

TM Forum Specification

Agreement Management API Conformance Profile

TMF668B
Release 19.0.0
July 2019

Release: R 19.0.0	Status: Member Evaluation
Version: 4.0.0	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2019. 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. +1 973 998 7916

TM Forum Web Page: www.tmforum.org

Table of Contents

NOTICE	2
Table of Contents.....	3
INTRODUCTION - API DESCRIPTION.....	5
RESOURCE MODEL CONFORMANCE.....	6
API MANDATORY AND OPTIONAL RESOURCES	6
AgreementSpecification MANDATORY AND OPTIONAL ATTRIBUTES	6
Agreement MANDATORY AND OPTIONAL ATTRIBUTES.....	7
NOTIFICATION MODEL CONFORMANCE	9
API MANDATORY AND OPTIONAL NOTIFICATIONS.....	9
API OPERATIONS CONFORMANCE.....	10
AgreementSpecification MANDATORY AND OPTIONAL OPERATIONS.....	10
Agreement MANDATORY AND OPTIONAL OPERATIONS	10
API GET FILTERING OPERATION CONFORMANCE.....	11
GET /AgreementSpecification?fields=...&{filtering}	11
GET /AgreementSpecification/{ID}?fields=...&{filtering}	11
GET /Agreement?fields=...&{filtering}	11
GET /Agreement/{ID}?fields=...&{filtering}.....	11
API POST OPERATION CONFORMANCE	12
POST /AgreementSpecification/	12
POST /Agreement/.....	12
API PATCH OPERATION CONFORMANCE.....	13
PATCH /agreementSpecification/	13
PATCH /agreement/.....	13
API DELETE OPERATION CONFORMANCE.....	14
DELETE /agreementSpecification/	14
DELETE /agreement/.....	14
API CONFORMANCE TEST SCENARIOS.....	15
AgreementSpecification and Agreement resource TEST CASES.....	15
Document History.....	19

Release History 19

INTRODUCTION - API DESCRIPTION

The Agreement API provides standardized mechanism for managing agreements, especially in the context on partnerships between partners.

The API allows creation, update and query of *agreement* instances as well as creation, update and query of *agreement specifications* – serving as templates for agreement instances.

The following APIs are typically used in conjunction with this API:

- Product Catalog Management API to refer to the product offerings that are linked to the agreements or agreement specifications.
- Party Management API to refer to the engaged parties that have signed agreements

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
AgreementSpecification	M	
Agreement	M	

AgreementSpecification MANDATORY AND OPTIONAL ATTRIBUTES

The table below reminds mandatory and optional attributes for resource "AgreementSpecification". For the list of mandatory and optional attributes of all contained entities see the specification document Agreement Management API R19.0.0.

Attribute Name	Mandatory or Optional	Comments
@type	M (in response messages) M (in request in polymorphic context: see comment) O (otherwise)	@type is mandatory if we create a user-specific sub-class via polymorphism mechanism.
@baseType	O (in request and response in polymorphic context: see comment)	@baseType is mandatory if we create a user-specific sub-class via polymorphism mechanism.
@schemaLocation	O (in request and response in polymorphic context: see comment)	@schemaLocation is mandatory if we create a user-specific sub-class via polymorphism mechanism.
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
id	M (in response messages)	Id is auto-generated at creation time
name	M	
description	O	
isBundle	O	
lastUpdate	O	
lifecycleStatus	O	

Attribute Name	Mandatory or Optional	Comments
validFor	O	
version	O	
serviceCategory	O	
attachment	M	
relatedParty	O	
specificationCharacteristic	O	
specificationRelationship	O	

Agreement MANDATORY AND OPTIONAL ATTRIBUTES

The table below reminds mandatory and optional attributes for resource "Agreement". For all the details on the attributes of sub-entities or referred entities being used see the specification document Agreement Management API R19.0.0.

Attribute Name	Mandatory or Optional	Comments
@type	M (in response messages) M (in request in polymorphic context: see comment) O (otherwise)	@type is mandatory if we create a user-specific sub-class via polymorphism mechanism.
@baseType	O (in request and response in polymorphic context: see comment)	@baseType is mandatory if we create a user-specific sub-class via polymorphism mechanism.
@schemaLocation	O (in request and response in polymorphic context: see comment)	@schemaLocation is mandatory if we create a user-specific sub-class via polymorphism mechanism.
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
id	M (in response messages)	Id is auto-generated at creation time
agreementPeriod	O	
agreementType	M	
completionDate	O	
description	O	
documentNumber	O	
initialDate	O	
name	M	
statementOfIntent	O	
status	O	
version	O	
associatedAgreement	O	
engagedPartyRole	M	
characteristic	O	
agreementItem	O	

Attribute Name	Mandatory or Optional	Comments
agreementSpecification	O	
agreementAuthorization	O	

NOTIFICATION MODEL CONFORMANCE

The Pub/Sub models are common and described in the TMF REST Design Guidelines.

API MANDATORY AND OPTIONAL NOTIFICATIONS

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

Notification Name	Mandatory or Optional	Comments
AgreementSpecificationCreateEvent	O	
AgreementSpecificationAttributeValueChangeEvent	O	
AgreementSpecificationStateChangeEvent	O	
AgreementSpecificationDeleteEvent	O	
AgreementCreateEvent	O	
AgreementAttributeValueChangeEvent	O	
AgreementStateChangeEvent	O	
AgreementDeleteEvent	O	

API OPERATIONS CONFORMANCE

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

AgreementSpecification MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for AgreementSpecification resource:

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource
POST	M	POST must be used to create a new resource
PATCH	M (JSON-MERGE) O (JSON-PATCH)	PATCH must be used to partially update a resource. Support of Json merge is mandatory, support of json-path is optional.
DELETE	M	DELETE must be used to remove a resource

Agreement MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for Agreement resource:

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource
POST	M	POST must be used to create a new resource
PATCH	M (JSON-MERGE) O (JSON-PATCH)	PATCH must be used to partially update a resource. Support of Json merge is mandatory, support of json-path is optional.
DELETE	M	DELETE must be used to remove a resource

API GET FILTERING OPERATION CONFORMANCE

This section defines the conformance for retrieval operations in this API.
The list of retrieval operations in this API are:

GET /AgreementSpecification?fields=...&{filtering}

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

GET /Agreement?fields=...&{filtering}

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

These operations allow retrieval of AgreementSpecifications and Agreement entities.

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

GET	M	
Response Status Code 200	M	With JSON resource objects as body response.
Response Status Code 404	M	When resource can't be found
Other Status Codes	N/A	

We remind below the filtered search and attribute selection mechanisms that are defined generically.

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.:?category=<value> &state=<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=category,state) will provide in the response only the values assigned to attributes category and state. Attribute selection capabilities are the same for collections retrieval and individual resource queries

Attribute selection is enabled for all first level attributes.

Filtering is mandatory to avoid answer with too many occurrences.

When using the form with {ID}, the Uniform Contract operation is used to returns a specific resource with given {ID}, where {ID} indicates the identifier assigned by the server when a resource was created.

Attribute selection is mandatory for all first level attributes. Filtering on sub-resources is optional for all compliance levels.

API POST OPERATION CONFORMANCE

This section defines the conformance for creation operations in this API.

The following operations are used to create Agreement specifications and Agreement resources in the server.

POST /AgreementSpecification/

POST /Agreement/

The response to these operations must include a Location header set to {baseUrl}/{resource}/{ID} where {ID} indicates the identifier assigned by the server to the new resource created.

The table below shows the expected return codes.

POST	M	
Response Status Code 201	M	
Other Status Codes	NA	

The response from the server must include a BODY with the contents of the new resource created, filled with at least the same information elements that were included in the request and are supported by the server. Notice that the value stored by the server may be different than the one set in the request.

If the POST request includes optional parameters (as per the model resource definition) that are not supported by the server, then the server must reject the request (replying with a 4xx error response) indicating the parameter not supported.

The BODY of the response from the server must include attribute "href" set to the same value as the one in the Location header.

For all the details on mandatory or optional attributes when creating resources, see the Agreement Management API R19.0.0 specification document.

API PATCH OPERATION CONFORMANCE

This section defines the conformance for update operations in this API.

The following operations are used to update Agreement specifications and Agreement resources in the server.

PATCH /agreementSpecification/

PATCH /agreement/

The table below shows the expected return codes.

PATCH	M	
Response Status Code 200	M	
Other Status Codes	NA	

See the Agreement Management API specification document for the list of attributes that are patchable.

API DELETE OPERATION CONFORMANCE

This section defines the conformance for delete operations in this API.

The following operations are used to delete Agreement specifications and Agreement resources in the server.

DELETE /agreementSpecification/

DELETE /agreement/

The table below shows the expected return codes.

POST	M	
Response 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 Agreement Management 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}/agreementManagement/v4, being {serverRoot} defines the certification endpoint

AgreementSpecification and Agreement resource TEST CASES

Pre-requisites of nominal scenarios

The nominal scenarios assume that there are attachments that already exist in the system.

The following Attachment is assumed existence in the system:

```
{
  "id": "44",
  "attachmentType": "Video",
  "url": "http://xxxxx"
}
```

The nominal scenarios assume that there are some organizations that already exist in the system.

The following Organization need to be created in the system:

```
{
  "id": "330",
  "role": "Supplier",
  "name": "Magic Tools Company"
}
```

The nominal scenarios assume that there are some product offerings that already exist in the system.

The following productOffering resource need to be created using the Product Offering API.

```
{
  "id": "9085",
  "name": "My Quick BB Offer"
}
```

Nominal Scenarios

TC_Agreement_N1 – Create new Agreement Specification with some Role specifications

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

```
{
  "name": "General Agreement Specification",
  "attachment": [
    {
      "id": "44",
```

```

    "attachmentType": "Video",
    "url": "http://xxxxx"
  }
]
}

```

- 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}/agreementSpecification/{ID1}` where `{ID1}` indicates the identifier assigned by the server to the agreement specification resource
 - The response message includes all mandatory parameters (“name” in this case)
 - The response message adds `@type`, `id` and `href` in `AgreementSpecification` object
 - The body of the response matches the values set in the original request

TC_Agreement_N2 – retrieve the created agreement specification

- Send a GET message to `{apiRoot}/agreementSpecification/{ID1}` (ID1 comes from TC_Agreement_N1)
- 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 Specification` resource structure that matches the values in the original request

TC_Agreement_N3 – create a agreement instance cthat complies with the agreement specification

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

```

{
  "name": "Summer Contract Agreement",
  "agreementType": "commercial",
  "agreementItem": [
    {
      "productOffering": [
        {
          "@referredType": "ProductOffering",
          "id": "9085",
          "name": "My Quick BB Offer"
        }
      ],
      "termOrCondition": [
        {
          "description": "This agreement term or condition ...",
          "id": "5890",
          "validFor": {
            "startDateTime": "2018-04-25T00:00Z",
            "endDateTime": "2018-11-20T00:00Z"
          }
        }
      ]
    }
  ],
  "engagedPartyRole": [
    {
      "@referredType": "Organization",

```



```
"id": "330",  
"role": "Supplier",  
"name": "Magic Tools Company"  
}  
]  
}
```

- 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/{ID3}` where `{ID3}` indicates the identifier assigned by the server to the agreement instance resource
 - The response message includes all mandatory parameters (“name” in this case)
 - The response message adds `@type`, `id` and `href` in the created Agreement object
 - The body of the response matches the values set in the original request

TC_Agreement_N4 – retrieve the created agreement instance

- Send a GET message to `{apiRoot}/agreement/{ID3}` (`ID3` comes from `TC_Agreement_N3`)
- 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

Error Scenarios

TC_Agreement_E1 – Unknown AgreementSpecification identifier

- Send a GET message to `{apiRoot}/agreementSpecification/{ID4}`, where `{ID4}` 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_Agreement_E2 – Unknown Agreement identifier

- Send a GET message to `{apiRoot}/agreement/{ID5}`, where `{ID5}` 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_Agreement_E3 – Invalid Request on Agreement Specification – Missing mandatory parameter

- Send a POST message to `{apiRoot}/agreementSpecification` where no “name” is passed in the payload
- Wait for an error response from the server indicating the mandatory parameter is missing in the request

TC_Agreement_E4 – Invalid Request on Agreement – Missing mandatory parameter

- Send a POST message to `{apiRoot}/agreementSpecification` where no “name” is passed in the payload

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

Document History

Release History

Version Number	Date	Release led by:	Description
Release 1.0	15/04/2016	Pierre Gauthier TM Forum pgauthier@tmforum.org Mariano Belaunde Orange mariano.belaunde@orange.com	First Release of the Document. Generated from the API Data Model.
Release 2.0	24/04/2018	Mariano Belaunde	Update with minor changes to comply with Guidelines V3.0
Release 4.0	30/05/2019	Mariano Belaunde Orange Labs	Regeneration after schematization