

# TM Forum Specification

## Recommendation API REST Specification

**TMF680**  
**Release 18.0.0**  
**June 2018**

<b>Latest Update: TM Forum Release 18.0.0</b>	<b>Member Evaluation</b>
<b>Version 2.0.1</b>	<b>IPR Mode: RAND</b>

**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](#), 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
TABLE OF CONTENTS .....	3
LIST OF TABLES .....	4
INTRODUCTION .....	5
MAPPING WITH SID ABE .....	6
MAPPING WITH ETOM PROCESS.....	6
DISTINCTION BETWEEN THIS API AND OTHER EXISTING TMF APIS.....	6
SAMPLE USE CASES.....	8
Support of polymorphism and extension patterns .....	9
RESOURCE MODEL.....	10
Managed Entity and Task Resource Models.....	10
Communication Message resource model .....	10
Field descriptions .....	10
JSON representation sample.....	15
API OPERATIONS.....	17
GET /recommendation?fields=...&{filtering}.....	18
API NOTIFICATIONS .....	20
Register listener .....	20
Unregister listener .....	21
Publish Event to listener .....	21
ACKNOWLEDGEMENTS .....	23
Release History .....	23
Version History .....	23
Contributors to Document .....	24

## LIST OF TABLES

N/A

## INTRODUCTION

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

Recommendation API manages the following data resources:

- **Recommendation**
  - o Recommendation is a type of automated (typically, conditional) system action to determine which products offerings to be presented as up-sells, cross-sells, and related products to each customer segment based on customer and session specific information. The design-time offers and product recommendations may come from Marketing and Catalog. The run-time evaluation and presentment will be executed with contextual/session information. Normally, it supports the recommendation of the offer which is proper for the customer, provides Up-sell and cross-sell based on contextual and catalog rules.

Recommendation API is used to recommend offering quickly based on the history and real-time context of customer. It is a real-time and personalized recommendation API. It is usually provided by e-commerce or BSS, CRM system in omni-channel.

The typical example of recommendation is on the online e-commerce site.

The screenshot displays a user interface for a mobile service provider. At the top, there are navigation links: Shop, Explore, My Telco+, and Support. Below this, there are several sections:

- LAST BILL:** Shows a bill amount of €35.00 with a due date of Sep 10, 2016.
- PAYMENT STATUS:** Shows a green checkmark and the text "Great! There's no arrears!".
- RECOMMENDATION (HOT):** A red-bordered box highlights two recommendation cards. The first is for a "Red Maxi" phone priced at €49.9. The second is for a "Wal-Mart Carrefour" gift card worth €100.
- RECHARGE:** A section with buttons for recharging amounts: 10, 20, 30, 50, 100, 150, 200, and Other. A "Recharge" button is also present.
- USAGE:** A section showing usage statistics for Mobile Voice (0 Lst), Data (8192 MB Lst), Message (0 Lst), and Wlan (0 Lst). A banner below this section reads "Mobile Voice is not enough? Just check the recommendation on the right >".
- BONUS TRAFFIC:** A section with a "4G Open Click" button and the text "You have a chance to grab traffic."

---

## MAPPING WITH SID ABE

Recommendation is related to “*Product Domain::Product Offering ABE*” in TMF Information Framework (SID).

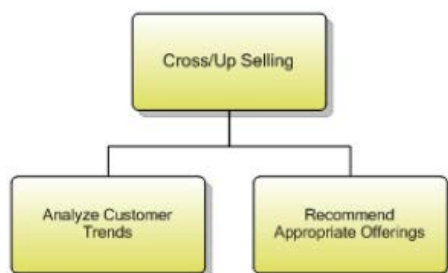
---

## MAPPING WITH ETOM PROCESS

In Business Process Framework there is the description for recommendation:

- Level 2 Process: 1.1.9 Selling
- Level 3 Process: 1.1.9.3 Cross/Up Selling
- Level 4 Process: 1.1.9.3.2 Recommend Appropriate Offerings

The description of “Recommend Appropriate Offerings” is to recommend the appropriate offering to the customer. It is the important approach to attract the customer and propel more revenue increase.



---

## DISTINCTION BETWEEN THIS API AND OTHER EXISTING TMF APIS

Here the differences between Recommendation API and other existing published TMF APIs are explained to clarify why this separate API is not covered simply with those APIs.

### ✓ **Difference with Product Catalog API**

Recommendation and Product Offering in the Product Catalog has some similarities. The product offering is the object which is recommended to the customer.

Product Catalog API focuses on the configuration of product offering. When querying with Product Catalog API, all the existing offerings will be included in the query result. Instead, recommendation API only fetches the offerings which possibly cause the interest of the customer.

## Recommendation API REST Specification

✓ **Difference with Product Offering Qualification API**

Product Offering Qualification acquires the validated and available product offering for the customer to purchase. These offerings are all the allowed options for the customer.

Recommendation API provides the most possibly-chosen offering for the customer. Such offerings are selected for the customer as the first and primary choice, not only the normal sellable objects.

✓ **Difference with Shopping Cart API**

The Shopping Cart API is a container to load the selected offerings for the customer to purchase. It does not replace the recommendation.

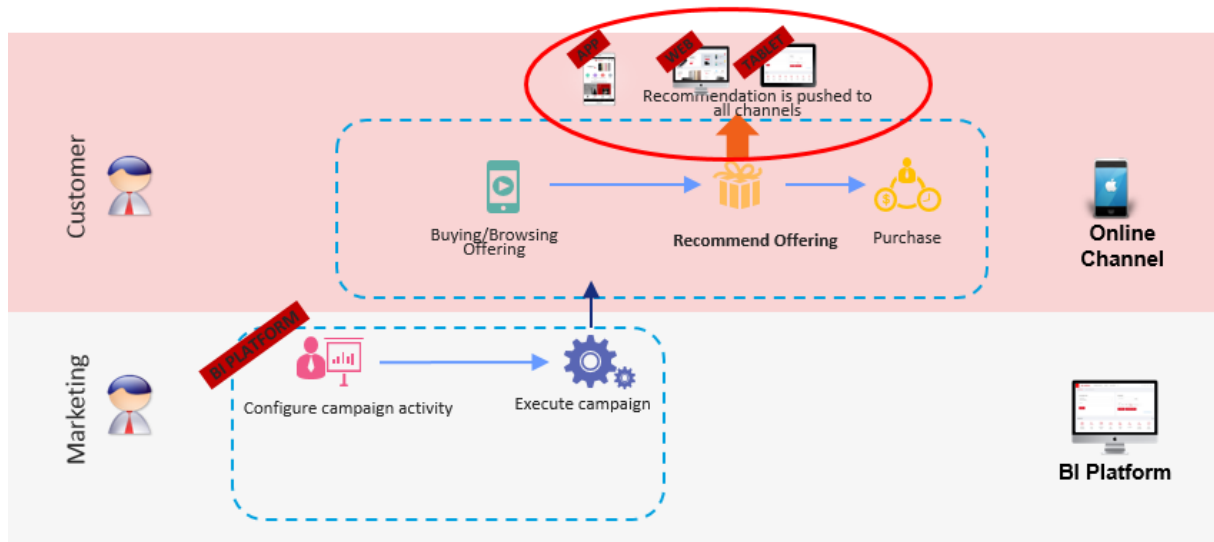
The recommendation often accompanies with the shopping cart. Based on the selected offerings in the cart, the recommended offerings are shown nearby. But the recommendation is not done by the shopping cart itself.

The screenshot displays a shopping cart interface. On the left, under the heading "Shopping Cart", there is a table with columns for "Price" and "Quantity". A single item is listed: "Sapiens: A Brief History of Humankind" by Yuval Noah Harari, priced at \$15.17 with a quantity of 1. The item details include "Paperback", "Available for Pre-order", and "Eligible for FREE Shipping". There are also options to "Delete" or "Save for later".

On the right, a recommendation box titled "Frequently Bought With Sapiens: A Brief History of Humankind" is highlighted with a red border. It features a book cover for "Homo Deus: A Brief History of Humankind" by Yuval Noah Harari, with a 5-star rating and 131 reviews. A button labeled "See all buying options" is located at the bottom of the recommendation box.

## SAMPLE USE CASES

Examples of use cases using Recommendation API is as following





## 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.

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. All resources and sub-resources of this API have a `@type` attributes that can be provided when this is useful. Such as `Recommendation`, `RecommendationItem`. 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 `RelatedPartyRef`, `ChannelRef`, `CategoryRef`, `ProductOfferingRef`, `ProductOrderRef`, `ShoppingCartRef`) to explicitly denote the actual entity type of the referred class. Notice that in reference entities the `@type`, when used, denotes the class types of the reference itself, such as `RelatedPartyRef`, 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.

Notice that because these meta-attributes have a generic meaning we will not repeat their definition in the resource description tables of each resource and each sub-resource.

**RESOURCE MODEL**

Managed Entity and Task Resource Models

COMMUNICATION MESSAGE RESOURCE MODEL

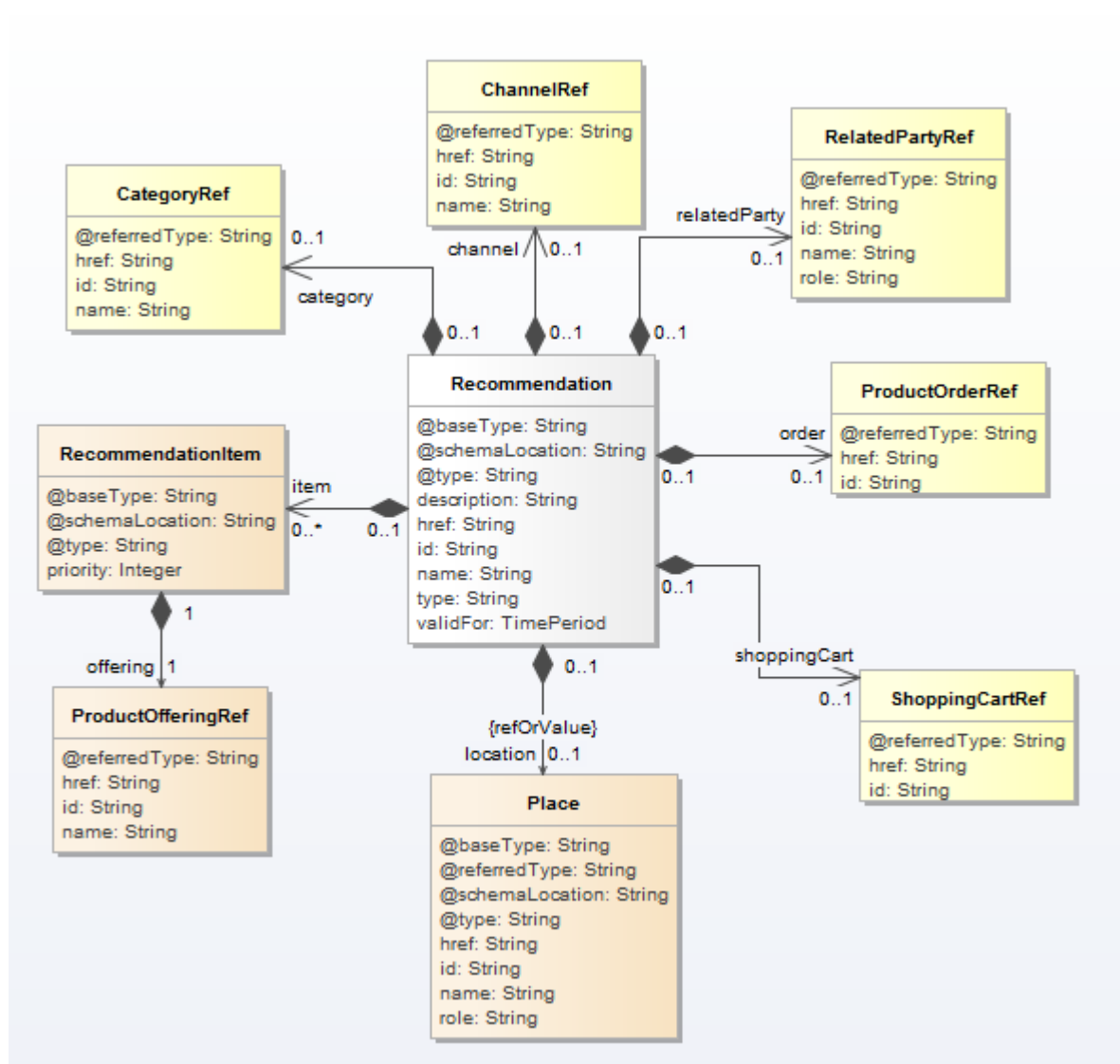


Fig.1. Recommendation Resource

**FIELD DESCRIPTIONS**

**Recommendation**

## Recommendation API REST Specification

Fields	Data Type	Description
@type	String	Indicates the type of resource. Here can be recommendation resource.
@schemaLocation	String	It provides the link to the schema describing REST resource
@baseType	String	It indicates the base type of REST resource.
description	String	Description of recommendation
id	String	Unique identifier of recommendation
href	String	Hypertext Reference of the recommendation.
name	String	Name of recommendation
validFor	TimePeriod	The period in which the recommendation is valid.
type	String	Category of recommendation.  The basic type is:  'AD': it means the recommendation is the advertisement for display  'OFFER': it means the recommended content is the offer entry page. By clicking it, the user can be forwarded to the details of the offering
item	RecommendationItem	Recommendation items. Every item is a product offering and its priority.
category	CategoryRef	Different kinds of recommendation. For example, it can be used to describe different recommendation positions on the e-commerce web site.
shoppingCart	ShoppingCartRef	The shopping cart which the recommendation is related with.
channel	ChannelRef	The channel where the recommendation is used. May be online web, mobile app, social, etc.
order	ProductOrderRef	The product order which the recommendation is related with.
location	Place	The geographic location which the recommendation is related with.

## Recommendation API REST Specification

Fields	Data Type	Description
relatedParty	RelatedParty Ref	The party which the recommendation is related with.

**RecommendationItem**

Recommendation items. Every item is a product offering and its priority.

Fields	Type	Description
@type	String	Indicates the type of resource.
@schemaLocation	String	It provides the link to the schema describing REST resource
@baseType	String	It indicates the base type of REST resource.
priority	Integer	Priority level for applying this alteration among all the defined alterations.
offering	ProductOfferingRef	Recommended Product offering

**CategoryRef**

Different kinds of recommendation. For example, it can be used to describe different recommendation positions on the e-commerce web site.

Fields	Type	Description
@referredType	String	Indicates the type of resource.
id	String	Unique identifier of category
href	String	Hypertext Reference of the category.
name	String	Name of the category.

**ShoppingCartRef**

The shopping cart which the recommendation is related with.

## Recommendation API REST Specification

Fields	Type	Description
@referredType	String	Indicates the type of resource.
Id	String	Unique identifier of shopping cart
href	String	Hypertext Reference of the shopping cart.

**ChannelRef**

The channel where the recommendation is used. May be online web, mobile app, social, etc.

Fields	Type	Description
@referredType	String	Indicates the type of resource.
Id	String	Unique identifier of channel
href	String	Hypertext Reference of the channel.
name	String	Name of the channel.

**ProductOrderRef**

The product order which the recommendation is related with.

Fields	Type	Description
@referredType	String	Indicates the type of resource.
Id	String	Unique identifier of product order
href	String	Hypertext Reference of the product order.

**Place**

The location which the recommendation is related with.

Fields	Type	Description
@type	String	Indicates the type of resource.
@schemaLocation	String	It provides the link to the schema describing REST resource
@baseType	String	It indicates the base type of REST resource.
@referredType	String	Indicates the type of resource.
id	String	Unique identifier of the location
href	String	Hypertext Reference of the location
name	String	Name of the location.
type	String	Type of the location.

**ProductOfferingRef**

Recommended Product offering

Fields	Type	Description
@referredType	String	Indicates the type of resource.
id	String	Unique identifier of product offering
href	String	Hypertext Reference of the product offering.
name	String	Name of the product offering.

**RelatedPartyRef**

The party which the recommendation is related with.

Fields	Type	Description
@referredType	String	Indicates the type of resource.
Id	String	Unique identifier of related party
href	String	Hypertext Reference of the related party.
name	String	Name of the related party.
role	String	Role of the related party.

---

**JSON REPRESENTATION SAMPLE**

We provide below the JSON representation of an example of Communication Message Resource object:

```
{
  "id": "1001",
  "href": "http://serverlocation:port/recommendation/v1/recommendation/1001",
  "name": "recommendation of the latest Apple iPhone",
  "description": "recommendation of the latest Apple iPhone for the customer with
high revenue contribution",
  "@type": "recommendation",
  "@schemaLocation": "http://serverlocation:port/recommendation/schema/
recommendation.yml",
  "@baseType": "",
  "item": [
    {
      "priority": 1,
      "offering": {
        "href":
"https://host:port/productOffering/v1/productOfferings/6547",
        "id": "6547",
```

```
        "name": "phone1"
      }
    },
    {
      "priority": 2,
      "offering": {
        "href":
"https://host:port/productOffering/v1/productOfferings/6547",
        "id": "6542",
        "name": " phone2"
      }
    }
  ],
  "validFor": {
    "startDateTime": "2017-12-19 T04:00:00.OZ",
    "endDateTime": "2017-12-31 T20:42:23.OZ"
  },
  "type": "OFFER",
  "channel": {
    "id": "13",
    "href": "http://serverlocation:port/recommendation/v1/channel/13",
    "name": "mobile app channel"
  },
  "relatedParty": {
    "id": "34",
    "href": "http://serverlocation:port/partyManagement/v1/individual/34",
    "name": "John Smith",
    "role": ""
  }
  "location": {
    "id": "334"
  }
}
```



## 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.

## GET /recommendation?fields=...&amp;{filtering}

**Description:**

- This operation is used to query recommendations by query conditions
- This operation is usually used by e-commerce client. When the client query recommendation, it will trigger the server to generate the recommendation quickly by Big Data analysis tech. or AI, Machine Learning tech. The server will response with the recommendation result.
- Attribute selection is enabled for all first level attributes.
- Attribute Filtering may be available depending on the compliance level supported by an implementation.

For example:

1. Get recommendations by relatedParty id and channel id

*GET /recommendation?relatedParty.id=VALUE&channel.id=VALUE*

2. Get recommendations by relatedParty id, channel id and shoppingcart id. Usually, we need to recommend specific offering when the customer change his shopping cart.

*GET /recommendation?relatedParty.id=VALUE&shoppingCart.id=120*

3. Get recommendations by relatedParty id, channel id and location id. Usually, we need to recommend specific offering when the customer's location changes.

*GET /recommendation?relatedParty.id=VALUE& location.id=120*

**Usage Samples**

Here's an example of a request for retrieving Recommendation resources.

REQUEST
GET /recommendation?relatedParty.id=10023&channel.id=2053 Content-type: application/json Accept: application/json
RESPONSE
<pre>{   "id": "1001",   "href": "http://serverlocation:port/recommendation/v1/recommendation/1001",   "name": "recommendation of the latest Apple iPhone",   "description": "recommendation of the latest Apple iPhone for the customer with high revenue contribution",   "@type": "recommendation",</pre>

```
"@schemaLocation": "http://serverlocation:port/recommendation/schema/
recommendation.yml",
"@baseType": "",
"item": [
  {
    "priority": 1,
    "offering": {
      "href": "https://host:port/productOffering/v1/productOfferings/6547",
      "id": "6547",
      "name": "phone1"
    }
  },
  {
    "priority": 2,
    "offering": {
      "href": "https://host:port/productOffering/v1/productOfferings/6547",
      "id": "6542",
      "name": " phone2"
    }
  }
],
"validFor": {
  "startDateTime": "2017-10-9 T04:00:00.0Z",
  "endDateTime": "2017-10-9 T20:42:23.0Z"
},
"type": "OFFER",
"channel": {
  "id": "2503",
  "href": "http://serverlocation:port/recommendation/v1/channel/13",
  "name": "mobile app channel"
},
"relatedParty": {
  "id": "10023",
  "href": "http://serverlocation:port/partyManagement/v1/individual/34",
  "name": "John Smith",
  "role": ""
}
}
```

## 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.

<b>Request</b>
<pre>POST /api/hub Accept: application/json  {"callback": "http://in.listener.com"}</pre>
<b>Response</b>
<pre>201 Content-Type: application/json Location: /api/hub/42  {"id": "42", "callback": "http://in.listener.com", "query": null}</pre>

## UNREGISTER LISTENER

### DELETE /hub/{id}

#### Description

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

#### Behavior

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

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

#### Usage Samples

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

<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.

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
17.5.0	January 2018	Hongxia Hao <a href="mailto:haohongxia@huawei.com">haohongxia@huawei.com</a>	Initial Release
18.0.0	June 2018	Hongxia Hao <a href="mailto:haohongxia@huawei.com">haohongxia@huawei.com</a>	Updated Release

### VERSION HISTORY

Version Number	Date	Modified by:	Description
Release 0.1	31/05/2017		First Release of Draft Version of the Document.
Release 0.12	12/07/2017		Update by Hongxia
1.0.0	9/10/2017		Address the comments from Orange, merge the etiya's contribution.
1.0.1	30/10/2017		Address the comments from Amdocs.
1.0.2	10-Jan-2018	Adrienne Walcott	Formatting/style edits prior to publishing
2.0.0	12-Jun-2018	Hongxia Hao	Modify some typos.  Change the table format of fields descriptions.  Change the descriptions of @Attributes related with DG3.0  Updated to TM Forum new brand guidelines
2.0.1	28-Jun-2018	Adrienne Walcott	Formatting/style edits prior to R18 publishing

## CONTRIBUTORS TO DOCUMENT

<ul style="list-style-type: none"><li>• MaXu, Huawei <a href="mailto:maxu@huawei.com">maxu@huawei.com</a></li><li>• Hongxia Hao, Huawei <a href="mailto:haohongxia@huawei.com">haohongxia@huawei.com</a></li></ul>	Initial version
<ul style="list-style-type: none"><li>• Ludovic Robert <a href="mailto:ludovic.robert@orange.com">ludovic.robert@orange.com</a></li><li>• Serafettin ACIR <a href="mailto:serafettin.acir@etiya.com">serafettin.acir@etiya.com</a></li><li>• Hongxia Hao <a href="mailto:haohongxia@huawei.com">haohongxia@huawei.com</a></li></ul>	Add some related information