

TM Forum Specification

Agreement Management API Conformance Profile

TMF651B

Release 18.0.0

April 2018

Latest Update: TM Forum Release 18.0.0	Member Evaluation
Version 2.0.1	IPR Mode: RAND

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

TABLE OF CONTENTS

- NOTICE 2
- TABLE OF CONTENTS 3
- INTRODUCTION - API DESCRIPTION 4
- RESOURCE MODEL CONFORMANCE 5
 - API MANDATORY AND OPTIONAL RESOURCES 5
 - Agreement MANDATORY AND OPTIONAL ATTRIBUTES 5
 - Agreement Specification MANDATORY AND OPTIONAL ATTRIBUTES 6
- NOTIFICATION MODEL CONFORMANCE 8
 - API MANDATORY AND OPTIONAL NOTIFICATIONS 8
- API OPERATIONS CONFORMANCE 9
 - API MANDATORY AND OPTIONAL OPERATIONS 9
- API GET OPERATION CONFORMANCE 10
 - /agreement?fields=...&{filtering} 10
 - /agreement/{id}?fields=...&{filtering} 10
 - /agreementSpecification?fields=...&{filtering} 11
 - /agreementSpecification/{id}?fields=...&{filtering} 11
- API POST OPERATION CONFORMANCE 12
 - /agreement 12
 - /agreementSpecification 13
- API PUT OPERATION CONFORMANCE 15
- API PATCH OPERATION CONFORMANCE 16
 - /agreement/{id} 16
 - /agreementSpecification/{id} 17
- API DELETE OPERATION CONFORMANCE 20
 - /agreement/{id} 20
 - /agreementSpecification/{id} 20
- API CONFORMANCE TEST SCENARIOS 21
 - Agreement resource TEST CASES 21
- ACKNOWLEDGEMENTS 26
 - Version History 26
 - Release History 26

INTRODUCTION - API DESCRIPTION

See the introduction of the related specification document.

RESOURCE MODEL CONFORMANCE

API MANDATORY AND OPTIONAL RESOURCES

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

Resource Name	Mandatory or Optional	Comments
Agreement	M	
AgreementSpecification	M	

AGREEMENT MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "Agreement"

Attribute Name	Mandatory or Optional	Comments
agreementPeriod	O	
completionDate	O	Attribute non patchable
description	O	
documentNumber	O	
href	M (in response messages) O (otherwise)	Url for the created resource
id	M (in response messages) O (otherwise)	Generated by the server
initialDate	O	
name	M (for resource creation) O (otherwise)	
statementOfIntent	O	

Attribute Name	Mandatory or Optional	Comments
status	O	
type	M (for resource creation) O (otherwise)	
version	O	
agreementSpecification	O	
agreementItem	M (for resource creation) O (otherwise)	
engagedPartyRole	M (for resource creation) O (otherwise)	
agreementAuthorization	O	
characteristic	O	
associatedAgreement	O	

AGREEMENT SPECIFICATION MANDATORY AND OPTIONAL ATTRIBUTES

The table below summarizes mandatory and optional attributes for resource "AgreementSpecification"

Attribute Name	Mandatory or Optional	Comments
description	O	
href	M (in response messages) O (otherwise)	Url for the created resource
id	M (in response messages) O (otherwise)	Generated by the server
isBundle	O	
lastUpdate	O	
lifecycleStatus	O	

Attribute Name	Mandatory or Optional	Comments
name	M (for resource creation) O (otherwise)	
validFor	O	
version	O	
serviceCategory	O	
specCharacteristic	O	
relatedParty	O	
attachment	M (for resource creation) O (otherwise)	
specificationRelationship	O	

NOTIFICATION MODEL CONFORMANCE

<Optional section in basic CONNECT certification>

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.

API MANDATORY AND OPTIONAL NOTIFICATIONS

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

<The table below is a reference example (actual notifications may depend on the specific API)>

Notification Name	Mandatory or Optional	Comments
AgreementCreationNotification	M	
AgreementAttributeValueChangeNotification	M	
AgreementStateChangeNotification	M	
AgreementRemoveNotification	M	

All attributes of the resource associated with the notification are mandatory

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

Fill the following table and indicate 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 resources: Agreement AgreementSpecification	POST must be used to create a new resource
PUT	O for resources: Agreement AgreementSpecification	PUT must be used to completely update a resource identified by its resource URI
PATCH	M for resources: Agreement AgreementSpecification	PATCH must be used to partially update a resource
DELETE	M for resources: Agreement AgreementSpecification	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.

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

<Repeat for each resource in the API where GET operation can be exercised>

All the GET operations in this API share the same status code pattern.

GET	M	
Response Status Code 200	M	
Other Status Codes	NA	<describe any other relevant response or error code>

/AGREEMENT?FIELDS=...&{FILTERING}

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

/AGREEMENT/{ID}?FIELDS=...&{FILTERING}

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

/AGREEMENTSPECIFICATION?FIELDS=...&{FILTERING}

This operation list agreement specification entities.

Attribute selection is mandatory for all first level attributes.

Filtering is mandatory for first compliance level (L1) and optional otherwise.

/AGREEMENTSPECIFICATION/{ID}?FIELDS=...&{FILTERING}

This operation retrieves an agreement specification entity.

Attribute selection is mandatory for all first level attributes.

Filtering on sub-resources is optional for all compliance levels.

API POST OPERATION CONFORMANCE

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

<Repeat for each resource in the API where POST operation can be exercised>

All the POST operations in this API share the same status code pattern.

POST	M	
Status Code 201	M	
Other Status Codes	NA	<describe any other relevant response or error code>

/AGREEMENT

This operation creates an agreement entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	
type	
engagedPartyRole	
agreementItem	

Non Mandatory Attributes	Default Value	Rule
agreementPeriod		
completionDate	Current date	
description		

Non Mandatory Attributes	Default Value	Rule
documentNumber		
initialDate		
statementOfIntent		
status		
version	0	
agreementSpecification		
agreementAuthorization		
characteristic		
associatedAgreement		

Additional Rules

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating an Agreement resource.

Context	Mandatory Sub-Attributes
engagedPartyRole	id, name
associatedAgreement	id, href

Default Values Summary

When creating the resource, the following table summarizes the default values applicable to optional attributes of the resource (or sub-resources).

Attributes	Default Value
id	Automatically generated
completionDate	Current date
version	0

/AGREEMENTSPECIFICATION

This operation creates an agreement specification entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	
attachment	

Non Mandatory Attributes	Default Value	Rule
description		
isBundle	false	
lastUpdate		
lifecycleStatus		
validFor		
version		
serviceCategory		
specCharacteristic		
relatedParty		
specificationRelationship		

Default Values Summary

When creating the resource, the following table summarizes the default values applicable to optional attributes of the resource (or sub-resources).

Attributes	Default Value
isBundle	false

API PUT OPERATION CONFORMANCE

<This section is only required if the PUT operation is marked as mandatory for the basic CONNECT certification>

<Repeat for each resource in the API where PUT operation can be exercised>

All the PUT operations in this API share the same status code pattern.

PUT	O	
Status Code 201	M	
Other Status Codes	NA	<describe any other relevant response or error code>

API PATCH OPERATION CONFORMANCE

<This section is only required if the PATCH operation is marked as mandatory for the basic CONNECT certification>

<Repeat for each resource in the API where PUT operation can be exercised>

All the PATCH operations in this API share the same status code pattern.

PATCH	O	
Status Code 201	M	
Other Status Codes	NA	<describe any other relevant response or error code>

/AGREEMENT/{ID}

This operation allows partial updates of an agreement 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.

Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
agreementAuthorization	
agreementItem	
agreementPeriod	
description	
documentNumber	
initialDate	
name	

statementOfIntent
 status
 type
 version
 agreementSpecification
 engagedPartyRole
 characteristic
 associatedAgreement

Non Patchable Attributes Rule

id
 href
 completionDate
 agreementPeriod
 description
 documentNumber
 initialDate
 name
 statementOfIntent
 status
 type
 version
 agreementSpecification
 engagedPartyRole
 characteristic
 associatedAgreement

/AGREEMENTSPECIFICATION/{ID}

This operation allows partial updates of an agreement specification 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.

Notice that patching is possible only for 'admin' API users.

Patchable Attributes	Rule
description	
isBundle	
lastUpdate	
lifecycleStatus	
name	
validFor	
version	
serviceCategory	
specCharacteristic	
relatedParty	
attachment	
specificationRelationship	

Non Patchable Attributes	Rule
id	
href	
description	
isBundle	
lastUpdate	
lifecycleStatus	
name	

validFor

version

serviceCategory

specCharacteristic

relatedParty

attachment

specificationRelationship

API DELETE OPERATION CONFORMANCE

<This section is only required if the DELETE operation is marked as mandatory for the basic CONNECT certification>

<Repeat for each resource in the API where DELETE operation can be exercised>

All the DELETE operations in this API share the same status code pattern.

DELETE	O	
Status Code 200	M	
Other Status Codes	NA	<describe any other relevant response or error code>

/AGREEMENT/{ID}

This operation deletes an agreement entity.

/AGREEMENTSPECIFICATION/{ID}

This operation deletes an agreement specification entity.

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}/agreement/v1, where {serverRoot} defines the certification endpoint

WE describe below test cases for Agreement resource. For the Agreement Specification Resource, the test cases are identical (simple replace “agreement” by “agreementSpecification” in the examples.

Agreement resource TEST CASES

Nominal Scenarios

TC_Trou_N1 – Create new Agreement with minimum required information

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

```
{
  "name": "<anytext>",
}
```

- 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}/agreement/{IDtt1} where {IDtt1} indicates the identifier assigned by the server to the new Agreement resource
 - The response message includes all mandatory parameters (including description, severity and type that were not sent in the original request)
 - The body of the response matches the values set in the original request

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

- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one Agreement resource with ID set to {IDtt1}, 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 {IDtt1} matches the values set in the original request
- Send a GET message to `/{apiRoot}/agreement/{IDtt1}`
- 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 Agreement resource structure that matches the values in the original request

TC_Trou_N2 – Create new Agreement with minimum set of parameters supported by server

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

```
{
  "name": "nanana",
  "characteristic": [
    {
      "name": "foo.",
      "value": "bar"
    }
  ]
}
```

- 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}/account/{IDtt2}` where `{IDtt2}` indicates the identifier assigned by the server to the new Agreement resource
- The response message includes all mandatory parameters that were not sent in the original request)
- The body of the response matches the values set in the original request
- Send a GET message to `/{apiRoot}/agreement/`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one Agreement resource with ID set to `{IDtt2}`, 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 `{IDtt2}` matches the values set in the original request
- Send a GET message to `/{apiRoot}/agreement/{IDtt2}`
- 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 Agreement resource structure that matches the values in the original request

TC_Trou_N3 – Search for Agreements with specific characteristics

- Send a GET message to `/{apiRoot}/agreement`

- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes at least two Agreement resources referring to {IDtt1} and {IDtt2}
 - 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}/ Agreement?status=Active`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one Agreement resource referring to {IDtt1} and there is no reference to Agreement resource {IDtt2}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {IDtt1} matches the values in the original request

TC_Trou_N4 – Filtered retrieval of Agreements

- Send a GET message to `/{apiRoot}/agreement/{IDtt1}?fields=name,status`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one Agreement resource referring to {IDtt1} and including only attributes name and status, matching the values in the original request

Error Scenarios

TC_Trou_E1 – Unknown Agreement identifier

- Send a GET message to `/{apiRoot}/agreement/{IDtt3}`, where {IDtt3} 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_Trou_E2 – Invalid Request – Missing mandatory parameter

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

```
{
  "status": true,
}
```

Notice that this request is missing mandatory parameter "name" 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_Trou_E3 – Invalid Request – Missing parameter mandatory in context

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

```
{
  "name": "some name",
  "characteristic":
  {
    "value": "coco"
  }
}
```

Notice that this request is missing mandatory parameters "name" in the included sub object.

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

ACKNOWLEDGEMENTS

VERSION HISTORY

Version Number	Date	Release led by:	Description
1.0	07/15/2015	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of Draft Version of the Document.
1.1	15-Mar-2017	Mariano Belaunde Orange	Updated version including Test scenarios
2.0	24-Apr-2018	Mariano Belaunde Orange	Updated version to be aligned with Guidelines 3.0
2.0.1	27-Jun-2018	Adrienne Walcott	Formatting/style edits prior to R18 publishing

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 18.0.0	25-Jun-2018	Pierre Gauthier	Initial Release