TM Forum Specification

Communication API User Guide

TMF681
Team Approved Date: 28-May-2020

Release Status: Production	**Approval Status: TM Forum Approved**
Version 4.0.1	**IPR Mode: RAND**

NOTICE

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

Table of Contents

NOTICE 2
Table of Contents 4
List of Tables 5
Introduction 6
MAPPING WITH SID ABE 7
MAPPING WITH ETOM PROCESS 8
DISTINCTION BETWEEN THIS API AND EVENT MANAGEMENT API 8
DISTINCTION BETWEEN THIS API AND CHANGE REQUEST API 11
SAMPLE USE CASES 14
CASE1: COMMUNICATION WITH CUSTOMER 14
CASE2: COMMUNICATION WITH AGENT (CSR) 15
Support of polymorphism and extension patterns 17
RESOURCE MODEL 18
Managed Entity and Task Resource Models 18
Communication Message resource 18
Notification Resource Models 23
Communication Message Attribute Value Change Event 24
Communication Message State Change Event 25
API OPERATIONS 26
Operations on Communication Message 26
List communication messages 26
Retrieve communication message 29
Create communication message 30
Patch communication message 35
Delete communication message 37
API NOTIFICATIONS 38
Register listener 38
Unregister listener 39
Publish Event to listener 39
Acknowledgements 41
Version History 41
Release History 41
Contributors to Document 42

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**
    - 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 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:
    - Creation of Communication Message
    - Updating Communication Message
    - Deletion of Communication Message

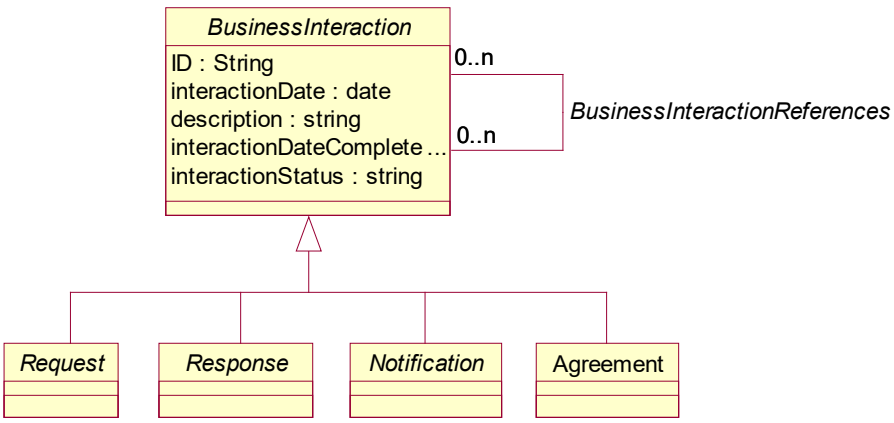
Operations for sending Communication Message.

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

MAPPING WITH SID ABE

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

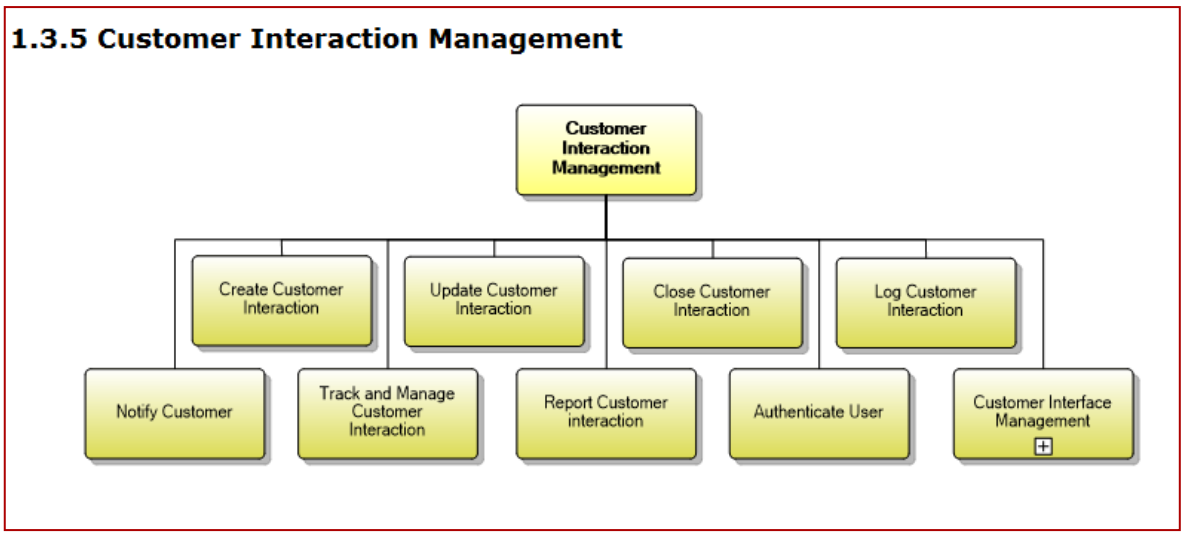
The Business Interaction ABE is illustrated as the diagram below:



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
Definition	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
User Role	Sender: Enterprise (e.g. TelCo). Receiver: Customer, agent, O&M staff	IT System. No natural person (customer or agent) is involved.
Business Scenario	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.
Relevant IT System	Interaction or contact module of the system	Any module inside the IT system

Comparison Points	Communication API	Event Management API
Data Model	Content of communication message. The content is visible to the user (customer, et al)	The description of how an event is triggered and handled. 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. 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”.
Information Framework (SID) Mapping	Common Business Entities Domain **Business Interaction ABE** **Notification Entity** 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.	Common Business Entities Domain **Event ABE** The Event ABE contains entities that are used to represent events, their occurrence and their recording within systems.
Operation	1) CRUD of communication message in IT system, i.e., create, read, update and deletion. 2) Send the communication message to the user	Create event (i.e. trigger the event) in the system. Query event Update event

Comparison Points	Communication API	Event Management API
Relationship	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. 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. 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.	

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	Change Management API
Definition	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
User Role	Sender: Enterprise (e.g. TelCo). Receiver: Customer, agent, O&M staff	The involved roles of the API can be: - Change Management Application - External application No natural person (customer or agent) is involved.

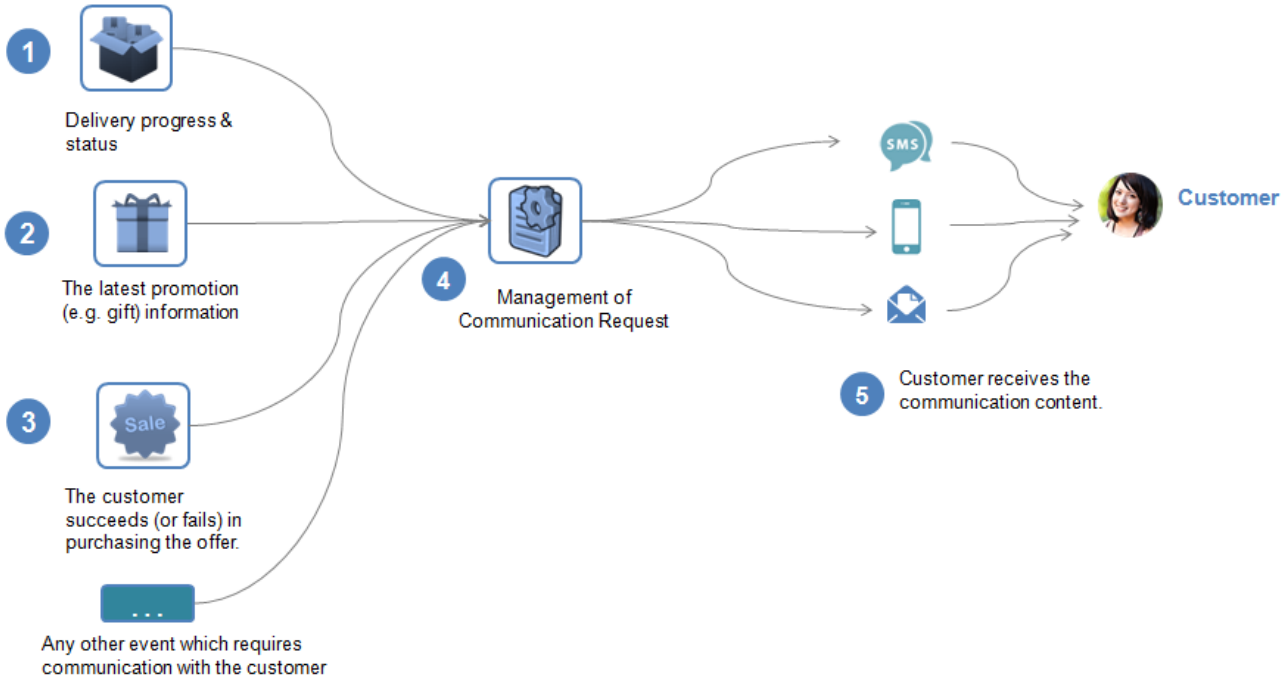
Comparison Points	Communication API	Change Management API
Business Scenario	The enterprise needs to send the notification information to the customer or the agent. Note: It is used to support the direct interaction with the users.	- Asset sharing - NFV MULTI-DIGITAL SERVICE PROVIDER offer the products - Manage Service of Network Operation or OSS
Data Model	Inherit from "Business Interaction". Content of communication message. The content is visible to the user (customer, et al).	Inherit from "Business Interaction". nota bene: The "Business Interaction" has a group of derivative objects. "Change Request" is mapped to "request" object, not "notification". Additionally, this API data model contains: - Attachment: *This is also included in Communication API* - Related Party: The role which is involved. *This is also included in Communication API* - Target Entity, Impacted Entity: both are referred to "realted party". - Work log ✓ If the work log is a type of task for the staff to execute, it is not required in *Communication API* ✓ If the work log is the pure log (record) of the system action, it is a default function of the system. *Communication API* 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 *Communication API* - Note: It is an optional attribute. In *Communication API*, the "content" is used to describe the purpose of the communication message.
Information Framework (SID) Mapping	Common Business Entities Domain **Business Interaction ABE** **Notification Entity**	Common Business Entities Domain **Business Interaction ABE**

Comparison Points	Communication API	Change Management API
Operation	1) CRUD of communication message in IT system, i.e., create, read, update and deletion. 2) Send the communication message to the user	CRUD operations of the change request

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	1) The system needs to notify the customer about the delivery progress & status. 2) The system needs to notify the customer about the latest promotion (e.g. gift) information. 3) The system needs to notify the customer about the result of purchasing the offer. Or any other event happens which requires communication with the customer. 4) The system manages the Communication Message, such as the creation, modification, updating and deletion of the communication. 5) The system sends the communication and the customer receives the communication content.
Ends When	*In case of communication is done successfully:* The customer will receive the message. *In case of failure:* 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.
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.
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	The company needs to notify the agent that the latest SLA policy in the market has been published. The agent needs to know such company rule (policy) and obey it when working.
Ends When	*In case of communication is done successfully:* The agent will receive the message. *In case of failure:* 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.
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. Polymorphism in collections occurs when entities inherit from base entities, for instance an EnhancedCommunicationMessage inheriting properties from the CommunicationMessage entity.

Generic support of polymorphism and pattern extensions is described in the TMF API Design Guidelines.

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.

The @referredType can be used within reference entities 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 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.

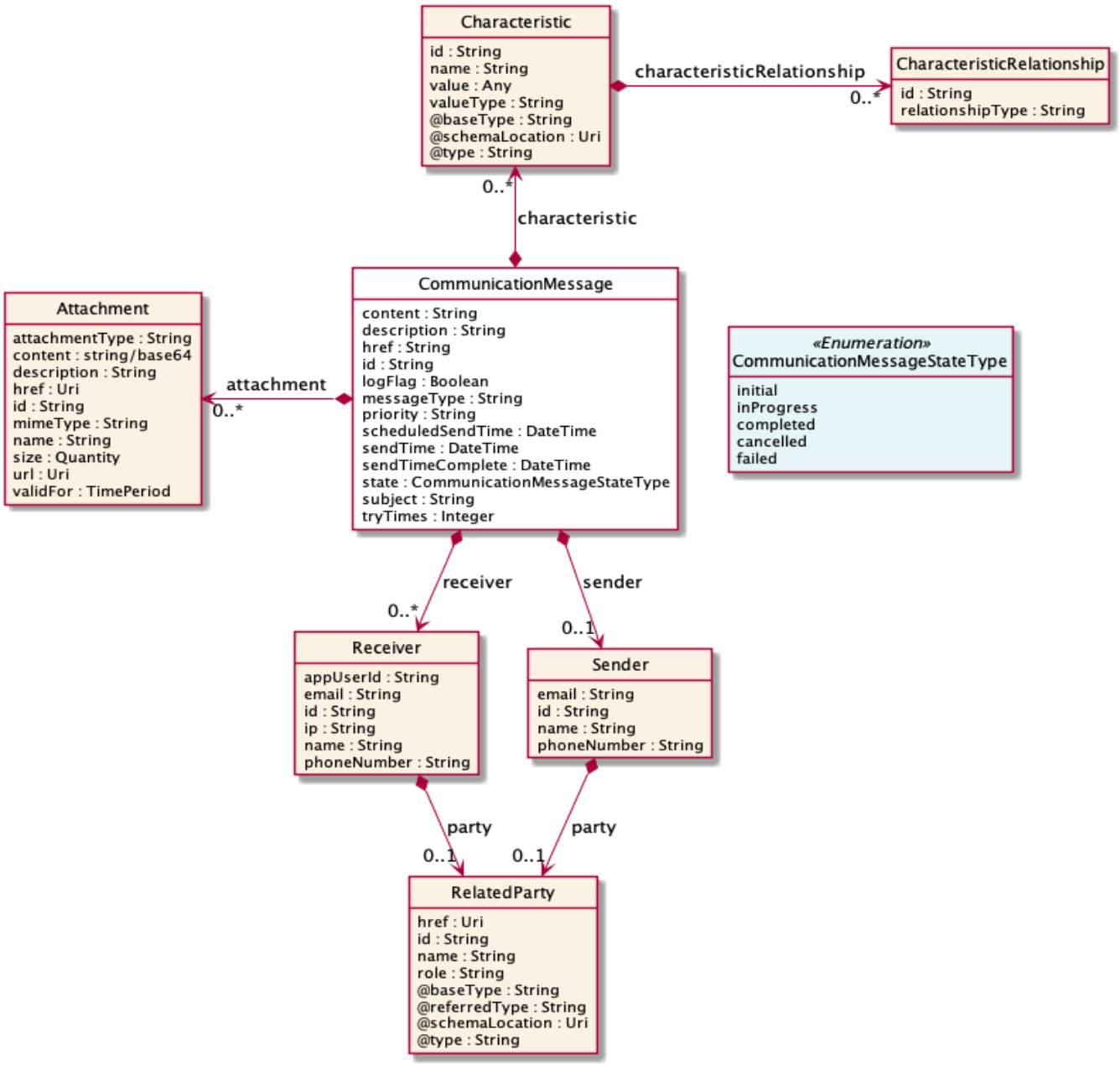
RESOURCE MODEL

Managed Entity and Task Resource Models

Communication Message resource

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).

Resource model



Field descriptions

CommunicationMessage fields

attachment	A list of attachments (Attachment [*]). Any attachment associated with this message.
characteristic	A list of characteristics (Characteristic [*]). Any additional characteristic(s) of this message.
content	A string. The content of the communication message.
description	A string. Description for the whole object.
href	A string. Hypertext Reference of the Communication Message.
id	A string. Unique identifier of Communication Message.
logFlag	A boolean.
messageType	A string. The type of message, such as: SMS, Email, Mobile app push notification.
priority	A string. The priority of the communication message. Small number means higher priority.
receiver	A list of receivers (Receiver [*]). The receiver(s) of this message.
scheduledSendTime	A date time (DateTime). The scheduled time for sending the communication message.
sendTime	A date time (DateTime). The time of sending communication message.
sendTimeComplete	A date time (DateTime). The time of completion of sending communication message.
sender	A sender (Sender). The sender of this message.
state	A communication message state type (CommunicationMessageStateType). Status of communication message.
subject	A string. The title of the message, necessary for the email and mobile app push.
tryTimes	An integer. How many times do you want to retry the delivery of this message?.

Attachment sub-resource

Complements the description of an element (for instance a product) through video, pictures...

attachmentType	A string. Attachment type such as video, picture.
content	A base 6 4 (Base64). The actual contents of the attachment object, if embedded, encoded as base64.
description	A string. A narrative text describing the content of the attachment.

href	An uri (Uri). URI for this Attachment.
id	A string. Unique identifier for this particular attachment.
mimeType	A string. Attachment mime type such as extension file for video, picture and document.
name	A string. The name of the attachment.
size	A quantity (Quantity). The size of the attachment.
url	An uri (Uri). Uniform Resource Locator, is a web page address (a subset of URI).
validFor	A time period. The period of time for which the attachment is valid.

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
characteristicRelationship	A list of characteristic relationships (CharacteristicRelationship [*]). Another Characteristic that is related to the current Characteristic;.
id	A string. Unique identifier of the characteristic.
name	A string. Name of the characteristic.
value	An any (Any). The value of the characteristic.
valueType	A string. Data type of the value of the characteristic.

CharacteristicRelationship sub-resource

Another Characteristic that is related to the current Characteristic;.

id	A string. Unique identifier of the characteristic.
relationshipType	A string. The type of relationship.

Quantity sub-resource

An amount in a given unit.

amount	A float. Numeric value in a given unit.
units	A string. Unit.

Receiver sub-resource

Receivers of the communication message.

appUserId	A string. ID of the mobile app user.
email	A string. Receiver address of email, if the communication type is email.
id	A string. ID of the receiver.
ip	A string. IP address of the receiver.
name	A string. Name of the receiver.
party	A related party (RelatedParty). Related Entity reference. A related party defines party or party role linked to a specific entity.
phoneNumber	A string. Phone number of the receiver, if the communication type is SMS.

RelatedParty sub-resource

Related Entity 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.
name	A string. Name of the related entity.
href	An uri (Uri). Hyperlink reference.
id	A string. unique identifier.
@baseType	A string. When sub-classing, this defines the super-class.
@schemaLocation	An uri (Uri). A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class Extensible name.
role	A string. Role played by the related party.

Sender sub-resource

Sender of the communication message.

email	A string. Sender address of email, if the communication type is email.
id	A string. ID of the sender.
name	A string. Name of the sender.
party	A related party (RelatedParty). Related Entity reference. A related party defines party or party role linked to a specific entity.

phoneNumber A string. Phone number of the sender, if the communication type is SMS.

Json representation sample

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

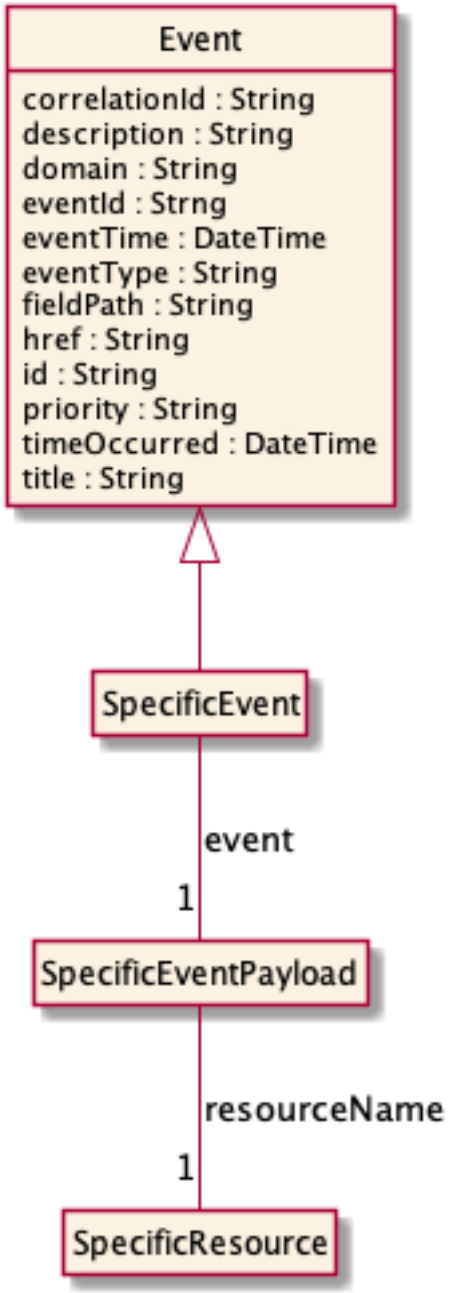

```
{
  "id": "1708",
  "href": "https://serverlocation/tmf-api/communicationManagement/v4/communicationMessage/1708",
  "subject": "News: the latest promotion for you",
  "scheduledSendTime": "2020-02-10T00:00:00+01:00",
  "state": "initial",
  "description": "this is communication message for promotion",
  "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
  "messageType": "SMS",
  "characteristic": [
    {
      "name": "$Parameter1",
      "value": "Mr. Jones",
      "valueType": "string"
    },
    {
      "name": "$Parameter2",
      "value": "4G_LTE Discount 30%",
      "valueType": "string"
    }
  ],
  "attachment": [
    {
      "url": "https://serverlocation/files/1",
      "name": "File_XYZ_001"
    }
  ],
  "sender": {
    "id": "10099",
    "name": "ABC Company",
    "phoneNumber": "10086"
  },
  "receiver": [
    {
      "id": "10234",
      "name": "Customer",
      "phoneNumber": "008613811112222",
      "party": {
        "id": "991",
        "href": "https://serverlocation/tmf-api/partyManagement/v4/individual/1",
        "role": "customer",
        "name": "John White",
        "@referredType": "Individual"
      }
    }
  ]
}
```

Notification Resource Models

2 notifications are defined for this API

Notifications related to CommunicationMessage:
- CommunicationMessageAttributeValueChangeEvent
- CommunicationMessageStateChangeEvent

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).



Communication Message Attribute Value Change Event

Notification CommunicationMessageAttributeValueChangeEvent case for resource CommunicationMessage

Json representation sample

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


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

```
   "event": {
     "communicationMessage" :
        {-- SEE CommunicationMessage RESOURCE SAMPLE --}
   }
}
```

Communication Message State Change Event

Notification CommunicationMessageStateChangeEvent case for resource CommunicationMessage

Json representation sample

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

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"CommunicationMessageStateChangeEvent",
    "event": {
      "communicationMessage" :
         {-- SEE CommunicationMessage 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
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 Communication Message

List communication messages

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

Description

This operation list communication message 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 multiple communication messages.

Request

```
GET /tmf-api/communicationManagement/v4/communicationMessage?status=inprogress
Accept: application/json
```

Response

```
200

[
  {
    "id": "2708",
    "href": "https://serverlocation/tmf-api/communicationManagement/v4/communicationMessage/2708",
    "subject": "News: the latest promotion for you",
    "scheduledSendTime": "2020-02-10T00:00:00+01:00",
    "sendTime": "2020-02-10T05:00:00+01:00",
    "sendTimeComplete": "2020-02-10T05:05:00+01:00",
    "state": "completed",
    "description": "this is communication message for promotion",
    "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
    "messageType": "SMS",
    "characteristic": [
      {
        "name": "$Parameter1",
        "value": "Mrs. Jones",
        "valueType": "string"
      },
      {
        "name": "$Parameter2",
        "value": "4G_LTE Discount 30%",
        "valueType": "string"
      }
    ],
    "attachment": [
      {
        "url": "https://serverlocation/files/1",
        "name": "File_XYZ_001"
      }
    ],
    "sender": {
      "id": "10099",
```

```
      "name": "ABC Company",
      "phoneNumber": "10086"
    },
    "receiver": [
      {
        "id": "10234",
        "name": "Customer",
        "phoneNumber": "008613222211111",
        "party": {
          "id": "1091",
          "href": "https://serverlocation/tmf-api/partyManagement/v4/individual/1",
          "role": "customer",
          "name": "Sara White",
          "@referredType": "Individual"
        }
      }
    ]
  },
  {
    "id": "2708",
    "href": "https://serverlocation/tmf-api/communicationManagement/v4/communicationMessage/2708",
    "subject": "News: the latest promotion for you",
    "scheduledSendTime": "2020-02-10T00:00:00+01:00",
    "state": "initial",
    "description": "this is communication message for promotion",
    "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
    "messageType": "Email",
    "characteristic": [
      {
        "name": "$Parameter1",
        "value": "Mrs. Jones",
        "valueType": "string"
      },
      {
        "name": "$Parameter2",
        "value": "4G_LTE Discount 30%",
        "valueType": "string"
      }
    ],
    "attachment": [
      {
        "url": "https://serverlocation/files/2",
        "name": "File_XYZ_002"
      }
    ],
    "sender": {
      "id": "10099",
      "name": "ABC Company",
      "phoneNumber": "10086"
    },
    "receiver": [
      {
        "id": "10234",
        "name": "Customer",
        "phoneNumber": "008613222211111",
        "party": {
```

```
            "id": "1091",
            "href": "https://serverlocation/tmf-api/partyManagement/v4/individual/1",
            "role": "customer",
            "name": "Sara White",
            "@referredType": "Individual"
          }
        }
    ]
  }
]
```


Retrieve communication message

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

Description

This operation retrieves a communication message 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 specific communication message.

Request

```
GET /tmf-api/communicationManagement/v4/communicationMessage/3830
Accept: application/json
```

Response

200


```
{
  "id": "1708",
  "href": "https://serverlocation/tmf-api/communicationManagement/v4/communicationMessage/1708",
  "subject": "News: the latest promotion for you",
  "scheduledSendTime": "2020-02-10T00:00:00+01:00",
  "sendTime": "2020-02-10T00:00:00+01:00",
  "sendTimeComplete": "2020-02-10T00:10:00+01:00",
  "state": "completed",
  "description": "this is communication message for promotion",
  "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
  "characteristic": [
    {
      "name": "$Parameter1",
```

```
        "value": "Mr. Jones",
        "valueType": "string"
      },
      {
        "name": "$Parameter2",
        "value": "4G_LTE Discount 30%",
        "valueType": "string"
      }
    ],
    "attachment": [
      {
        "url": "https://serverlocation/files/1",
        "name": "File_XYZ_001"
      }
    ],
    "sender": {
      "id": "10099",
      "name": "ABC Company",
      "phoneNumber": "10086"
    },
    "receiver": [
      {
        "id": "10234",
        "name": "Customer",
        "phoneNumber": "008613811112222",
        "party": {
          "id": "991",
          "href": "https://serverlocation//tmf-api/partyManagement/v4/individual/1",
          "role": "customer",
          "name": "John White",
          "@referredType": "Individual"
        }
      }
    ]
}
```


Create communication message

POST /communicationMessage

Description

This operation creates a communication message entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a CommunicationMessage, 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
content	
receiver	

Mandatory Attributes	Rule
sender	
messageType	

Non Mandatory Attributes	Rule
attachment	
characteristic	
description	
logFlag	
priority	
scheduledSendTime	
sendTime	
sendTimeComplete	
state	
subject	
tryTimes	

Usage Samples

Here's an example of a request for creating a communication message. In this example the request is created in the initial state. The message will be available for sending by later setting the status to inProgress.

Request

```
POST /tmf-api/communicationManagement/v4/communicationMessage
Content-Type: application/json

{
   "subject": "News: the latest promotion for you",
   "scheduledSendTime": "2020-02-10T00:00:00+01:00",
   "state": "initial",
   "description": "this is communication message for promotion",
   "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
   "messageType": "SMS",
   "characteristic": [
      {
         "name": "$Parameter1",
         "value": "Mr. Jones",
         "valueType": "string"
      },
      {
         "name": "$Parameter2",
         "value": "4G_LTE Discount 30%",
         "valueType": "string"
      }
   ],
   "attachment": [
      {
         "url": "https://serverlocation/files/1",
```

```
        "name": "File_XYZ_001"
      }
    ],
    "sender": {
      "id": "10099",
      "name": "ABC Company",
      "phoneNumber": "10086"
    },
    "receiver": [
      {
        "id": "10234",
        "name": "Customer",
        "phoneNumber": "008613811112222",
        "party": {
          "id": "991",
          "href": "https://serverlocation/tmf-api/partyManagement/v4/individual/1",
          "role": "customer",
          "name": "John White",
          "@referredType": "Individual"
        }
      }
    ]
}
```


Response

201


```
{
    "id": "1708",
    "href": "https://serverlocation/tmf-api/communicationManagement/v4/communicationMessage/1708",
    "subject": "News: the latest promotion for you",
    "scheduledSendTime": "2020-02-10T00:00:00+01:00",
    "state": "initial",
    "description": "this is communication message for promotion",
    "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
    "messageType": "SMS",
    "characteristic": [
      {
        "name": "$Parameter1",
        "value": "Mr. Jones",
        "valueType": "string"
      },
      {
        "name": "$Parameter2",
        "value": "4G_LTE Discount 30%",
        "valueType": "string"
      }
    ],
    "attachment": [
      {
        "url": "https://serverlocation/files/2",
        "name": "File_XYZ_002"
```

```
      }
   ],
   "sender": {
      "id": "10099",
      "name": "ABC Company",
      "phoneNumber": "10086"
   },
   "receiver": [
      {
         "id": "10234",
         "name": "Customer",
         "phoneNumber": "008613811112222",
         "party": {
            "id": "991",
            "href": "https://serverlocation/tmf-api/partyManagement/v4/individual/1",
            "role": "customer",
            "name": "John White",
            "@referredType": "Individual"
         }
      }
   ]
}
```

Here's an example of a request for creating a communication message. In this example the request is created in the inProgress state and will be sent at the scheduled send time.

Request

```
POST /tmf-api/communicationManagement/v4/communicationMessage
Content-Type: application/json

{
   "subject": "News: the latest promotion for you",
   "scheduledSendTime": "2020-02-10T00:00:00+01:00",
   "state": "inProgress",
   "description": "this is communication message for promotion",
   "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
   "messageType": "Email",
   "characteristic": [
      {
         "name": "$Parameter1",
         "value": "Mrs. Jones",
         "valueType": "string"
      },
      {
         "name": "$Parameter2",
         "value": "4G_LTE Discount 30%",
         "valueType": "string"
      }
   ],
   "attachment": [
      {
```

```
        "url": "https://serverlocation/files/1",
        "name": "File_XYZ_001"
      }
    ],
    "sender": {
      "id": "10099",
      "name": "ABC Company",
      "phoneNumber": "10086"
    },
    "receiver": [
      {
        "id": "10234",
        "name": "Customer",
        "phoneNumber": "008613811112222",
        "party": {
          "id": "991",
          "href": "http://serverlocation/tmf-api/partyManagement/v4/individual/1",
          "role": "customer",
          "name": "John White",
          "@referredType": "Individual"
        }
      }
    ]
}
```


Response


```
201

{
    "id": "2708",
    "href": "https://serverlocation/tmf-api/communicationManagement/v4/communicationMessage/2708",
    "subject": "News: the latest promotion for you",
    "scheduledSendTime": "2020-02-10T00:00:00+01:00",
    "state": "initial",
    "description": "this is communication message for promotion",
    "content": "Dear $Parameter1, Here is the information of the promotion $Parameter2",
    "messageType": "Email",
    "characteristic": [
      {
        "name": "$Parameter1",
        "value": "Mrs. Jones",
        "valueType": "string"
      },
      {
        "name": "$Parameter2",
        "value": "4G_LTE Discount 30%",
        "valueType": "string"
      }
    ],
    "attachment": [
      {
        "url": "https://serverlocation/files/1",
```

```
        "name": "File_XYZ_001"
      }
    ],
    "sender": {
      "id": "10099",
      "name": "ABC Company",
      "phoneNumber": "10086"
    },
    "receiver": [
      {
        "id": "10234",
        "name": "Customer",
        "phoneNumber": "008613222211111",
        "party": {
          "id": "1091",
          "href": "https://serverlocation/tmf-api/partyManagement/v4/individual/1",
          "role": "customer",
          "name": "Sara White",
          "@referredType": "Individual"
        }
      }
    ]
}
```


Patch communication message

PATCH /communicationMessage/{id}

Description

This operation allows partial updates of a communication message 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
attachment	
characteristic	
content	
description	
logFlag	
messageType	
priority	

Patchable Attributes	Rule
receiver	
scheduledSendTime	
sendTime	
sendTimeComplete	
sender	
state	
subject	
tryTimes	

Non Patchable Attributes	Rule
href	
id	

Usage Samples

Here's an example of a request for updating a communication message. In this example, the message status is set to inProgress, together with a new scheduled send time.

Request

```
PATCH /tmf-api/communicationManagement/v4/communicationMessage/3830
Content-Type: application/merge-patch+json

{
   "scheduledSendTime": "2020-02-15T00:00:00+01:00",
   "state": "inProgress"
}
```

Response

```
200

{
   "scheduledSendTime": "2020-02-15T00:00:00+01:00",
   "state": "inProgress",
   "attachment": [
      {
         "url": "https://serverlocation/files/2",
         "name": "File_XYZ_002"
      }
   ]
}
```

Delete communication message

`DELETE /communicationMessage/{id}`

Description

This operation deletes a communication message entity.

Usage Samples

Here's an example of a request for deleting a communication message.

Request
DELETE /tmf-api/communicationManagement/v4/communicationMessage/3830
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

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
2.0.2	10-Sep-2018	Adrienne Walcott	Updated to reflect TM Forum Approved Status
4.0.0	28-May-2020	Knut Johannessen	Updated to reflect new schema based approach.
4.0.1	20-Jul-2020	Adrienne Walcott	Updated to reflect TM Forum Approved Status

Release History

Release Number	Date	Release led by:	Description
Release 1.0	04/15/2017	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of the Document.

Release Number	Date	Release led by:	Description
		Mariano Belaunde Orange Labs	
Release 2.0	11/06/2018	Mariano Belaunde Orange Labs	Alignment with Guidelines 3.0
Pre-production	28-May-2020	Knut Johannessen	Updated to reflect new schema based approach.
Production	20-Jul-2020	Adrienne Walcott	Updated to reflect TM Forum Approved Status

Contributors to Document

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