

# TM Forum Specification

## Communication API Conformance Profile

**TMF681B**

**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
- INTRODUCTION - API DESCRIPTION ..... 4
- RESOURCE MODEL CONFORMANCE ..... 5
  - Communication API MANDATORY AND OPTIONAL RESOURCES ..... 5
  - Communication Message Resource MANDATORY AND OPTIONAL ATTRIBUTES ..... 5
- NOTIFICATION MODEL CONFORMANCE ..... 9
  - API MANDATORY AND OPTIONAL NOTIFICATIONS..... 9
- API OPERATIONS CONFORMANCE ..... 10
  - API MANDATORY AND OPTIONAL OPERATIONS ..... 10
- API GET OPERATION CONFORMANCE ..... 11
  - GET /communicationMessage/{id}?fields=...&{filtering} ..... 11
  - GET /communicationMessage?fields=...&{filtering} ..... 11
- API POST OPERATION CONFORMANCE ..... 12
  - POST /communicationMessage..... 12
  - POST /communicationMessage/send ..... 14
  - POST /communicationMessage/{ID}/send..... 15
- API PATCH OPERATION CONFORMANCE ..... 16
  - PATCH /communicationMessage/{ID}..... 16
- API DELETE OPERATION CONFORMANCE ..... 18
  - DELETE /communicationMessage/{ID}..... 18
- API CONFORMANCE TEST SCENARIOS ..... 19
  - CommunicationMessage Resource TEST CASES..... 19
- ACKNOWLEDGEMENTS ..... 26
  - Release History ..... 26
  - Version History ..... 26

## INTRODUCTION - API DESCRIPTION

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 existing Communication Messages depending on filter criteria
- Retrieval of an existing Communication Message
- Creation of a new Communication Message
- Partial update of an existing Communication Message
- Deletion of an existing Communication Message
- Operations for sending Communication Message.
  - Send a new message with the whole communication message body (POST operation)
  - Send a message with the predefined communication message body (POST operation)

## RESOURCE MODEL CONFORMANCE

### COMMUNICATION API MANDATORY AND OPTIONAL RESOURCES

For the Resources defined by the API, here the following table indicates which are mandatory and which ones are optional.

Resource Name	Mandatory / Optional	Comments
Communication Message	M	

### COMMUNICATION MESSAGE RESOURCE MANDATORY AND OPTIONAL ATTRIBUTES

For the resources defined by the API the following table indicates which ones are mandatory and which ones are optional.

Parameter	Mandatory/Optional	Comments
@type	O	
@baseType	O	
@schemaLocation	O	
id	M (in response messages) O (otherwise)	Generated by the server and provided in the response upon resource creation.  Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
href	M (in response messages) O (otherwise)	Value in response must be the same as the one set in Location header provided upon entity creation

priority		O	
type		M	
subject		O	
content		M	
sendTime		O	
sendTimeComplete		O	
status		O	
description		O	
logFlag		O	
callbackFlag		O	
tryTimes		O	
version		O	
<b>characteristic</b>		<b>O</b>	<b>List of parameters. (CommunicationRequestCharacteristic[*])</b>
	name	M	
	value	M	
<b>sender</b>		<b>M</b>	<b>Sender</b>
	@type	O	
	@baseType	O	
	@schemaLocation	O	
	email	O	
	id	M	
	name	O	
	phoneNumber	O	
	relatedParty	O	relatedParty(RelatedParty Ref)
		href	O
		id	O

	name	O	
	role	O	
	@referredType	O	
<b>attachment</b>		O	<b>A list of attachments (Attachment[*])</b>
	@type	O	
	@baseType	O	
	@schemaLocation	O	
	id	O	
	name	O	
	path	O	
	description	O	
	href	O	
	mimeType	O	
	size	O	
	sizeUnit	O	
	url	O	
	validFor	O	
<b>Receiver</b>		<b>M</b>	<b>List of receivers(Receiver[*])</b>
	appUserId	O	
	email	O	
	id	M	
	ip	O	
	Name	O	
	phoneNumber	O	
	type	O	
	@type	O	
	@baseType	O	

	@schemaLocation	0	
	relatedParty	0	relatedParty(RelatedParty Ref)
	@referredType	0	
	href	0	
	id	0	
	name	0	
	role	0	



**NOTIFICATION MODEL CONFORMANCE**

The Pub/Sub models are common and described in the TMF REST Design Guidelines. Use the following templates to describe the Hub Mandatory and Optional attributes and filtering support.

For the Notifications defined by the API the following table indicates which ones are mandatory and which ones are optional.

**API MANDATORY AND OPTIONAL NOTIFICATIONS**

Resource Name	Mandatory or Optional	Comments
CommunicationMessageCreationNotification	O	
CommunicationMessageDeletionNotification	O	
CommunicationMessageUpdateNotification	O	

## API OPERATIONS CONFORMANCE

For every single resource use the following templates and define what operations are optional and what operations are mandatory.

### API MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for each one of the resources in the API (default is for all resources).

Uniform API Operation	Mandatory/Optional	Comments
GET	M for all resources	GET must be used to retrieve a representation of a resource
POST	M for all resources	POST must be used to create a new resource
PUT	O for all resources	PUT must be used to completely update a resource identified by its resource URI
PATCH	O for all resources	PATCH must be used to partially update a resource
DELETE	O for all resources	DELETE must be used to remove a resource

## API GET OPERATION CONFORMANCE

For every single resource use the following template to specify the mandatory and optional features supported by the GET operation.

GET	M	
Response Status Code 200	M	
Other Status Codes	NA	

### Definitions

**Filtered Search:** A filtered search can be applied using query parameters in order to obtain only the resource entities that meet the criteria defined by the filtering parameters included in the query request. Several elements can be applied to the filtered search. In that case logic, a logical AND is applied to combine the criteria (e.g.:?severity=<value> &status=<value>)

**Attribute selection (Filtered Response Data):** In order to apply a filter and limit the number of attributes included in the response, the GET request can include the “?fields=” query parameter. Several elements can be applied to the filter. In that case, a logical AND is applied to combine the values (e.g.:?fields=severity,status) will provide in the response only the values assigned to attributes category and channel. Attribute selection capabilities are the same for collections retrieval and individual resource queries

### GET /COMMUNICATIONMESSAGE/{ID}?FIELDS=...&{FILTERING}

This operation retrieves a communicationMessage entity.  
Attribute selection is mandatory for all first level attributes.  
Filtering on sub-resources is optional for all compliance levels.

### GET /COMMUNICATIONMESSAGE?FIELDS=...&{FILTERING}

This operation list communicationMessage entities.  
Attribute selection is mandatory for all first level attributes.  
Filtering is mandatory for first compliance level (L1) and optional otherwise.

## API POST OPERATION CONFORMANCE

For every single resource, use the following template to specify the mandatory and optional features supported by the POST operation.

### POST /COMMUNICATIONMESSAGE

This Uniform Contract operation is used to create a Communication Message entity.

POST	M	
Status Code 201	M	
Other Status Codes	NA	

The response to this operation must include a Location header set to `/communicationMessage/{ID}` where {ID} indicates the identifier assigned by the server to the new Communication Message resource created

#### Mandatory and Non Mandatory Attributes

The following tables specifies the attributes required when an entity is created (and their default values if not) as well as attributes with special considerations:

Attribute name	Mandatory/Option	Rule
id	O	Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
@type	O	
@baseType	O	
@schemaLocation	O	
callbackFlag	O	
content	O	

Attribute name	Mandatory/Option	Rule
description	O	
priority	O	
type	M	
subject	O	
content	M	
sendTime	O	
sendTimeComplete	O	
status	O	
subject	O	
tryTimes	O	
version	O	
characteristic	O	
sender	M	
sender.id	N	Mandatory if sender included.
receiver	M	
receiver.id	N	Mandatory if receiver included.
attachment	O	

Specify the attributes that must be included in the BODY of the response, even if they are not included in the request because they are mandatory in the definition of the resource to be created as per the resource model defined

- href
- id
- ...

**POST /COMMUNICATIONMESSAGE/SEND**

This is NON-CRUD operation which is used to create a Communication Message and send it from the “sender” to the “receiver”. When executing this API, the message will be directly sent to the receiver, i.e., the final customer.

POST	M	
Status Code 200	M	
Other Status Codes	NA	

The response to this operation must include a Location header set to /communicationMessage/{ID} where {ID} indicates the identifier assigned by the server to the new Communication Message resource created

**Mandatory and Non Mandatory Attributes**

The following tables specifies the attributes required when an entity is created (and their default values if not) as well as attributes with special considerations:

Attribute name	Mandatory/Option
@type	O
@baseType	O
@schemaLocation	O
callbackFlag	O
content	O
description	O
priority	O
type	M
subject	O
content	M

Attribute name	Mandatory/Option
sendTime	O
sendTimeComplete	O
status	O
subject	O
tryTimes	O
version	O
characteristic	O
sender	M
sender.id	M
receiver	M
receiver.id	M
attachment	O

Specify the attributes that must be included in the BODY of the response, even if they are not included in the request because they are mandatory in the definition of the resource to be created as per the resource model defined

- href
- id
- ...

### POST /COMMUNICATIONMESSAGE/{ID}/SEND

This is NON-CRUD operation which is used to send a pre-defined Communication message from the “sender” to the “receiver”.

POST	M	
Status Code 200	M	
Other Status Codes	NA	

## API PATCH OPERATION CONFORMANCE

### PATCH /COMMUNICATIONMESSAGE/{ID}

This Uniform Contract operation is used to partially update the representation of a Communication Message Resource.

PATCH	O	
Status Code 200	M	
Other Status Codes	NA	

Specify which attributes are patchable using the following table:

Attribute Name	Patchable ? (Y/N)	Rule
id	N	
href	N	
@type	N	
@baseType	N	
@schemaLocation	N	
callbackFlag	Y	
content	Y	
description	Y	
priority	Y	
type	Y	



Attribute Name	Patchable ? (Y/N)	Rule
subject	Y	
content	Y	
sendTime	Y	
sendTimeComplete	Y	
status	Y	
subject	Y	
tryTimes	Y	
version	Y	
characteristic	Y	
sender	Y	
sender.id	Y	
receiver	Y	
receiver.id	Y	
attachment	Y	

**API DELETE OPERATION CONFORMANCE****DELETE /COMMUNICATIONMESSAGE/{ID}**

This Uniform Contract operation is used to delete a CommunicationMessage entity.

DELETE	O	
Status Code 204	M	
Other Status Codes	NA	

## API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of the API.

Test Cases must be executed in the order defined for each resource because the result from one of the scenarios will be input for the next one.

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as {serverRoot}/communicationMessage/v2, where {serverRoot} defines the certification endpoint

## COMMUNICATIONMESSAGE RESOURCE TEST CASES

### Nominal Scenarios

#### TC\_Communication\_N1 – Create new Communication Message with minimum required information

- Send a POST message to {apiRoot}/communicationMessage/ with the following contents in the BODY

```
{  
  "type": "sms",  
  "content": "****",  
  "sender":{"id":"s1"},  
  "receiver": [{"id":"r1"}]  
}
```

- Wait for a response from the server with the following characteristics
  - Response Code 201-Created
  - Include a location header in the body set to /{apiRoot}/communicationMessage/{ID1} where {ID1} indicates the identifier assigned by the server to the new communication message resource
  - The response message includes all mandatory parameters
  - The body of the response matches the values set in the original request

- Send a GET message to /{apiRoot}/communicationMessage/

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Communication Message resource with ID set to {ID1}, the same identifier as assigned by the server to the new resource created
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID1} matches the values set in the original request
- Send a GET message to `/{apiRoot}/communicationMessage/{ID1}`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The response message includes all mandatory parameters
  - The body of the response includes a Communication Message resource structure that matches the values in the original request

#### **TC\_Communication\_N2 – Create new Communication Message with minimum set of parameters supported by server**

- Send a POST message to `{apiRoot}/communicationMessage/` with the following contents in the BODY

```
{
  "id": "123",
  "type": "email",
  "content": "***",
  "sender":{"id":"s2"},
  "receiver": [{"id":""}],
  "characteristic": [{"name":"","value":""}]
}
```

- Wait for a response from the server with the following characteristics

- Response Code 201-Created
- Include a location header in the body set to `/{apiRoot}/communicationMessage/{ID2}` where {ID2} indicates the identifier assigned by the server to the new Communication Message resource
- The response message includes all mandatory parameters
- The body of the response matches the values set in the original request
- Send a GET message to `/{apiRoot}/communicationMessage/`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Communication Message resource with ID set to {ID2}, the same identifier as assigned by the server to the new resource created
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID2} matches the values set in the original request
- Send a GET message to `/{apiRoot}/communicationMessage/{ID2}`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The response message includes all mandatory parameters
  - The body of the response includes a Communication Message resource structure that matches the values in the original request

**TC\_Communication\_N3 – Search for CommunicationMessages with specific characteristics**

- Send a GET message to `/{apiRoot}/communicationMessage`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes at least two communication message resources referring to {ID1} and {ID2}
  - The body of the response for the resource with each identifier matches the values in the corresponding original request
- Send a GET message to `/{apiRoot}/communicationMessage?type=sms`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Communication Message resource referring to {ID1} and there is no reference to Communication Message resource {ID2}
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID1} matches the values in the original request
- Send a GET message to `/{apiRoot}/communicationMessage?sender.id=s1`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Communication Message resource referring to {ID1} and there is no reference to Communication Message resource {ID2}
  - The response message includes all mandatory parameters

- The body of the response for the resource with identifier {ID1} matches the values in the original request

#### **TC\_Communication\_N4 – Filtered retrieval of CommunicationMessages**

- Send a GET message to  
`/{apiRoot}/communicationMessage/{ID1}?fields=content`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Communication Message resource referring to {ID1} and including only attributes content , matching the values in the original request
- Send a GET message to `/{apiRoot}/communicationMessage/{ID2}?fields=content,type`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Communication Message resource referring to {ID2} and including only attributes content and type, matching the values in the original request

Notice that this test case is using parameters " content" and " type" to filter the data included in the response but any other parameter could be used

#### **TC\_Communication\_N5 – Filtered Search and Filtered data response**

- Send a GET message to `/{apiRoot}/communicationMessage? type=sms& sender.id=s1`
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

- The body of the response includes one Communication Message resource referring to {ID1} and there is no reference to Communication Message resource {ID2}
- The body of the response for the resource with each identifier includes only attribute description, matching the values in the corresponding original request

Notice that this test case is using the parameter “type” and “sender.id” to filter the data included in the response but any other parameter could be used

## Error Scenarios

### TC\_Communication\_E1 – Unknown Communication Message identifier

- Send a GET message to `{apiRoot}/communicationMessage/{ID3}`, where {ID3} does not match any of the identifiers previously created in the server
- Wait for a response from the server with the following characteristics
  - Response Code 404-Not Found

### TC\_Communication\_E2 – Invalid Request – Missing mandatory parameter

- Send a POST message to `{apiRoot}/communicationMessage` with the following contents in the BODY.

```
{
  "type": "sms",
  "content": "s1"
}
```

Notice that this request is missing mandatory parameter “sender” and “receiver” , but any other mandatory parameter could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request



**TC\_Communication\_E3 – Invalid Request – Missing parameter mandatory in sender**

- Send a POST message to {apiRoot}/communicationMessage/ with the following contents in the BODY.

```
{  
  
  "type": "****",  
  "content": "****",  
  "sender":{"name":""},  
  "receiver":[{"id":""}]  
}
```

Notice that this request is missing mandatory parameters “id” when information element “sender” is included in the request, but any other parameter that becomes mandatory based on the context could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

Note: Not all the test cases are included in this document. There are more test cases in CTK file which is coming soon.

## ACKNOWLEDGEMENTS

## RELEASE HISTORY

Release Number	Date	Release led by:	Description
17.5.0	January 2018	Hongxia Hao, Ma Xu	Initial Release
18.0.0	June 2018	Hongxia Hao	Second Release

## VERSION HISTORY

Version Number	Date	Modified by	Description
Version 1.0.0	8/05/2017	Ma Xu <a href="mailto:Maxu@huawei.com">Maxu@huawei.com</a>  Hongxia Hao <a href="mailto:haohongxia@huawei.com">haohongxia@huawei.com</a>	initial version
Version 1.01	14-Jun-2017		Updated version
Version 1.02	9-Sep-2017	Hongxia Hao  Ma Xu	Align with the newest specification document
Version 1.0.3	23-Jan-2018	Adrienne Walcott	Formatting/style edits prior to publishing
Version 2.0.0	12-June-2018	Hongxia Hao	Modify some typos.  Change the descriptions of @attributes related with DG3.0  Add test cases
Version 2.0.1	29-Jun-2018	Adrienne Walcott	Formatting/style edits prior to R18 publishing.