**TM Forum Specification**

**XXX Management API REST Specification**

**TMFXXX**

**Release 19.0.0**

**June 2019**

|  |  |
| --- | --- |
| **Latest Update: TM Forum Release 19.0.0** | **Member Evaluation** |
| **Version 4.0.0** | **IPR Mode: RAND** |

NOTICE

Copyright © TM Forum 2019. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](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. +1 973 998 7196

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

Table of Contents

[NOTICE 2](#_Toc514836422)

[Table of Contents 3](#_Toc514836423)

[List of Tables 4](#_Toc514836424)

[Introduction 5](#_Toc514836425)

[SAMPLE USE CASES 6](#_Toc514836426)

[Support of polymorphism and extension patterns 7](#_Toc514836427)

[RESOURCE MODEL 8](#_Toc514836428)

[Managed Entity and Task Resource Models 8](#_Toc514836429)

[FIRST resource 8](#_Toc514836430)

[Notification Resource Models 8](#_Toc514836431)

[First Notification 8](#_Toc514836432)

[API OPERATIONS 9](#_Toc514836433)

[VERB url 9](#_Toc514836434)

[API NOTIFICATIONS 10](#_Toc514836435)

[Register listener 10](#_Toc514836436)

[Unregister listener 11](#_Toc514836437)

[Publish Event to listener 11](#_Toc514836438)

[Acknowledgements 13](#_Toc514836439)

[Release History 13](#_Toc514836440)

[Contributors to Document 13](#_Toc514836441)

List of Tables

N/A

Introduction

The following document is the specification of the REST API for Any management. It includes the model definition as well as all available operations..

# 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 patterns

Support of polymorphic collections and types and schema based extension is provided by means of a list of generic meta-attributes that we describe below. Polymorphism in collections occurs when entities inherit from base entities, for instance a BillingAccount and SettlementAccount inheriting properties from the abstract Account 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 Account instances some may be instances of BillingAccount where other could be instances of SettlementAccount. 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 an AccountRef 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 BillingAccountRef or SettlementAccountRef, 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.

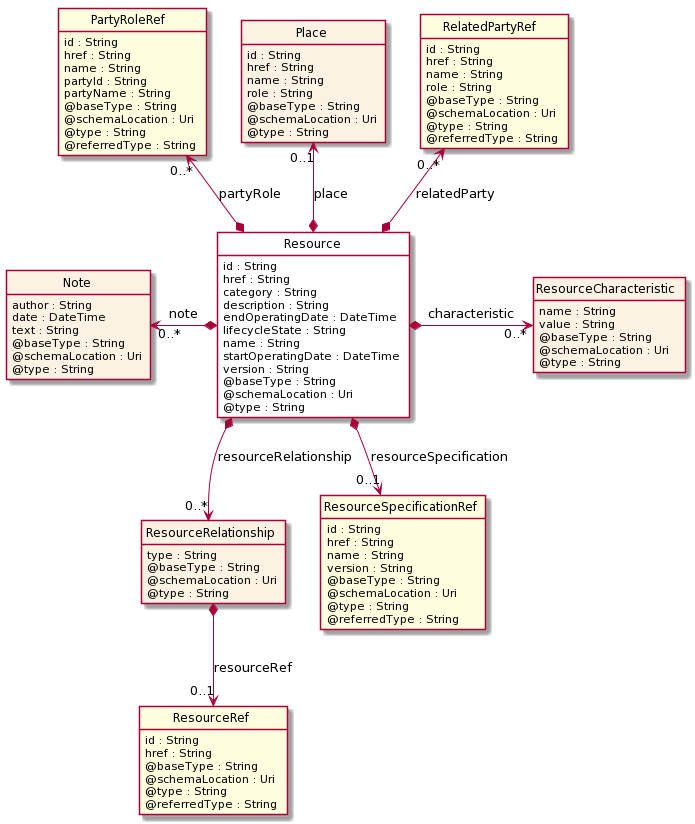
# RESOURCE MODEL

## Managed Entity and Task Resource Models

### Resource resource

Resource is an abstract entity that describes the common set of attributes shared by all concrete resources (e.g. TPE, EQUIPMENT) in the inventory.

**Resource model**



**Field descriptions**

*Resource* fields

|  |  |
| --- | --- |
| category | A string. Category of the concrete resource. e.g Gold, Silver for MSISDN concrete resource. |
| characteristic | A list of resource characteristics (ResourceCharacteristic [\*]). |
| description | A string. free-text description of the resource. |
| endOperatingDate | A date time (DateTime). A date time( DateTime). The date till the resource is operating. |
| href | A string. The URI for the object itself. |
| id | A string. Identifier of an instance of the resource. Required to be unique within the resource type. Used in URIs as the identifier for specific instances of a type. |
| lifecycleState | A string. The life cycle state of the resource. |
| name | A string. A string used to give a name to the resource. |
| note | A list of notes (Note [\*]). Extra information about a given entity. |
| partyRole | A list of party role references (PartyRoleRef [\*]). A party role represents the part played by a party in a given context. |
| place | A place (Place). Place reference. Place defines the places where the products are sold or delivered. |
| relatedParty | A list of related party references (RelatedPartyRef [\*]). A related party defines party or party role linked to a specific entity. |
| resourceRelationship | A list of resource relationships (ResourceRelationship [\*]). |
| resourceSpecification | A resource specification reference (ResourceSpecificationRef). The ResourceSpecification is required to realize a ProductSpecification. |
| startOperatingDate | A date time (DateTime). A date time( DateTime). The date from which the resource is operating. |
| version | A string. A field that identifies the specific version of an instance of a resource. |

*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. |
| 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 (for instance: 'home delivery', 'shop retrieval'). |

*ResourceCharacteristic* sub-resource

|  |  |
| --- | --- |
| name | A string. name of the characteristic. |
| value | A string. value of the characteristic. |

*ResourceRelationship* sub-resource

|  |  |
| --- | --- |
| resourceRef | A resource reference (ResourceRef). |
| type | A string. |

*PartyRoleRef* relationship

Party role reference. A party role represents the part played by a party in a given context.

|  |  |
| --- | --- |
| @referredType | A string. The actual type of the target instance when needed for disambiguation. |
| href | A string. Reference of the product. |
| id | A string. Unique identifier of the product. |
| name | A string. The name of the referred party role. |
| partyId | A string. The identifier of the engaged party that is linked to the PartyRole object. |
| partyName | A string. The name of the engaged party that is linked to the PartyRole object. |

*RelatedPartyRef* relationship

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

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

*ResourceRef* relationship

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

*ResourceSpecificationRef* relationship

Resource Specification reference: The ResourceSpecification is 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 resource specification. |
| id | A string. Unique identifier of the resource specification. |
| name | A string. Name of the requiredResourceSpecification. |
| version | A string. Resource specification version. |

**Json representation sample**

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

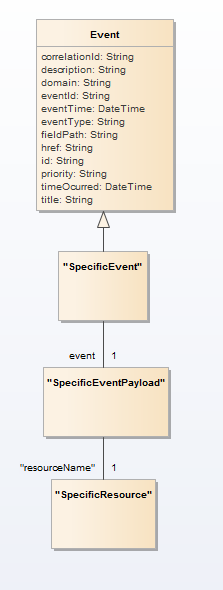
|  |
| --- |
| {  "category": "a string ...",  "characteristic": [  {}  ],  "description": "This resource ...",  "endOperatingDate": "2019-07-04T00:00",  "href": "https:/host:port/tmf-api/resource/v1/resource/8031",  "id": "8031",  "lifecycleState": "a string ...",  "name": "a string ...",  "note": [  {}  ],  "partyRole": [  {}  ],  "place": {},  "relatedParty": [  {}  ],  "resourceRelationship": [  {}  ],  "resourceSpecification": {},  "startOperatingDate": "2019-07-04T00:00",  "version": "a string ..." } |

## Notification Resource Models

3 notifications are defined for this API

Notifications related to Resource:  
 - ResourceCreateEvent  
 - ResourceAttributeValueChangeEvent  
 - ResourceDeleteEvent

The notification structure for all notifications in this API follow the pattern depicted by the figure below.  
A notification event resource (depicted by "SpecificEvent" placeholder) is a sub class of a generic Event structure containing at least an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the resource (eventType).   
This notification structure owns an event payload structure ("SpecificEventPayload" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



### Resource Create Event

Notification ResourceCreateEvent case for resource Resource

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ResourceCreateEvent",  "event": {  "resource" :   {-- SEE Resource RESOURCE SAMPLE --}  } } |

### Resource Attribute Value Change Event

Notification ResourceAttributeValueChangeEvent case for resource Resource

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ResourceAttributeValueChangeEvent",  "event": {  "resource" :   {-- SEE Resource RESOURCE SAMPLE --}  } } |

### Resource Delete Event

Notification ResourceDeleteEvent case for resource Resource

**Json representation sample**

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

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"ResourceDeleteEvent",  "event": {  "resource" :   {-- SEE Resource 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 Resource

### List resources

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

**Description**

This operation list resource 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 Resource resources.

|  |
| --- |
| **Request** |
| GET {apiRoot}/resource Accept: application/json |
| **Response** |
| 200  [ {  "category": "a string ...",  "characteristic": [  {}  ],  "description": "This resource ...",  "endOperatingDate": "2019-07-04T00:00",  "href": "https:/host:port/tmf-api/resource/v1/resource/8031",  "id": "8031",  "lifecycleState": "a string ...",  "name": "a string ...",  "note": [  {}  ],  "partyRole": [  {}  ],  "place": {},  "relatedParty": [  {}  ],  "resourceRelationship": [  {}  ],  "resourceSpecification": {},  "startOperatingDate": "2019-07-04T00:00",  "version": "a string ..." } ] |

### Retrieve resource

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

**Description**

This operation retrieves a resource 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 Resource resource.

|  |
| --- |
| **Request** |
| GET {apiRoot}/resource/8031 Accept: application/json |
| **Response** |
| 200  {  "category": "a string ...",  "characteristic": [  {}  ],  "description": "This resource ...",  "endOperatingDate": "2019-07-04T00:00",  "href": "https:/host:port/tmf-api/resource/v1/resource/8031",  "id": "8031",  "lifecycleState": "a string ...",  "name": "a string ...",  "note": [  {}  ],  "partyRole": [  {}  ],  "place": {},  "relatedParty": [  {}  ],  "resourceRelationship": [  {}  ],  "resourceSpecification": {},  "startOperatingDate": "2019-07-04T00:00",  "version": "a string ..." } |

### Create resource

**POST /resource**

**Description**

This operation creates a resource entity.

**Mandatory and Non Mandatory Attributes**

The following tables provide the list of mandatory and non mandatory attributes when creating a Resource, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

|  |  |
| --- | --- |
| Mandatory Attributes | Rule |
| name |  |

|  |  |
| --- | --- |
| Non Mandatory Attributes | Rule |
| category |  |
| characteristic |  |
| description |  |
| endOperatingDate |  |
| lifecycleState |  |
| note |  |
| partyRole |  |
| place |  |
| relatedParty |  |
| resourceRelationship |  |
| resourceSpecification |  |
| startOperatingDate |  |
| version |  |

**Usage Samples**

Here's an example of a request for creating a Resource resource. In this example the request only passes mandatory attributes.

|  |
| --- |
| **Request** |
| POST {apiRoot}/resource Content-Type: application/json  {  "name": "a string ..." } |
| **Response** |
| 201  {  "href": "https:/host:port/tmf-api/resource/v1/resource/8031",  "id": "8031",  "name": "a string ..." } |

### Patch resource

**PATCH /resource/{id}**

**Description**

This operation allows partial updates of a resource 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 |  |
| characteristic |  |
| description |  |
| endOperatingDate |  |
| lifecycleState |  |
| name |  |
| note |  |
| partyRole |  |
| place |  |
| relatedParty |  |
| resourceRelationship |  |
| resourceSpecification |  |
| startOperatingDate |  |
| version |  |

|  |  |
| --- | --- |
| Non Patchable Attributes | Rule |
| href |  |
| id |  |

**Usage Samples**

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

|  |
| --- |
| **Request** |
| PATCH {apiRoot}/resource/8031 Content-Type: application/merge-patch+json  {  "name": "new name" } |
| **Response** |
| 200  {  "category": "a string ...",  "characteristic": [  {}  ],  "description": "This resource ...",  "endOperatingDate": "2019-07-04T00:00",  "href": "https:/host:port/tmf-api/resource/v1/resource/8031",  "id": "8031",  "lifecycleState": "a string ...",  "name": "new name",  "note": [  {}  ],  "partyRole": [  {}  ],  "place": {},  "relatedParty": [  {}  ],  "resourceRelationship": [  {}  ],  "resourceSpecification": {},  "startOperatingDate": "2019-07-04T00:00",  "version": "a string ..." } |

### Delete resource

**DELETE /resource/{id}**

**Description**

This operation deletes a resource entity.

**Usage Samples**

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

|  |
| --- |
| **Request** |
| DELETE {apiRoot}/resource/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

## Release History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release Number** | **Date** | **Release led by:** | **Description** |
| Release 1.0 | 04/15/2017 | Pierre Gauthier TM Forum [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Mariano Belaunde Orange Labs | First Release of the Document. |
| Release 2.0 | 11/06/2018 | Mariano Belaunde  Orange Labs | Alignment with Guidelines 3.0 |

## Contributors to Document

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