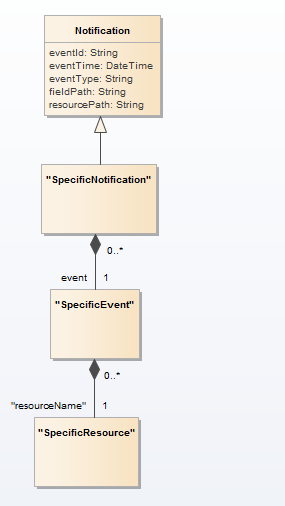
|  |
| --- |
| {  "id": "42",   "href": "http://serverlocation:port/serviceOrderingManagement/v4/serviceOrder/42",   "externalId": "OrangeBSS747",   "priority": "1",   "description": "Service order description",   "category": "TMF resource illustration",   "state": "acknowledged",   "orderDate": "2018-01-12T09:37:40.508Z",   "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": "ServiceOrder",   "note": [{  "date": "2018-01-15T09:37:40.508Z",   "author": "Jean Pontus",   "text": "Some text"  }],   "relatedParty": [  {  "id": "456",   "href": "http://serverlocation:port/partyManagement/v4/party/456",   "role": "requester",   "name": "Jean Pontus",   "@referredType": "Individual"  }  ],   "orderItem": [  {  "id": "1",   "action": "add",   "state": "acknowledged",   "service": {  "@type": "vCPE",   "@schemaLocation": "http//my.schemas/vCpe.schema.json",   "state": "active",   "serviceType": "CFS",   "serviceCharacteristic": [  {  "name": "vCPE\_IP",   "valueType": "object",   "value": {  "@type": "IPAddress",   "@schemaLocation": "http://my.schemas/IPAddress.schema.json",   "vCPE\_IP": "193.218.236.21"  }  }  ],   "serviceSpecification": {  "@type": "ONAPServiceSpec",   "@schemaLocation": "http//my.schemas/OnapServiceSpec.schema.json",   "id": "12",   "href": "http://.../serviceSpecification/12",   "name": "vCPE",   "version": "1",   "invariantUUID": "456-852-357",   "toscaModelURL": "http//...",   "targetServiceSchema": {  "@type": "vCPE",   "@schemaLocation": "http//my.schemas/vCpe.schema.json"  }  }  }  },   {  "id": "2",   "action": "modify",   "state": "acknowledged",   "service": {  "id": "456",   "href": "http://serverlocation:port/serviceInventoryManagement/v4/service/456",   "state": "active",   "serviceType": "CFS",   "serviceCharacteristic": [  {  "name": "Characteristic1",   "value": "newValue"  }  ],   "supportingResource": [  {  "id": "3456\_DFG5-H690",   "href": "http://...",   "@referredType": "CloudResource"  }  ]  }  },   {  "id": "3",   "action": "add",   "state": "acknowledged",   "service": {  "state": "active",   "serviceType": "CFS",   "serviceRelationship": [  {  "relationshipType": "reliesOn",   "service": {  "href": "https://.../serviceManagement/v4/service/45",   "id": "45"  }  }  ],   "serviceSpecification": {  "id": "48",   "href": "http//..../48",   "name": "genericService48",   "version": "2"  }  }  },   {  "id": "4",   "action": "modify",   "state": "acknowledged",   "service": {  "id": "12",   "href": "http://serverlocation:port/serviceInventoryManagement/v4/service/12",   "state": "inactive"  }  }  ] } |

## Notification Resource Models

4 notifications are defined for this API

Notifications related to ServiceOrder:  
 - ServiceOrderCreateNotification  
 - ServiceOrderAttributeValueChangeNotification  
 - ServiceOrderStateChangeNotification  
 - ServiceOrderDeleteNotification

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 Create Notification

Notification ServiceOrderCreateNotification case for resource ServiceOrder

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceOrderCreateNotification",  "event": {  "serviceOrder" :   {-- SEE ServiceOrder RESOURCE SAMPLE --}  } } |

### Service Order Attribute Value Change Notification

Notification ServiceOrderAttributeValueChangeNotification case for resource ServiceOrder

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceOrderAttributeValueChangeNotification",  "event": {  "serviceOrder" :   {-- SEE ServiceOrder RESOURCE SAMPLE --}  } } |

### Service Order State Change Notification

Notification ServiceOrderStateChangeNotification case for resource ServiceOrder

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceOrderStateChangeNotification",  "event": {  "serviceOrder" :   {-- SEE ServiceOrder RESOURCE SAMPLE --}  } } |

### Service Order Delete Notification

Notification ServiceOrderDeleteNotification case for resource ServiceOrder

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ServiceOrderDeleteNotification",  "event": {  "serviceOrder" :   {-- SEE ServiceOrder RESOURCE SAMPLE --}  } } |

# API OPERATIONS

Remember the following Uniform Contract:

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

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

Notifications are also described in a subsequent section.

## Operations on Service 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** |
| GET {apiRoot}/serviceOrder?state=completed&category= TMFresourceillustration &completionDate.gt=2018-01-14T00:00:00.000Z Accept: application/json |
| **Response** |
| 200  [  {  "id": "42",   "href": "http://serverlocation:port/serviceOrderingManagement/v4/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": "ServiceOrder",   "note": [{  "date": "2018-01-15T09:37:40.508Z",   "author": "Jean Pontus",   "text": "Some text"  }],   "relatedParty": [  {  "id": "456",   "href": "http://serverlocation:port/partyManagement/v4/party/456",   "role": "requester",   "name": "Jean Pontus",   "@referredType": "Individual"  }  ],   "orderItem": [  {  "id": "1",   "action": "add",   "state": "acknowledged",   "service": {  "@type": "vCPE",   "@schemaLocation": "http//my.schemas/vCpe.schema.json",   "state": "active",   "serviceType": "CFS",   "serviceCharacteristic": [  {  "name": "vCPE\_IP",   "valueType": "object",   "value": {  "@type": "IPAddress",   "@schemaLocation": "http://my.schemas/IPAddress.schema.json",   "vCPE\_IP": "193.218.236.21"  }  }  ],   "serviceSpecification": {  "@type": "ONAPServiceSpec",   "@schemaLocation": "http//my.schemas/OnapServiceSpec.schema.json",   "id": "12",   "href": "http://.../serviceSpecification/12",   "name": "vCPE",   "version": "1",   "invariantUUID": "456-852-357",   "toscaModelURL": "http//...",   "targetServiceSchema": {  "@type": "vCPE",   "@schemaLocation": "http//my.schemas/vCpe.schema.json"  }  }  }  },   {  "id": "2",   "action": "modify",   "state": "acknowledged",   "service": {  "id": "456",   "href": "http://serverlocation:port/serviceInventoryManagement/v4/service/456",   "state": "active",   "serviceType": "CFS",   "serviceCharacteristic": [  {  "name": "Characteristic1",   "value": "newValue"  }  ],   "supportingResource": [  {  "id": "3456\_DFG5-H690",   "href": "http://...",   "@referredType": "CloudResource"  }  ]  }  },   {  "id": "3",   "action": "add",   "state": "acknowledged",   "service": {  "state": "active",   "serviceType": "CFS",   "serviceRelationship": [  {  "relationshipType": "reliesOn",   "service": {  "href": "https://.../serviceManagement/v4/service/45",   "id": "45"  }  }  ],   "serviceSpecification": {  "id": "48",   "href": "http//..../48",   "name": "genericService48",   "version": "2"  }  }  },   {  "id": "4",   "action": "modify",   "state": "acknowledged",   "service": {  "id": "12",   "href": "http://serverlocation:port/serviceInventoryManagement/v4/service/12",   "state": "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** |
| GET {apiRoot}/serviceOrder/43 Accept: application/json |
| **Response** |
| 200  {  "id": "42",   "href": "http://serverlocation:port/serviceOrderingManagement/v4/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": "ServiceOrder",   "note": [{  "date": "2018-01-15T09:37:40.508Z",   "author": "Jean Pontus",   "text": "Some text"  }],   "relatedParty": [  {  "id": "456",   "href": "http://serverlocation:port/partyManagement/v4/party/456",   "role": "requester",   "name": "Jean Pontus",   "@referredType": "Individual"  }  ],   "orderItem": [  {  "id": "1",   "action": "add",   "state": "acknowledged",   "service": {  "@type": "vCPE",   "@schemaLocation": "http//my.schemas/vCpe.schema.json",   "state": "active",   "serviceType": "CFS",   "serviceCharacteristic": [  {  "name": "vCPE\_IP",   "valueType": "object",   "value": {  "@type": "IPAddress",   "@schemaLocation": "http://my.schemas/IPAddress.schema.json",   "vCPE\_IP": "193.218.236.21"  }  }  ],   "serviceSpecification": {  "@type": "ONAPServiceSpec",   "@schemaLocation": "http//my.schemas/OnapServiceSpec.schema.json",   "id": "12",   "href": "http://.../serviceSpecification/12",   "name": "vCPE",   "version": "1",   "invariantUUID": "456-852-357",   "toscaModelURL": "http//...",   "targetServiceSchema": {  "@type": "vCPE",   "@schemaLocation": "http//my.schemas/vCpe.schema.json"  }  }  }  },   {  "id": "2",   "action": "modify",   "state": "acknowledged",   "service": {  "id": "456",   "href": "http://serverlocation:port/serviceInventoryManagement/v4/service/456",   "state": "active",   "serviceType": "CFS",   "serviceCharacteristic": [  {  "name": "Characteristic1",   "value": "newValue"  }  ],   "supportingResource": [  {  "id": "3456\_DFG5-H690",   "href": "http://...",   "@referredType": "CloudResource"  }  ]  }  },   {  "id": "3",   "action": "add",   "state": "acknowledged",   "service": {  "state": "active",   "serviceType": "CFS",   "serviceRelationship": [  {  "relationshipType": "reliesOn",   "service": {  "href": "https://.../serviceManagement/v4/service/45",   "id": "45"  }  }  ],   "serviceSpecification": {  "id": "48",   "href": "http//..../48",   "name": "genericService48",   "version": "2"  }  }  },   {  "id": "4",   "action": "modify",   "state": "acknowledged",   "service": {  "id": "12",   "href": "http://serverlocation:port/serviceInventoryManagement/v4/service/12",   "state": "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. Notice that it is up to an implementer to add additional mandatory attributes.

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

|  |  |
| --- | --- |
| Mandatory Attributes | Rule |
| orderItem |  |

|  |  |  |
| --- | --- | --- |
| Non Mandatory Attributes | Default Value | Rule |
| category |  |  |
| completionDate |  |  |
| description |  |  |
| expectedCompletionDate |  |  |
| externalId |  |  |
| note |  |  |
| notificationContact |  |  |
| orderDate |  |  |
| orderRelationship |  |  |
| priority | 4 (lowest) |  |
| relatedParty |  |  |
| requestedCompletionDate |  |  |
| requestedStartDate |  |  |
| startDate |  |  |
| state |  |  |

**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 |
| orderItem | id, action, service |
| relatedParty | name, role |

The following table indicates non-mandatory sub-attributes when creating a service order.

|  |  |  |
| --- | --- | --- |
| Non Mandatory Sub Attributes | Default Value | Rule |
| orderItem.ServiceSpecification |  | The serviceSpecification may not be useful when doing a “modify” or “delete” on an owned Service |
| orderItem.place |  |  |
| orderItem.appointment |  |  |
| orderItem.orderItemRelationship |  |  |
| orderRelationship |  |  |
| ServiceOrderItem.Service.serviceState | active |  |

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’ |

The following pre-conditions apply for this operation.

|  |
| --- |
| Pre-conditions |
| When creating a service order (post) an order item should not have the state field valorized |

**Usage Samples**

Here's a sample of a request for creating a ServiceOrder resource. We use the ‘fields’ parameter to restrict the numbers of attributes returned in the response.

|  |
| --- |
| **Request** |
| POST {apiRoot}/serviceOrder?fields=id,href 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": "ServiceOrder",   "orderItem": [  {  "id": "1",   "action": "add",   "@type": "ServiceOrder",   "service": {  "serviceState": "active",   "type": "CFS",   "serviceCharacteristic": [  {  "name": "vCPE\_IP",   "valueType": "object",   "value": {  "@type": "IPAddress",   "@schemaLocation": "http://my.schemas/IPAddress.schema.json",   "vCPE\_IP": "193.218.236.26"  }  }  ],   "serviceSpecification": {  "id": "12",   "href": "http://...:serviceSpecification/12",   "name": "vCPE",   "version": "1",   "@type": "vCPE",   "@schemaLocation": "http..."  }  }  }  ] } |
| **Response** |
| 201  {  "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 |
| category |  |
| description |  |
| expectedCompletionDate |  |
| note |  |
| notificationContact |  |
| orderItem |  |
| orderRelationship |  |
| startDate |  |

|  |  |
| --- | --- |
| Conditional Patchable Attributes | Rule |
| relatedParty | Only when order is in “Acknowledged” state  – delivery process not started |
| requestedCompletionDate | Only when order is in “Acknowledged” state  – delivery process not started |
| requestedStartDate | 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 |
| serviceOrderItem.relatedParty | Only when order is in “Acknowledged” state –  delivery process not started |

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

**Additional Rules**

The following pre-conditions apply for this operation.

|  |
| --- |
| Pre-conditions |
| When creating a service order (post) an order item should not have the state field valorized |

**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 {apiRoot}/serviceOrder/42 Content-Type: application/json-patch+json  ServiceOrder\_partialupdate\_1\_request.sample.json |
| **Response** |
| 200  {  "id": "42",   "href": "https://host:port/serviceOrder/42" } |

### Delete service order

**DELETE /serviceOrder/{id}**

**Description**

This operation deletes a service order entity.

**Usage Samples**

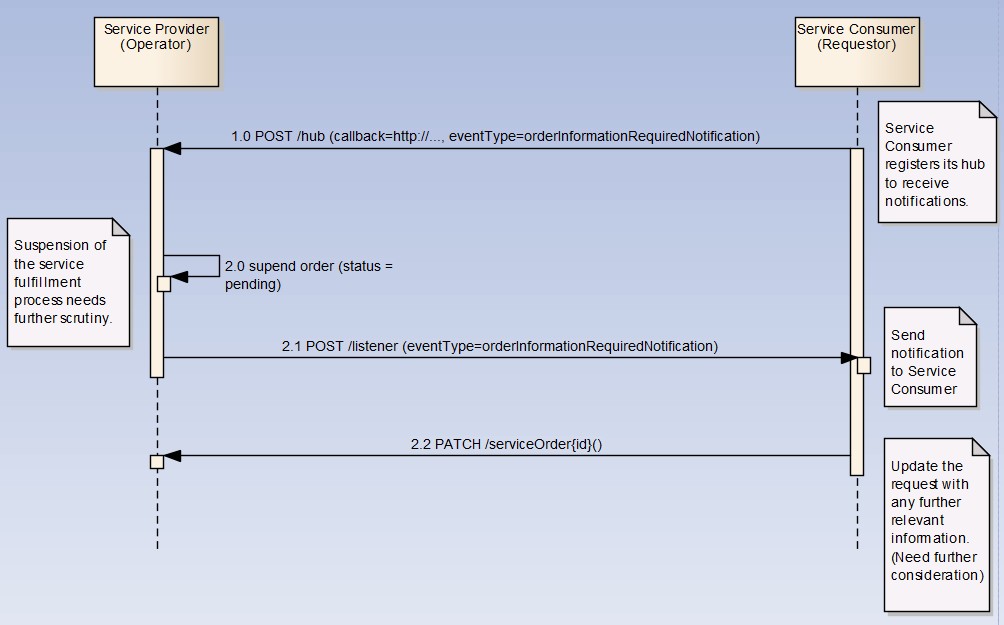
Here's an example of a request for deleting a ServiceOrder resource.

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

## Unregister listener

**DELETE /hub/{id}**

**Description**

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

**Behavior**

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

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

**Usage Samples**

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

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

## Publish Event to listener

**POST /client/listener**

**Description**

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

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

**Behavior**

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

**Usage Samples**

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

|  |
| --- |
| **Request** |
| POST /client/listener  Accept: application/json  {  "event": {  EVENT BODY  },  "eventType": "EVENT\_TYPE"  } |
| **Response** |
| 201 |

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

# Acknowledgements

## Document History

### Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Release led by:** | **Description** |
| Release 1.0 |  | Pierre Gauthier, TM Forum  [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Andrew Forth  [aforth@amdocs.com](file:///C:\Users\olri6401\AppData\Local\Temp\ludovic.robert@orange.com)  August-Wilhelm Jagau  August-wilhelm.jagau@ericsson.com | First Release of the Document. |
| Release 2.0 | 24 Oct 2016 | Pierre Gauthier, TM Forum [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Ludovic Robert Orange [ludovic.robert@orange.com](file:///C:\Users\olri6401\AppData\Local\Temp\ludovic.robert@orange.com)  Nicoleta Stoica  Vofadone [nicoleta.stoica@vodafone.com](mailto:nicoleta.stoica@vodafone.com)  Jean-Luc Tymen Orange [jeanluc.tymen@orange.com](mailto:jeanluc.tymen@orange.com) | Alignment with Guidelines 3.0 |
| Release 4.0 | 16 Jan 2019 | Pierre Gauthier, TM Forum  [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Ludovic Robert Orange [ludovic.robert@orange.com](file:///C:\Users\olri6401\AppData\Local\Temp\ludovic.robert@orange.com)  Mariano Belaunde, Orange  [mariano.belaunde@orange.com](mailto:mariano.belaunde@orange.com) |  |

### Release History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release Number** | **Date** | **Release led by:** | **Description** |
| 18.5.0 | 16 Jan 2019 | Pierre Gauthier TM Forum [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Ludovic Robert Orange [ludovic.robert@orange.com](file:///C:\Users\olri6401\AppData\Local\Temp\ludovic.robert@orange.com)  Mariano Belaunde  Orange  [mariano.belaunde@orange.com](mailto:mariano.belaunde@orange.com) |  |

## Contributors to Document

|  |  |
| --- | --- |
|  |  |