

# TM Forum Specification

## Communication API REST Specification

**TMF681**  
**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.....	5
INTRODUCTION.....	6
MAPPING WITH SID ABE.....	7
MAPPING WITH ETOM PROCESS.....	8
DISTINCTION BETWEEN THIS API AND EVENT MANAGEMENT API.....	9
DISTINCTION BETWEEN THIS API AND CHANGE REQUEST API.....	12
SAMPLE USE CASES.....	16
CASE1: COMMUNICATION WITH CUSTOMER.....	16
CASE2: COMMUNICATION WITH AGENT (CSR).....	17
Support of polymorphism and extension patterns.....	19
RESOURCE MODEL.....	20
Managed Entity and Task Resource Models.....	20
Communication Message resource model.....	20
Field descriptions.....	20
JSON representation sample.....	24
Notification Resource Models.....	25
Communication Message Creation Notification.....	25
Communication Message Deletion Notification.....	26
Communication Message Update Notification.....	27
API OPERATIONS.....	28
POST /communicationMessage.....	29
POST /communicationMessage/send.....	31
POST /communicationMessage/{ID}/send.....	32
GET /communicationMessage/{ID}?fields=...&{filtering}.....	34
GET /communicationMessage?fields=...&{filtering}.....	35
PATCH /communicationMessage/{ID}.....	36
DELETE /communicationMessage/{ID}.....	37
API NOTIFICATIONS.....	39
Register listener.....	39
Unregister listener.....	40

## Communication API REST Specification

Publish Event to listener .....	40
ACKNOWLEDGEMENTS .....	42
Release History .....	42
Version History .....	42
Contributors to Document .....	43

## LIST OF TABLES

N/A

## INTRODUCTION

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

It provides a standardized mechanism for Communication management such as creation, update, retrieval, deletion and notification of the system communication events.

Communication API manages the following data resources:

- **Communication Message**

- o Communication message means a notification approach in the format of a message which can be dispatched (sent) to the certain user by the system with the content which can be felt and understood by the recipient. The user can be either a final customer or a customer service agent. The message can reach the customer in different interaction channels, including: email, short message, mobile app notification (push).

Normally the communication is implemented as a common shared service for all the IT applications. Whenever there is an application which needs to manage or send the message to the customer, this application can invoke the “communication” API to dispatch the notification.

To help clarify the concept of “communication API”, here all the possible “man-machine” contact approaches are listed as below. The “tick” shows the interaction types for which the “communication API” is designed to support.

Business Interaction Method	Communication API Related
SMS to customer	✓
Email to customer	✓
Mobile app push message to customer	✓
Proactive calling to the customer (human initiated, i.e. person-call-person)	
Proactive calling to the customer (system initiated, i.e. machine-call-person)	
Face to face contact	
Customer browsing web page, open mobile app, calling IVR etc.	

## Communication API REST Specification

Communication API performs the following operation on the resource of “Communication Message”. There are two types of operations provided in this API. One is the management of the request message body. Another is for sending the communication message to the customer.

### *Operations for Communication Message body management*

- Retrieval of an existing Communication Message depending on filter criteria
- Creation of a new Communication Message
- Partial update of an existing Communication Message
- Deletion of an existing Communication Message
- Notification of events:
  - o Creation of Communication Message
  - o Updating Communication Message
  - o Deletion of Communication Message

### *Operations for sending Communication Message.*

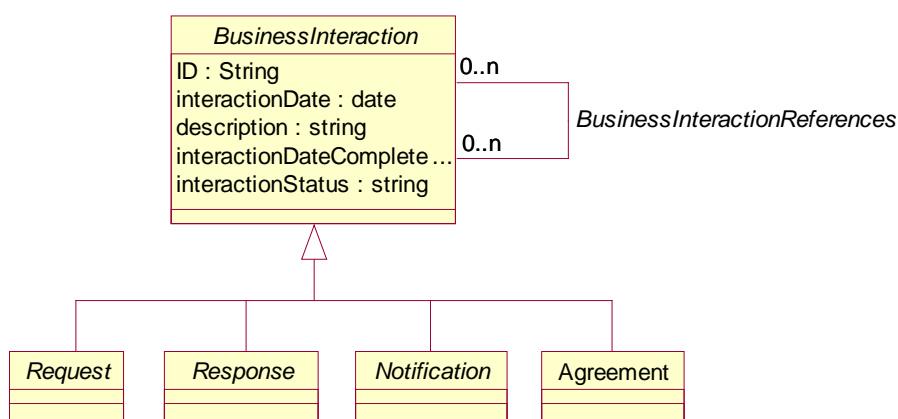
- Send a message, including:
  - o Send a new message with the whole communication message body (POST operation)
  - o Send a message with the predefined communication message body (POST operation)

---

## MAPPING WITH SID ABE

Communication Message is mapped to “*Business Interaction ABE:: Notification ABE*” in TMF Information Framework (SID).

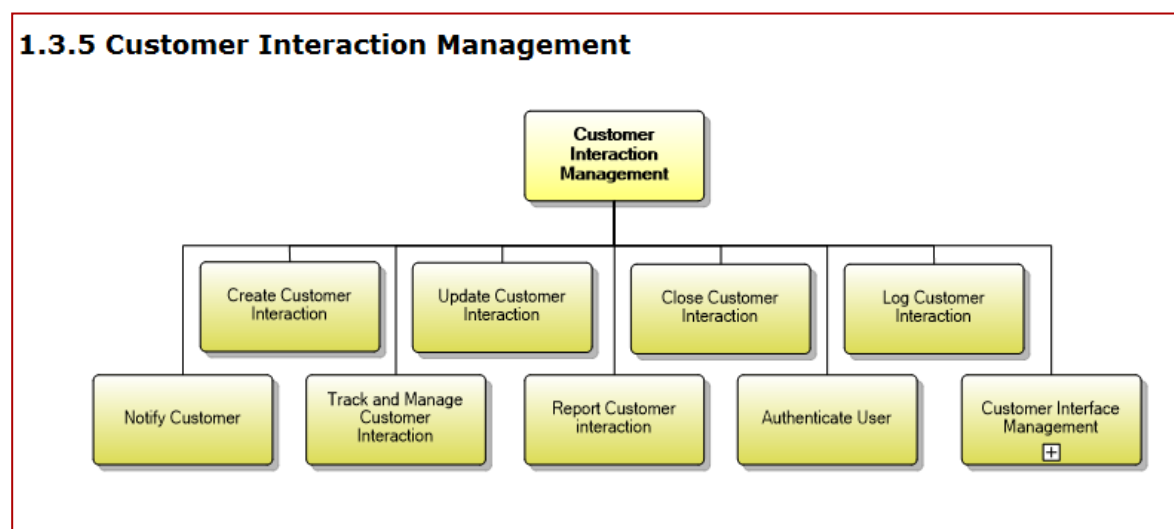
The Business Interaction ABE is illustrated as the diagram below:



## MAPPING WITH ETOM PROCESS

Communication API can be mapped to the “Customer Interaction Management” in TM Forum Process Framework (eTOM).

The relevant process is: **Notify Customer**.



The definition of TM Forum Process Framework is:

**Category:** (3) eTOM Process Type

**Process Identifier:** 1.3.5.5

**Original Process Identifier:** 1.1.1.18.5

**Maturity Level:** 4

### Description

Notify the customer when interesting events happen.

### Extended Description

The purpose of this process is to notify the customer when events related to existing interactions or to significant customer experience happen. Some notifications can be sent immediately using interactive media (such as SMS, Push to applications, etc.) and other notifications can be sent later using asynchronous media such as mail.



---

## DISTINCTION BETWEEN THIS API AND EVENT MANAGEMENT API

This API and Event Management API own some similarities.

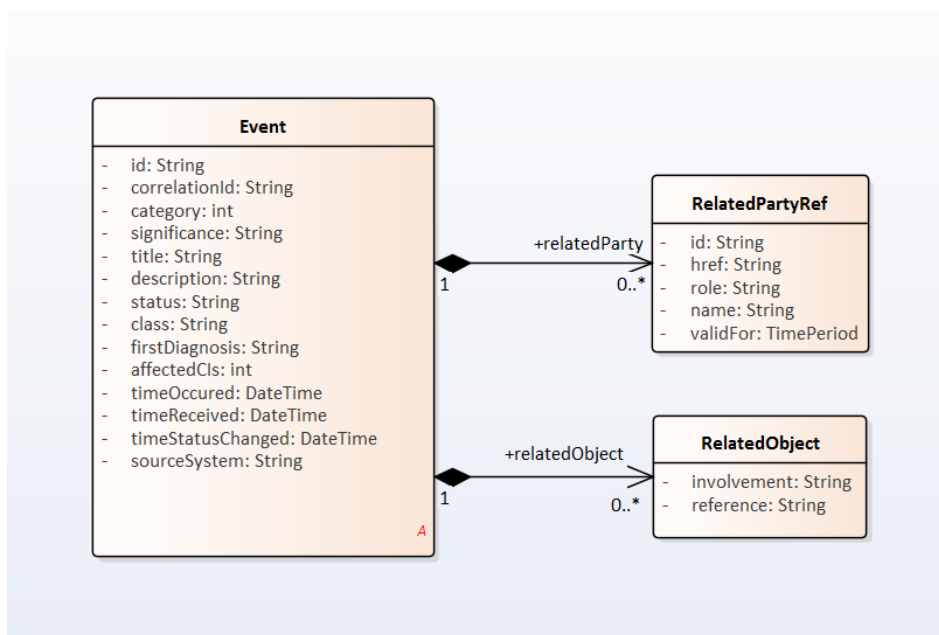
The distinctions between them are explained in the table below:

Comparison Points	Communication API	Event Management API
<b>Definition</b>	Communication message means a message which can be dispatch (sent) to the certain user by the system in the format which can be felt and understood by the recipient	An Event represents a change in the state of a configuration item, service or business data entity
<b>User Role</b>	Sender: Enterprise (e.g. TelCo).  Receiver: Customer, agent, O&M staff	IT System.  No natural person (customer or agent) is involved.
<b>Business Scenario</b>	The enterprise needs to send the notification information to the customer or the agent.  Note: It is used to support the direct interaction with the users.	It can signal status changes or exceptions that allow the appropriate person or system to perform early response actions to ensure service performance and continuity or used as a trigger for automation run books.  Note: It is used to manage the intrinsic event inside the system. The event is always created in the EM backend referring to the internal Event Management system.
<b>Relevant IT System</b>	Interaction or contact module of the system	Any module inside the IT system

Comparison Points	Communication API	Event Management API
<p><b>Data Model</b></p>	<p>Content of communication message. The content is visible to the user (customer, et al)</p>	<p>The description of how an event is triggered and handled.</p> <p>It has no “content” to contain the concrete communication message information. The “related object” in this API does not distinctly express the meaning of content. The “related object” could be the “attachment” of the message.</p> <p>It has no “sender” and “receiver” for the communication message information. The explanation of “related party” in this API has an example which is “assignee support group”. It shows this API parameter is used to depict who is the “event handler”.</p>
<p><b>Information Framework (SID) Mapping</b></p>	<p>Common Business Entities Domain</p> <p><b>Business Interaction ABE</b></p> <p><b>Notification Entity</b></p> <p>A communication that informs about something that has or will happen. A Notification is typically one-sided, in that no Response is expected. A Notification can be created as the result of a Request.</p>	<p>Common Business Entities Domain</p> <p><b>Event ABE</b></p> <p>The Event ABE contains entities that are used to represent events, their occurrence and their recording within systems.</p>
<p><b>Operation</b></p>	<p>1) CRUD of communication message in IT system, i.e., create, read, update and deletion.</p> <p>2) Send the communication message to the user</p>	<p>Create event (i.e. trigger the event) in the system.</p> <p>Query event</p> <p>Update event</p>

Comparison Points	Communication API	Event Management API
<b>Relationship</b>	<p>In the widest conception, any system action can be an event, such as the creation of new offering, execution of order, adding one item into the shopping cart. On this level, the “communication message” is also an event.</p> <p>In the narrow conception, event is for the technical terminology to describe the one-off change of the certain module, such as an event of “memory stack overflow” or “switch the web server”. Such event is collected by the Event Manager to monitor the status of whole running system.</p> <p>According to the definition of Event Management API, this API focuses more on the second concept. “Communication” is also taking place in one-off mode, but it expresses the interactive human-machine contact instead of the pure system action.</p>	

Reference: Event Management API Data Model



## DISTINCTION BETWEEN THIS API AND CHANGE REQUEST API

This API and Change Request API own some similarities.

The distinctions between them are explained in the table below:

Comparison Points	Communication API	Event Management API
<b>Definition</b>	Communication message means a message which can be dispatch (sent) to the certain user by the system in the format which can be felt and understood by the recipient	Change Management process is to respond to the customer’s changing business requirements.  The Change Management API provides the standard integration capabilities between external applications and Change Management Application

## Communication API REST Specification

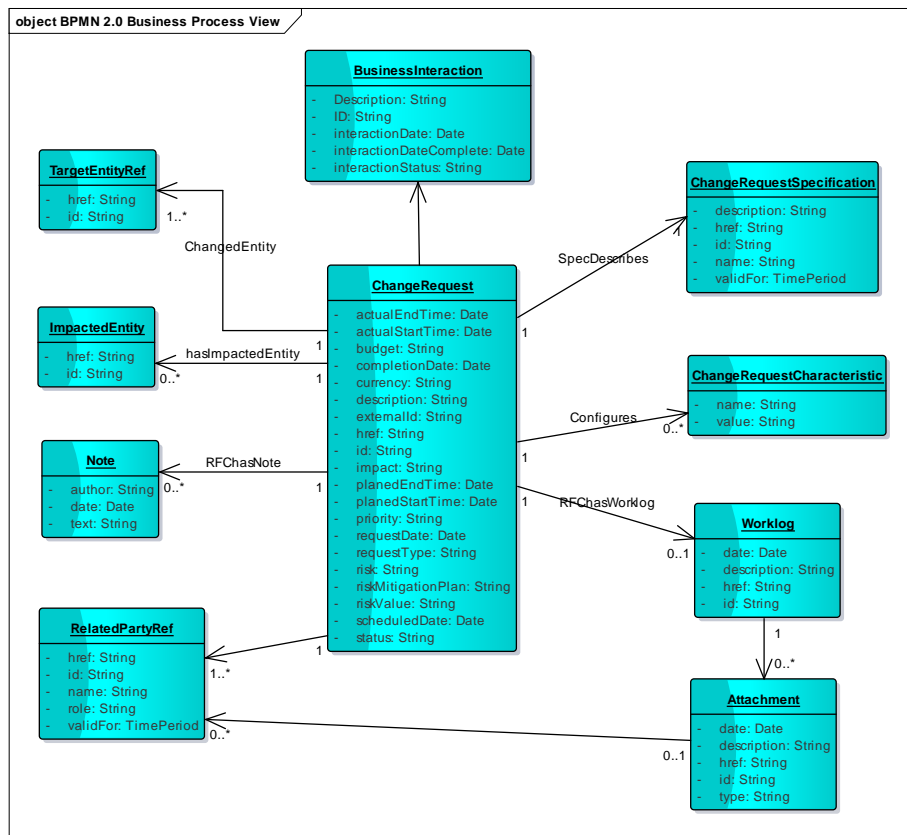
Comparison Points	Communication API	Event Management API
<b>User Role</b>	<p>Sender: Enterprise (e.g. TelCo).</p> <p>Receiver: Customer, agent, O&amp;M staff</p>	<p>The involved roles of the API can be:</p> <ul style="list-style-type: none"> <li>- Change Management Application</li> <li>- External application</li> </ul> <p>No natural person (customer or agent) is involved.</p>
<b>Business Scenario</b>	<p>The enterprise needs to send the notification information to the customer or the agent.</p> <p>Note: It is used to support the direct interaction with the users.</p>	<ul style="list-style-type: none"> <li>- Asset sharing</li> <li>- NFV MULTI-DIGITAL SERVICE PROVIDER offer the products</li> <li>- Manage Service of Network Operation or OSS</li> </ul>

Comparison Points	Communication API	Event Management API
<p><b>Data Model</b></p>	<p>Inherit from “Business Interaction”.</p> <p>Content of communication message. The content is visible to the user (customer, et al).</p>	<p>Inherit from “Business Interaction”.</p> <p>nota bene: The “Business Interaction” has a group of derivative objects. “Change Request” is mapped to “request” object, not “notification”.</p> <p>Additionally, this API data model contains:</p> <ul style="list-style-type: none"> <li>- Attachment: <i>This is also included in Communication API</i></li> <li>- Related Party: The role which is involved. <i>This is also included in Communication API</i></li> <li>- Target Entity, Impacted Entity: both are referred to “related party”.</li> <li>- Work log <ul style="list-style-type: none"> <li>✓ If the work log is a type of task for the staff to execute, it is not required in <i>Communication API</i></li> <li>✓ If the work log is the pure log (record) of the system action, it is a default function of the system. <i>Communication API</i> has the “log flag” to indicate whether the log should be generated after invoking API. The detail of the log is not required to be explicitly expressed in <i>Communication API</i></li> </ul> </li> <li>- Note: It is an optional attribute. In <i>Communication API</i>, the “content” is used to describe the purpose of the communication message.</li> </ul>

Communication API REST Specification

Comparison Points	Communication API	Event Management API
<b>Information Framework (SID) Mapping</b>	<p>Common Business Entities Domain</p> <p><b>Business Interaction ABE</b></p> <p><b>Notification Entity</b></p>	<p>Common Business Entities Domain</p> <p><b>Business Interaction ABE</b></p>
<b>Operation</b>	<p>1) CRUD of communication message in IT system, i.e., create, read, update and deletion.</p> <p>2) Send the communication message to the user</p>	<p>CRUD operations of the change request</p>

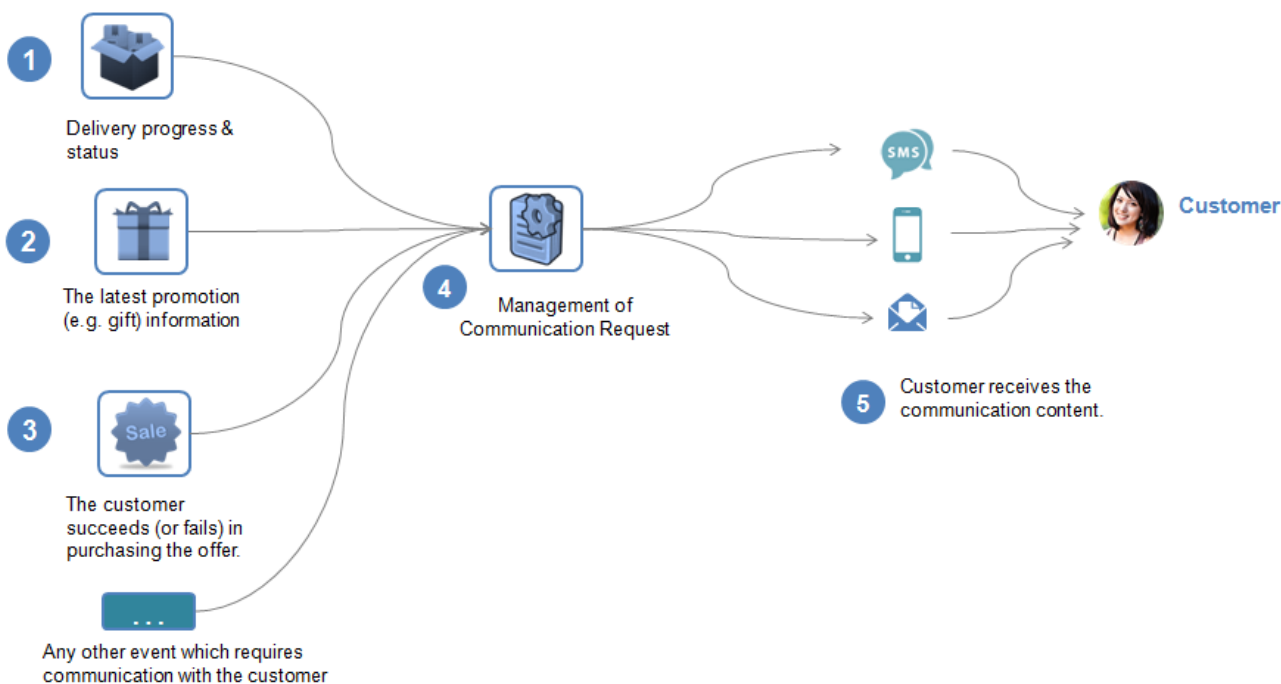
Reference: Change Request API Data Model



## SAMPLE USE CASES

Examples of use cases using Communication API is as following

### CASE1: COMMUNICATION WITH CUSTOMER



Use Case Id	UC_TMF_CommunicationMessage_0001
Use Case Name	Customer receives communication message.
Summary	This case describes the system manages the communication message and sends the communication message to the customer.
Actor(s)	Customer (person)
Pre-Conditions	NA
Begins When	When the sales/marketing/service or any other business activities requires notifying the customer, the application will initiate the communication message.



Use Case Id	UC_TMF_CommunicationMessage_0001
Description	<ol style="list-style-type: none"> <li>1) The system needs to notify the customer about the delivery progress &amp; status.</li> <li>2) The system needs to notify the customer about the latest promotion (e.g. gift) information.</li> <li>3) The system needs to notify the customer about the result of purchasing the offer.</li> </ol> <p>Or any other event happens which requires communication with the customer.</p> <ol style="list-style-type: none"> <li>4) The system manages the Communication Message, such as the creation, modification, updating and deletion of the communication.</li> <li>5) The system sends the communication and the customer receives the communication content.</li> </ol>
Ends When	<p><i>In case of communication is done successfully:</i></p> <p>The customer will receive the message.</p> <p><i>In case of failure:</i></p> <p>The system records the failure and the customer cannot know the message. Normally the system will retry to send the message when the network or environment is available.</p>
Post-Conditions	
Exceptions	
Traceability	

---

## CASE2: COMMUNICATION WITH AGENT (CSR)

Use Case Id	UC_TMF_CommunicationMessage_0002
Use Case Name	The agent (CSR staff) receives communication message.
Summary	This case describes the system manages the communication message and sends the communication message to the agent.

## Communication API REST Specification

Actor(s)	Agent staff (person)
Pre-Conditions	NA
Begins When	When the sales/marketing/service or any other business activities requires notifying the agent, the application will initiate the communication message.
Description	<p>The company needs to notify the agent that the latest SLA policy in the market has been published.</p> <p>The agent needs to know such company rule (policy) and obey it when working.</p>
Ends When	<p><i>In case of communication is done successfully:</i></p> <p>The agent will receive the message.</p> <p><i>In case of failure:</i></p> <p>The system records the failure and the customer cannot know the message. Normally the system will retry to send the message when the network or environment is available.</p>
Post-Conditions	
Exceptions	
Traceability	

## 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 `CommunicationMessage`, `Sender`, `Receiver` and `Attachment Entity`. 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`) 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**

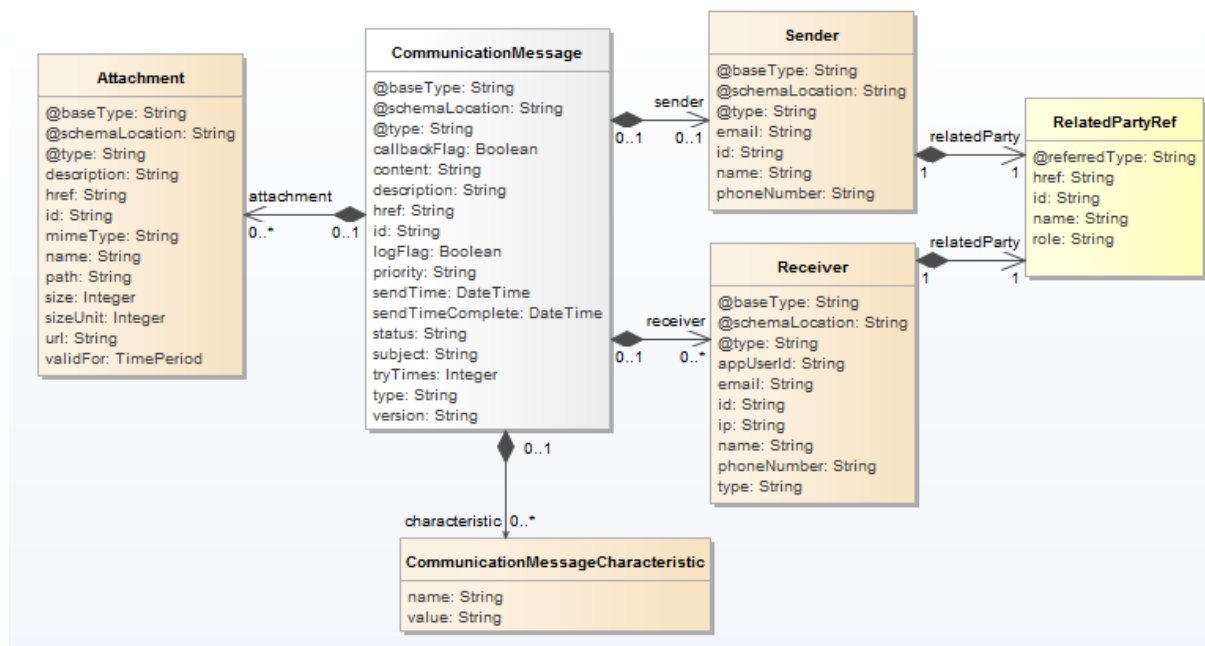


Fig.1. Communication Message Resource

**FIELD DESCRIPTIONS**

**Communication Message**

Fields	Data Type	Description
@type	String	Indicates the type of resource.
@baseType	String	Indicates the type of resource.
@schemaLocation	String	Link to schema describing this resource.
id	String	Unique identifier of Communication Message (inherit from Business Interaction ABE)
href	String	Hypertext Reference of the Communication Message.
Priority	Integer	Priority level for applying this alteration among all the defined alterations.
Type	String	It could be SMS, Email, "MobileApp" (push notification)
Subject	String	The title of the message. It is necessary for the email and mobile app push.

## Communication API REST Specification

Fields	Data Type	Description
Content	String	The content of the communication message.
SendTime	DateTime	The time of sending communication message. (inherit from Business Interaction ABE)
sendTimeComplete	DateTime	The time of completion of sending communication message. (inherit from Business Interaction ABE)
status	String	Status of communication message (inherit from Business Interaction ABE)
description	String	Description for the whole object (inherit from Business Interaction ABE)
logFlag	Boolean	It is used to decide whether the contact log needs to be recorded.
callbackFlag	Boolean	It is used to decide whether the contact message needs to be replied.
tryTimes	Integer	If fail to send the communication message, how many times the system will retry.
version	String	The version of the message template.
characteristic	CommunicationRequestCharacteristic	The values of parameters which are used in the content if the content contains them.
sender	Sender	Sender of the communication message.
receiver	Receiver	Receivers of the communication message.
relatedParty	RelatedPartyRef	The party entity of the receiver (user)
attachment	Attachment	The attachments of the communication message (when it is email type).

**CommunicationRequestCharacteristic**

The values of parameters which are used in the content if the content contains them.

Fields	Data Type	Descriptions
name	String	Content parameter identifier.
value	String	Content parameter value.

**Sender**

Sender of the communication message.

## Communication API REST Specification

Fields	Data Type	Descriptions
@type	String	Indicates the type of resource
@basetype	String	Indicates the type of resource
@schemaLocation	String	It provides the link to the schema describing REST resource
email	String	Sender address of email, if the communication type is email
id	String	ID of the sender
name	String	Name of the sender
phoneNumber	String	Phone number of the sender, if the communication type is SMS.

**Receiver**

Receivers of the communication message.

Fields	Data Type	Descriptions
@type	String	Indicates the type of resource
@basetype	String	Indicates the type of resource
@schemaLocation	String	It provides the link to the schema describing REST resource
appUserId	String	ID of the mobile app user
email	String	Receiver address of email, if the communication type is email
id	String	ID of the receiver
ip	String	IP address of the receiver
name	String	Name of the receiver
phoneNumber	String	Phone number of the receiver, if the communication type is SMS.

Fields	Data Type	Descriptions
type	String	The type of receiver, like the email receiver could be in "to", "cc" or "bcc" mode.

### **Attachment**

The attachments of the communication message (when it is email type).

Fields	Type	Description
@type	String	Indicates the type of resource.
@baseType	String	Indicates the type of resource.
@schemaLocation	String	Link to schema describing this resource.
name	String	The name of attached file in the communication message.
path	String	The path of the attached file in the communication message.
description	String	Description of the attached file
href	String	href of the attached file
contentType	String	Multi-purpose Internet Mail Extensions Type
Size	Integer	Size of the attached file
sizeUnit	Integer	Size Unit of the attached file
url	String	URL of the attached file
validFor	TimePeriod	"Valid For" period of the attached file

### **RelatedPartyRef**

*The party entity of the receiver (user) or the sender*

Fields	Type	Description
@referredType	String	Indicates the type of resource.
href	String	Hyperlink to access a related party.
id	String	Identifier of a related party.
name	String	Name of related party.
role	String	Role of related Party.

---

## JSON REPRESENTATION SAMPLE

We provide below the JSON representation of an example of Change Request Resource object:

```
{
  "id": "1001",
  "href": "http://serverlocation:port/communicationMessage/v2/communicationMessage/564",
  "type": "1",
    "priority": "1",
  "subject": "News: the latest promotion for you",
  "sendTime": "2016-12-19 T04:00:00.0Z",
  "sendTimeComplete": "2016-12-19 T05:00:00.0Z",
  "status": "Completed",
  "type": "***",
  "description": "this is communication message for promotion",
  "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
  "@type": "communication",
  "@schemaLocation": "http://serverlocation:port/communication/schema/communication.yml",
  "@baseType": "",
  "callbackFlag": "true",
  "tryTimes": "3",
  "version": "0.1",
  "characteristic": [
    {
      "name": "$Parameter1",
      "value": "Mr. Bush"
    },
    {
      "Name": "$Parameter2",
      "value": "4G_LTE Discount 30%"
    }
  ],
  "attachment": [
    {
      "path": "/attachedfile/1, /attachedfile/",
      "name": "File_XYZ_001"
    }
  ],
  "sender": {
    "id": "10099",
    "name": "ABC Company",
    "phoneNumber": "10086"
  },
  "receiver": [
    {
      "id": "10234",
      "name": "Customer",

```



```

        "phoneNumber" : "008613811112222"
        "relatedParty": {
            "id": "991",
            "href": "http://serverlocation:port/partyManagement/individual/1",
            "role": "customer",
            "name": "John Lock"
        },
    ]
}

```

## Notification Resource Models

3 notifications are defined for this API

Notifications related to CommunicationMessage:

- CommunicationMessageCreationNotification
- CommunicationMessageUpdateNotification
- CommunicationMessageDeletionNotification

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).

---

## COMMUNICATION MESSAGE CREATION NOTIFICATION

It is used to notify that a Communication Message is created.

```

{
  "eventType": " CommunicationMessageCreationNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "8976",
  "event":
  {
    "CommunicationMessage": [

```

```
{
  "id": "s1234",
  Following a whole representation of the Communication Message resource with all its
  attributes.

  Refer to communication message Resource.
}
```

---

## COMMUNICATION MESSAGE DELETION NOTIFICATION

It is used to notify that a Communication Message is deleted.

```
{
  "eventType": "CommunicationMessageDeletionNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "8976",
  "CommunicationMessage": {
    "id": "s1234",
    Following a whole representation of the Communication Message resource with all its
    attributes.

    Refer to Communication Message Resource.
  }
}
```

---

## COMMUNICATION MESSAGE UPDATE NOTIFICATION

```
{
  "eventType": "CommunicationMessageUpdateNotification",
  "eventTime": "2014-09-27T05:46:25.0Z",
  "eventId": "8976",
  "CommunicationMessage":{
    "id": "s1234",
    Following a whole representation of the Communication Message resource with all its
    attributes.
    Refer to Communication Message Resource.
  }
}
```

## 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.

## POST /communicationMessage

### Description

- This API is used to create a new Communication Message. The Communication Message is used to express the message itself. After the Communication Message has been created, it can be sent by the system to the “receiver” later.
- Condition:  
There is no parameter in this POST verb URI
- Return Status Codes:

Status Code	Description
201	The resource has been added successfully
400	Request Error
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Other	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

### Usage Samples

REQUEST
POST /communicationMessage Content-type: application/json  <pre>{   "id": "1001",   "href": "http://serverlocation:port/communicationMessage/v1/communicationMessage/56445633245",   "type": "1",   "priority": "1",</pre>

```
    "subject" : "News: the latest promotion for you",
    "sendTime" : "2016-12-19 T04:00:00.0Z",
    "sendTimeComplete" : "2016-12-19 T05:00:00.0Z",
    "status" : 'Completed',
    "description" : "this is communication message for promotion",
    "content" : "Dear $Parameter1, Here is the information of the promotion
$Parameter2",
    "contactLogFlag" : "false",
    "callbackFlag" : "true",
    "tryTimes" : "3",
    "version" : "0.1",
    "characteristic" : [
        {
            "name" : "$Parameter1",
            "value" : "Mr. Bush"
        },
        {
            "name" : "$Parameter2",
            "value" : "4G_LTE Discount 30%"
        }
    ],
    "attachment" : [{
        "path" : "/attachedfile/1, /attachedfile/",
        "name" : "File_XYZ_001"
    }],

    "sender" : {
        "id" : "10099",
        "name" : "ABC Company",
        "phoneNumber" : "10086"
    },
    "receiver" : [ {
        "id" : "10234",
```

<pre>        "name" : "Customer",         "phoneNumber" : "008613811112222"     "relatedParty": {         "id": "991",         "href": "http://serverlocation:port/partyManagement/individual/1",         "role": "customer",         "name": "John Lock"     },     ] }</pre>
<b>RESPONSE</b>
<p>201 Content-Type: application/json</p> <p><i>Following a whole representation of the Communication Message resource with all its attributes.</i></p> <p><i>Refer to Communication Message Resource.</i></p>

## POST /communicationMessage/send

### Description:

- This API is used to send a new Communication message from the “sender” to the “receiver”.  
When executing this API, the message will be directly sent to the receiver, i.e., the final customer.  
The full message body and attributes should be filled in this operation request.
- Condition:  
The parameter of POST is “send” to indicate this operation is for the “sending message” action.

- Return Status Codes:

Status Code	Description
200	The message of communication message has been sent.
400	Request Error
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Other	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

Sample:

REQUEST
POST /communicationMessage/send Content-type: application/json  <i>The following is a whole representation of the Communication Message resource with all its attributes.</i>  <i>Please refer to the "Communication Message Creation" for the content of example.</i>
RESPONSE
200

POST /communicationMessage/{ID}/send

Description:

- This API is used to send a pre-defined Communication message from the "sender" to the "receiver".  
When executing this API, the message will be sent to the receiver, i.e., the final customer.



## Communication API REST Specification

In this mode, the message body should be created in advance. The “Communication Message Creation” needs to be invoked firstly, so the system can send the pre-defined message.

- **Condition:**  
The parameter of POST is “send” to indicate this operation is for the “sending message” action.
- **Return Status Codes:**

Status Code	Description
200	The message of communication message has been sent.
400	Request Error
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Other	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

Sample:

<b>REQUEST</b>
POST /communicationMessage/1001/send Content-type: application/json
<b>RESPONSE</b>
200

GET /communicationMessage/{ID}?fields=...&{filtering}

Description:

- This API is used to query an existing pre-defined message body by query conditions

Behavior:

- Return Status Codes:

Status Code	Description
200	The resource has been retrieved
404	If no record was found
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Others	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

Sample:

<b>REQUEST</b>
GET /communicationMessage/11006 Content-type: application/json Accept: application/json
<b>RESPONSE</b>
200 Content-Type: application/json  <i>Following a whole representation of the Communication Message resource with all its attributes.</i>

## Communication API REST Specification

*Refer to Communication Message Resource.*

### GET /communicationMessage?fields=...&{filtering}

#### Description:

- This API is used to query existing pre-defined messages by query conditions

#### Behavior:

- Return Status Codes:

Status Code	Description
200	The resource has been retrieved
404	If no record was found
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Others	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

#### Sample:

<b>REQUEST</b>
GET /communicationMessage Content-type: application/json Accept: application/json
<b>RESPONSE</b>
200

```

Content-Type: application/json
[
{
    Following a whole representation of the Communication Message resource with all its
    attributes.

    Refer to Communication Message Resource.
}
]

```

## PATCH /communicationMessage/{ID}

### Description:

- This API is used to partially update an existing pre-defined message body.

### Behavior:

- Return Status Codes :

Status Code	Description
201	Update the communication message successfully
400	Request Error
404	If no record was found
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Others	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

## Communication API REST Specification

Example:

REQUEST
PATCH / communicationMessage/1006 Content-type: application/json <pre>{   "lifecycleStatus": "Release",   "name": "Gift_On_Birthday" }</pre>
RESPONSE
200 Content-Type: application/json  <i>Following a whole representation of the communication message resource with all its attributes.</i>  <i>Refer to communication message Resource.</i>

## DELETE /communicationMessage/{ID}

Description:

- This API is used to delete an existing pre-defined message body.

Behavior:

- Return Status Codes :

Status Code	Description
204	Delete the communication message successfully

## Communication API REST Specification

Status Code	Description
400	Request Error
404	If no record was found
500	The server encountered an unexpected condition which prevented it from fulfilling the request
Others	The server may use other HTTP error status codes to reflect the error, the client must be processed in accordance with the error messages in other HTTP specifications.

<b>REQUEST</b>
DELETE / communicationMessage/1006 Content-type: application/json
<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.

<b>Request</b>
POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}
<b>Response</b>
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.

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	Maxu <a href="mailto:maxu@huawei.com">maxu@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
1.0.0	13-Nov-2016	Maxu	Initial Document.
1.1.0	09-Oct-2017	Maxu Hongxia Hao	Addressed some comments from Orange & Vodafone & TMF.
1.1.1	23-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. Align with DG3.0 Updated to TM Forum new brand guidelines
2.0.1	29-Jun-2018	Adrienne Walcott	Formatting/style edits prior to R18 publishing

## CONTRIBUTORS TO DOCUMENT

Hongxia Hao	Huawei
Maxu	Huawei
Mariano Belaunde	Orange
Nicoleta Stoica	Vodafone
Kuang Chunguang	Huawei
Jiang Yisong	Huawei
Pierre Gauthier	TM Forum