



## *Frameworkx Specification*

# Resource Ordering API REST Specification

**TMFXXX**  
**Release 16.5**  
**November 2016**

<b>Latest Update: Frameworkx Release 16.0</b>	<b>Member Evaluation</b>
<b>Version 1.0.0</b>	<b>IPR Mode: RAND</b>

## 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 – 10<sup>th</sup> Floor,  
Morristown, NJ 07960 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
Table of Contents .....	3
List of Tables .....	5
Introduction .....	6
SAMPLE USE CASES .....	8
RESOURCE MODEL .....	9
Managed Entity and Task Resource Models .....	9
Resource Order resource .....	9
Notification Resource Models .....	16
Resource Order Creation Notification .....	17
Resource Order Attribute Value Change Notification .....	18
Resource Order State Change Notification .....	18
Resource Order Remove Notification .....	18
API OPERATIONS .....	20
Operations on Resource Order .....	21
List resource orders .....	21
Retrieve resource order .....	21
Create resource order .....	24
Patch resource order .....	28
Delete resource order .....	30
API NOTIFICATIONS .....	31
Register listener .....	31
Unregister listener .....	32
Publish Event to listener .....	32
Acknowledgements .....	34

Release History ..... 34

## LIST OF TABLES

N/A

## INTRODUCTION

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

A Resource Order API provides a standard mechanism for placing a Resource Order with all necessary order parameters. A Resource Order is created based on a resource candidate that is defined in the resource catalog. The Resource candidate is an entity that makes a Resource Specification available to a resource catalog.

Resources are physical or non-physical components (or some combination of these) within an enterprise's infrastructure or inventory. They are typically consumed or used by Services (for example a physical port assigned to a service) or contribute to the realization of a Product (for example, a SIM card). They can be drawn from the Application, Computing and Network domains, and include, for example, Network Elements, software, IT systems, content and information, and technology components.

A Resource Specification is an abstract base class for representing a generic means for implementing a particular type of Resource. In essence, a Resource Specification defines the common attributes and relationships of a set of related Resources, while Resource defines a specific instance that is based on a particular Resource Specification.

Resource Order API manages resource order resource:

- A Resource Order is a type of order which can be used to place an order between a service provider and a partner and vice versa
- Main order items attributes are the ordered resource candidates and resource characteristics with the related action to be performed (e.g: add or delete resources), state.

Resource Order API performs the following operations on the Resource Order:

- Retrieval of a Resource Order or a collection of Resource Orders depending on filter criteria
- Partial update of a ResourceOrder
- Creation of a Resource order
- Deletion of a resource order (for admin purposes)
- Notifications of events on resource order
  - o Order creation
  - o Order removal
  - o Order state change
  - o Order value change
  - o Order information required

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

- The Order Management system has access to the catalog system



## SAMPLE USE CASES

Reader will find examples of use cases in “Open Digital Business Scenarios and Use Cases” document.



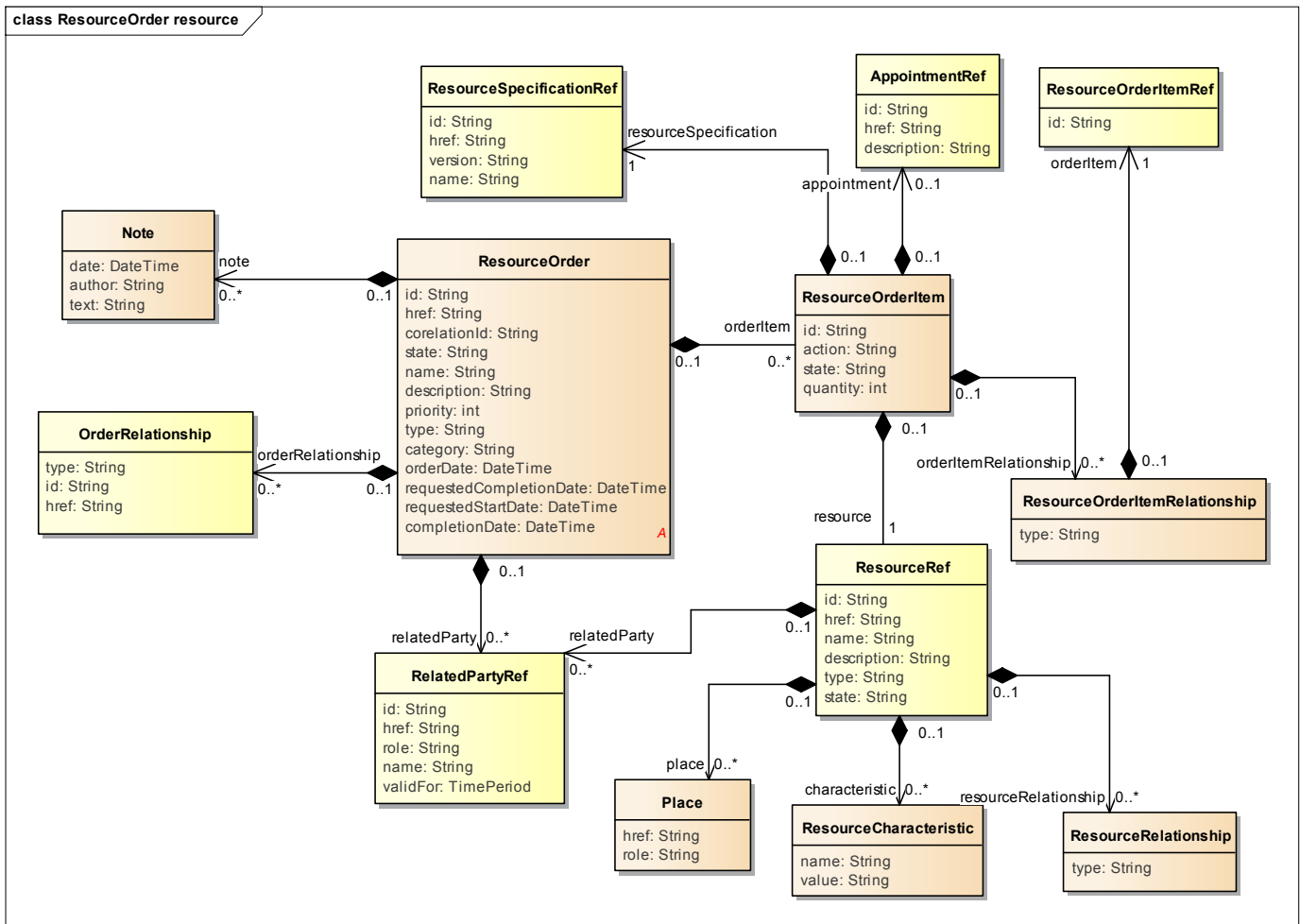
## RESOURCE MODEL

### Managed Entity and Task Resource Models

### RESOURCE ORDER RESOURCE

A Resource Order is a request to provision a set of Resources (logical and physical) triggered by the request to provision a Service through a Service Order.

#### Resource model



#### Lifecycle

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, for example :

Order, state : InProgress

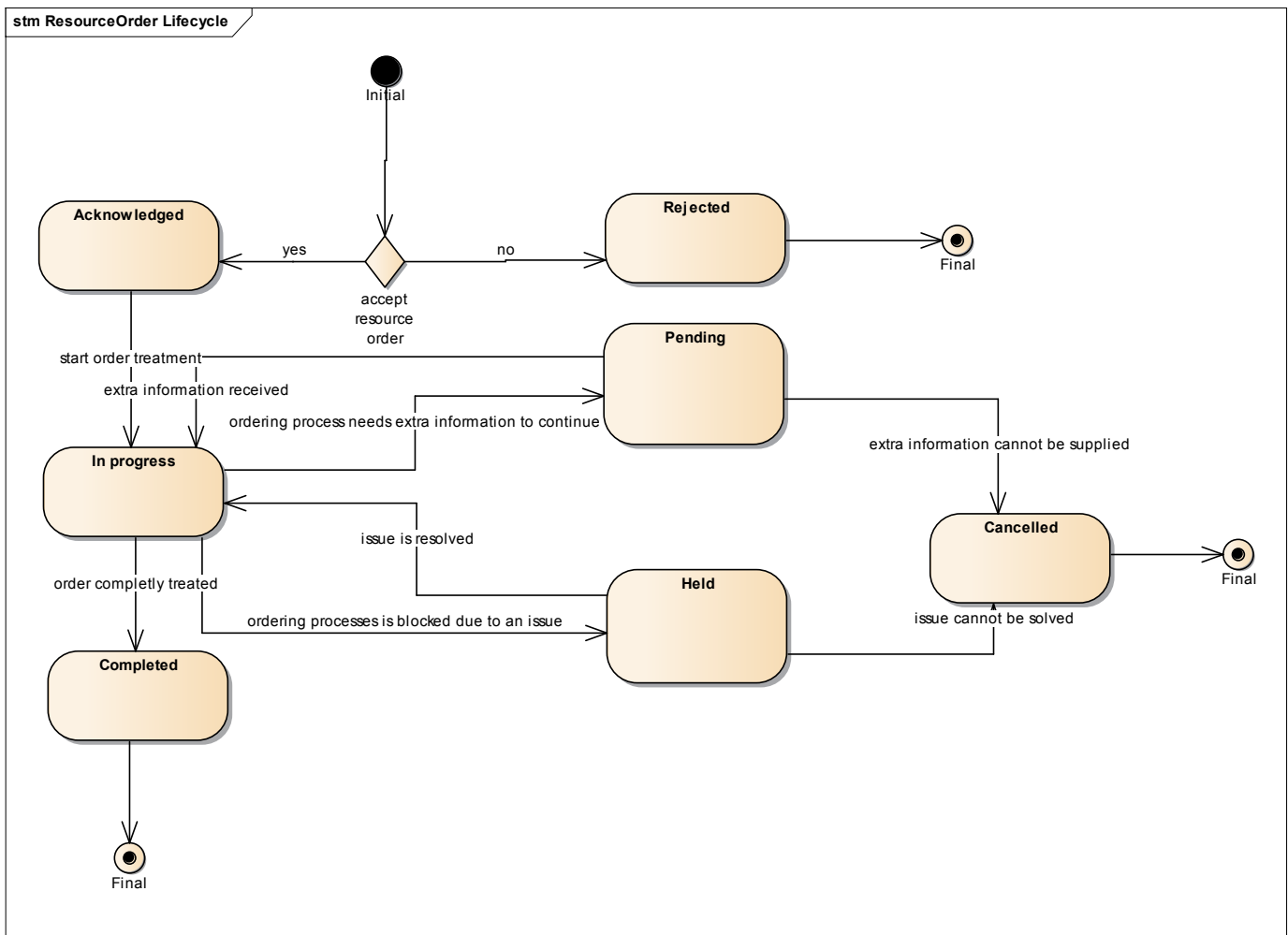
Order item 1, state : InProgress (Valid)

Order item 2, state : Pending (Valid)

Order item 3, state : Held (Valid)

Order item 4, state : Completed (Valid until there is another order item which is NOT completed)

Order item 5, state : Cancelled (Not valid : if an order item is cancelled, maybe the whole order should be cancelled too) The state machine specifying the typical state change transitions is provided below.



Acknowledged	The <i>Acknowledged</i> state is where an order has been received and has passed message and basic business validations.
In Progress	The <i>In Progress</i> state is where an order has passed the Order Feasibility check successfully and resource delivery has started.
Cancelled	The <i>Cancelled</i> state is where an In-Flight Order has been successfully cancelled.
Completed	The <i>Completed</i> state is where an order has complete provision and the resource is now active.
Rejected	The <i>Rejected</i> state is where: <ul style="list-style-type: none"> <li>- Invalid information is provided through the order request</li> <li>- The order request fails to meet business rules for ordering.</li> </ul>
Pending	The <i>Pending</i> 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.

## Field descriptions

### ResourceOrder fields

category	A string. Used to categorize the order from a business perspective that can be useful for the OM system.
completionDate	A date time (DateTime). Date when the order was completed.
corelationId	A string. ID given by the consumer and only understandable by him (to facilitate his searches afterwards).
description	A string. free-text description of the Resource Order.
href	A string. Hyperlink to access the order.
id	A string. Identifier of an instance of the Resource Order. Required to be unique within the resource type. Used in URIs as the identifier for specific instances of a type.
name	A string. A string used to give a name to the Resource Order.
orderDate	A date time (DateTime). Date when the order was created.
priority	An int (int). A way that can be used by consumers to prioritize orders in OM system (from 0 to 4 : 0 is the highest priority, and 4 the lowest).
requestedCompletionDate	A date time (DateTime). Requested delivery date from the requestor perspective.
requestedStartDate	A date time (DateTime). Order start date wished by the requestor.
state	A string. The life cycle state of the resource.
type	A string. Name of the Resource Order type.
orderRelationship	A list of resource order relationships (OrderRelationship [*]).
relatedParty	A list of related party references (RelatedPartyRef [*]). A related party defines party or party role linked to a specific entity.
note	A list of notes (Note [*]). Extra information about the ticket or a product order.

### Note sub-resource

Extra information about the resource order.

author	A string. Author of the note.
--------	-------------------------------

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

text A string. Text of the note.

#### OrderRelationship sub-resource

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 resource order in this case)
- “cross-ref” to keep track of the source order (a serviceOrder)

#### ResourceOrderItem

An identified part of the order. A resource 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 Resource. Can be:

- add
- modify
- delete
- noChange

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.

resourceSpecification A service specification (ResourceSpecificationRef). The resource specification (default values, etc. are fetched from the catalogue).

resourceOrderItemRelationship A list of order items relationships (ResourceOrderItemRelationship[\*]). Linked order items to the one containing this attribute.

resource A resource references (ResourceRef). The resource to be acted on by the order item.

### ResourceOrderItemRelationship

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"> <li>“dependency” if the order item needs to be “not started” until another order item is complete</li> </ul>
orderItem	A resource order item references (ResourceOrderItemRef). 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
description	An explanatory text regarding the appointment made with a party

### Place

Used to define a place useful for the resource (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).

### Resource

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

id	A string. Identifier of a resource instance. Required to be unique. Used in URIs as the identifier of the resource (for modify or delete use cases).
href	A string. Reference to the owned resource (useful for delete or modify command).
description	A string. A description of the resource (what it provides).
name	A string. Name of the resource.
state	A string. The lifecycle state of the resource.
place	A list of places (Place [*]). Used to defined places useful for the resource (for example a delivery geographical place).

resourceCharacteristic	A list of resource characteristics (ResourceCharacteristic[*]). 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.
resourceRelationship	A list or resource relationships (ResourceRelationship[*]). Linked Resource to the one instantiate, it can be : <ul style="list-style-type: none"> <li>• “reliesOn” if the resource needs another already owned resource to rely on (e.g. an option on an already owned mobile access Resource)</li> <li>• “targets” or “isTargeted” (depending on the way of expressing the link) for any other kind of links that may be useful</li> </ul>
relatedParty	A list of related party references (RelatedPartyRef[*]). Parties linked at the resource level (it may be a User for example).

#### RelatedPartyRef relationship

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

href	A string. Reference of the related party, could be a party reference or a party role reference.
id	A string. Unique identifier of a related party.
name	A string. Name of the related party.
role	A string. Role of the related party.
validFor	A time period. Validity period of the related party.

#### Json representation sample

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

```
{
  "id": "42",
  "href": "http://serverlocation:port/resourceOrderingManagement/resourceOrder/42",
  "corelationID": "MyResourceOrder_42",
  "priority": "1",
  "description": "A wonderful 42 order for brand new Resources",
  "category": "Uncategorized",
  "state": "InProgress",
  "orderDate": "2013-04-12T16:42:23-04:00",
  "requestedStartDate": "2013-04-12T16:42:23-04:00",
  "requestedCompletionDate": "2013-04-19T16:42:23-04:00",
  "completionDate": "2013-04-19T16:42:23-04:00",
  "orderRelationship": [{
```

```

        "type": "dependency",
        "href": "https://host:port/ resourceOrderingManagement /resourceOrder/3304",
        "id": "3304"
    }, {
        "type": "cross-ref",
        "href": "https://host:port/ serviceOrderingManagement /serviceOrder/3304",
        "id": "3304"
    }
],
"note": [{
    "text": "A free text detailing the note",
    "date": "2013-04-12T16:42:23-04:00",
    "author": "name"
}],
"relatedParty": [{
    "role": "owner",
    "id": "345221",
    "name": "John Doe"
}],
"resourceOrderItem": [{
    "id": "1",
    "action": "add",
    "state": "Acknowledged",
    "appointment": {
        "href": "http://serverlocation:port/appointment/100"
    },
    "resourceSpecification": {
        "id": "42",
        "href": "http://serverlocation: port/resourceCatalogManagement/resourceSpecification/42"
    },
    "resource": {
        "place": {
            "href": "http://map.google.com/.../1234112GDE",
            "role": "DeliveryPlace"
        },
        "resourceCharacteristic": [{
            "name": "Colour",
            "value": "White"
        }, {
            "name": "Memory",
            "value": "16"
        }
    ]
    }
}],
"resourceOrderItem": {
    "id": "2",
    "action": "modify",
    "state": "InProgress",
    "resourceSpecification": {

```

```

        "id": "43",
        "href": "http://serverlocation:port/resourceCatalogManagement/resourceSpecification/43"
    },
    "resource": {
        "id": "456",
        "href": "http://serverlocation:port/resourceInventoryManagement/logicalResource/456",
        "resourceCharacteristic": [{
            "name": "anotherCharacteristic",
            "value": "itsValue"
        }],
        "relatedParty": [{
            "role": "owner",
            "id": "5667443",
            "name": "Jimmy Doe"
        }]
    }
}, {
    "id": "3",
    "action": "add",
    "state": "InProgress",
    "resourceSpecification": {
        "id": "43",
        "href": "http://serverlocation:port/resourceCatalogManagement/resourceSpecification/51"
    },
    "resource": {
        "resourceRelationship": {
            "type": "reliesOn",
            "resource": {
                "href": "411",
                "id": "http://serverlocation:
port/resourceInventoryManagement/resource/411"
            }
        }
    }
}
}
}

```

## Notification Resource Models

5 notifications are defined for this API

Notifications related to ResourceOrder:

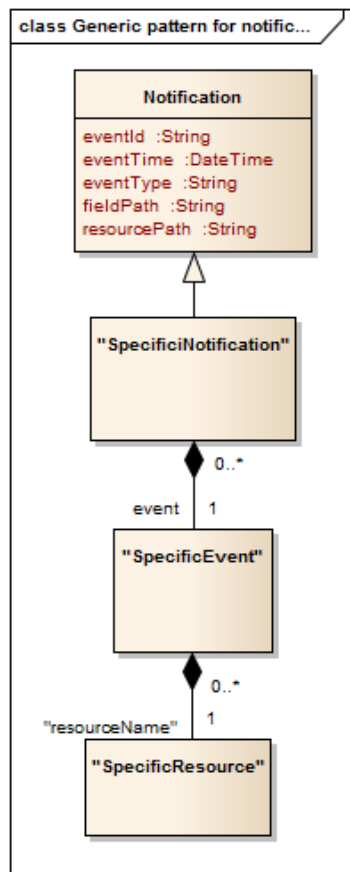
- ResourceOrderCreationNotification
- ResourceOrderAttributeValueChangeNotification
- ResourceOrderStateChangeNotification
- ResourceOrderRemoveNotification



### - ResourceInformationRequiredNotification

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).



## RESOURCE ORDER CREATION NOTIFICATION

Notification sent when a new ResourceOrder resource is created.

### Json representation sample

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

```

{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ResourceOrderCreationNotification",
  "event": {
    "resourceOrder" :
  
```

```
    {  
      {  
        {  
          {  
            {  
              {  
                {  
                  {  
                    {  
                      {  
                        {  
                          {  
                        }  
                      }  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

---

## RESOURCE ORDER ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when changing an attribute of a ResourceOrder resource.

### Json representation sample

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

```
{  
  "eventId":"00001",  
  "eventTime":"2015-11-16T16:42:25-04:00",  
  "eventType":"ResourceOrderAttributeValueChangeNotification",  
  "event": {  
    "resourceOrder" :  
      {  
        {  
          {  
            {  
              {  
                {  
                  {  
                    {  
                      {  
                        {  
                          {  
                        }  
                      }  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

---

## RESOURCE ORDER STATE CHANGE NOTIFICATION

Notification sent when changing the state of a ResourceOrder resource.

### Json representation sample

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

```
{  
  "eventId":"00001",  
  "eventTime":"2015-11-16T16:42:25-04:00",  
  "eventType":"ResourceOrderStateChangeNotification",  
  "event": {  
    "resourceOrder" :  
      {  
        {  
          {  
            {  
              {  
                {  
                  {  
                    {  
                      {  
                        {  
                          {  
                        }  
                      }  
                    }  
                  }  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

---

## RESOURCE ORDER REMOVE NOTIFICATION

Notification sent when removing a ResourceOrder resource.

### Json representation sample

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

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

---

## RESOURCE 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: requestedStartDate contact is missing

```
{
  "eventId":"00005",
  "eventTime":"2013-04-19T16:42:25-30:00",
  "eventType":"orderInformationRequiredNotification",
  "resourcePath":"/order/42 ",
  "fieldPath":"missing=requestedStartDate",
  "resourceOrder":{
    "id":" 42",
    "href":"http://serverlocation:port/resourceOrderingManagement/resourceOrder/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 RESOURCE ORDER

### LIST RESOURCE ORDERS

#### **GET /resourceOrder?fields=...&{filtering}**

##### Description

This operation list resource 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 ResourceOrder resources.

Searching resource orders started on January 1st 2015. The result items are shrinked to show only the ids (fields=id)

Request
GET /resourceOrderingManagement/resourceOrder?fields=id&orderDate="2015-01-01" Accept: application/json
Response
200 <pre>[   {     "id": "986781"   },   {     "id": "986782"   },   {     "id": "986783"   } ]</pre>

### RETRIEVE RESOURCE ORDER

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

## Description

This operation retrieves a resource 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 an example of a request for retrieving a ResourceOrder resource.

Request
GET /resourceOrderingManagement/resourceOrder/42 Accept: application/json
Response
<pre> 200  {   "id": "42",   "href": "http://serverlocation:port/resourceOrderingManagement/resourceOrder/42",   "corelationID": "MyResourceOrder_42",   "priority": "1",   "description": "A wonderful 42 order for brand new Resources",   "category": "Uncategorized",   "state": "InProgress",   "orderDate": "2013-04-12T16:42:23-04:00",   "requestedStartDate": "2013-04-12T16:42:23-04:00",   "requestedCompletionDate": "2013-04-19T16:42:23-04:00",   "completionDate": "2013-04-19T16:42:23-04:00",   "orderRelationship": [{     "type": "dependency",     "href": "https://host:port/ resourceOrderingManagement /resourceOrder/3304",     "id": "3304"   }, {     "type": "cross-ref",     "href": "https://host:port/ serviceOrderingManagement /serviceOrder/3304",     "id": "3304"   }   ],   "note": {     "text": "A free text detailing the note",     "date": "2013-04-12T16:42:23-04:00",     "author": "name"   } }           </pre>

```

    }},
    "relatedParty": [{
      "role": "owner",
      "id": "345221",
      "name": "John Doe"
    }],
  },
  "resourceOrderItem": [{
    "id": "1",
    "action": "add",
    "state": "Acknowledged",
    "appointment": {
      "href": "http://serverlocation:port/appointment/100"
    },
  },
  "resourceSpecification": {
    "id": "42",
    "href": "http://serverlocation:port/resourceCatalogManagement/resourceSpecification/42"
  },
  "resource": {
    "place": {
      "href": "http://map.google.com/.../1234112GDE",
      "role": "DeliveryPlace"
    },
    "resourceCharacteristic": [{
      "name": "Colour",
      "value": "White"
    }, {
      "name": "Memory",
      "value": "16"
    }
  ]
}, {
  "id": "2",
  "action": "modify",
  "state": "InProgress",
  "resourceSpecification": {
    "id": "43",
    "href": "http://serverlocation:port/resourceCatalogManagement/resourceSpecification/43"
  },
  "resource": {
    "id": "456",
    "href": "http://serverlocation:port/resourceInventoryManagement/logicalResource/456",
    "resourceCharacteristic": {
      "name": "anotherCharacteristic",
      "value": "itsValue"
    },
    "relatedParty": [{
      "role": "owner",
      "id": "5667443",
      "name": "Jimmy Doe"
    }
  ]
}

```

```

    }
  }, {
    "id": "3",
    "action": "add",
    "state": "InProgress",
    "resourceSpecification": {
      "id": "43",
      "href": "http://serverlocation:port/resourceCatalogManagement/resourceSpecification/51"
    },
    "resource": {
      "resourceRelationship": {
        "type": "reliesOn",
        "resource": {
          "href": "411",
          "id": "http://serverlocation:
port/resourceInventoryManagement/logicalResource/411"
        }
      }
    }
  }
}
}
}

```

## CREATE RESOURCE ORDER

### POST /resourceOrder

#### Description

This operation creates a resource order entity.

#### Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
orderItem	

Non Mandatory Attributes	Default Value	Rule
category	Uncategorized	
completionDate		
corelationId		
description		
name		
orderDate		
priority	4 (lowest)	
requestedCompletionDate		
requestedStartDate		



state	Acknowledged	
type		
orderRelationship		
relatedParty		
note		
orderItem.state	Acknowledged	
orderItem.ResourceSpecification		
orderItem.resource.place		
orderItem.appointment		

### Additional Rules

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

Context	Mandatory Sub-Attributes
note	text
appointment	id OR href
orderItem	id, action, resource
relatedParty	role, id OR href OR name
orderItem.resource.place	role, id OR href
orderItem.resourceSpecification	id OR href
orderItem.resource	characteristic (if action==add), id OR href (if action==modify or action==delete)

The following pre-conditions apply for this operation.

Pre-conditions
PATCH allowed if order not completed

### 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 an example of a request for creating a ResourceOrder resource composed of 2 order items:

- Line 1: for ordering a new resource that needs a physical place and an appointment to be delivered;
- Line 2: change of a characteristic value of an already owned Resource and change the user associated with this Resource

## Request

POST /resourceOrderingManagement/resourceOrder

Content-Type: application/json

```
"note": [{
    "text": "A free text detailing a note for the resource order"
  }],
"relatedParty": [{
  "role": "owner",
  "id": "345221",
  "name": "John Doe"
}],
"resourceOrderItem": [{
  "id": "1",
  "action": "add",

  "appointment": {
    "href": "http://serverlocation:port/appointment/100",

    },
    "resourceSpecification": {
      "href": "http://serverlocation:port/resourceCatalog/resourceSpecification/42"
    },
    "resource": {
"place": {
  "href": "http://map.google.com/.../1234112GDE",
  "role": "DeliveryPlace"
},
    "resourceCharacteristic": [{
      "name": "Colour",
      "value": "White"
    }, {
      "name": "Memory",
      "value": "16"
    }
  ]
}, {
  "id": "2",
  "action": "modify",
  "resource": {
    "href": "http://serverlocation:port/resourceInventoryManagement/logicalResource/456",
    "relatedParty": [{
      "role": "owner",
      "id": "5667443",
      "name": "Jimmy Doe"
    }
  ]
}
}
```

```

    }
  },
}

```

## Response

201

```

{
  "id": "42",
  "href": "http://serverlocation:port/ resourceOrderingManagement /resourceOrder/42",
  "priority": "1",
  "state": "Acknowledged",
  "orderDate": "2013-04-12T16:42:23-04:00",
  "note": [{
    "text": "A free text detailing a note for the resource order"
  }],
  "relatedParty": [{
    "role": "owner",
    "id": "345221",
    "name": "John Doe"
  }],
  "resourceOrderItem": [{
    "id": "1",
    "action": "add",
    "state": "Aknowledged",

    "appointment": {
      "href": "http://serverlocation:port/appointment/100"
    },
    "resourceSpecification": {
      "href": "http://serverlocation: port/resourceCatalogManagement/resourceSpecification/42"
    },
    "resource": {
      "href": "http://map.google.com/.../1234112GDE",
      "role": "DeliveryPlace"
    },
    "resourceCharacteristic": [{
      "name": "Colour",
      "value": "White"
    }, {
      "name": "Memory",
      "value": "16"
    }
  ]
}
}, {
  "id": "2",
  "action": "modify",

```

```

    "state": "Acknowledged",
    "resource": {
      "href": "http://serverlocation:port/resourceInventoryManagement/logicalResource/456",
      "relatedParty": [{
        "role": "owner",
        "id": "5667443",
        "name": "Jimmy Doe"
      }]
    }
  }
}

```

## PATCH RESOURCE ORDER

### PATCH /resourceOrder/{id}

#### Description

This operation allows partial updates of a resource 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	
name	
priority	
requestedCompletionDate	Only when order is in "Acknowledged" state – delivery process not started
requestedStartDate	Only when order is in "Acknowledged" state – delivery process not started
state	To manage the order delivery process : Acknowledged, InProgress (start process), Held / Pending (suspend process), Cancelled Influence the orderItem states
type	
orderRelationship	
relatedParty	Only when order is in "Acknowledged" state – delivery process not started
note	
orderItem.resourceSpecification	Only when order is in "Acknowledged" state – delivery process not started

	started or suspended
orderItem.resource.place	Only when order is in “Acknowledged” state – delivery process not started or suspended
orderItem.resource	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	
corelationId	
orderDate	
completionDate	
orderItem.id	
orderItem.action	

### Additional Rules

The following pre-conditions apply for this operation.

Pre-conditions
PATCH allowed if order not completed

### Usage Samples

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

Changing the priority of the order (using json-merge)

Request
PATCH /resourceOrderingManagement/resourceOrder/42 Content-Type: application/merge-patch+json  <pre>{   "priority": 1 }</pre>
Response
201  { Similar JSON as in GET response with priority changed }

---

## DELETE RESOURCE ORDER

**DELETE** `/resourceOrder/{id}`

*Note: this operation is available only to ADMIN API users*

### Description

This operation deletes a resource order entity.

### Usage Samples

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

<b>Request</b>
DELETE /resourceOrderingManagement/resourceOrder/42
<b>Response</b>
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.

<b>Request</b>
DELETE /api/hub/42 Accept: application/json
<b>Response</b>
204

## PUBLISH EVENT TO LISTENER

### **POST /client/listener**

#### **Description**

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

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

#### **Behavior**

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

#### **Usage Samples**

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



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

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

## ACKNOWLEDGEMENTS

### RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	04/15/2016	Pierre Gauthier TM Forum <a href="mailto:pgauthier@tmforum.org">pgauthier@tmforum.org</a>  Nicoleta Stoica Vodafone  Mariano Belaunde Orange Labs	Generated using the API data model