TM Forum Specification

Partnership 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

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
PartnershipSpecification MANDATORY AND OPTIONAL ATTRIBUTES 6
Partnership MANDATORY AND OPTIONAL ATTRIBUTES 7
API OPERATIONS CONFORMANCE 8
PartnershipSpecification MANDATORY AND OPTIONAL OPERATIONS 9
Partnership MANDATORY AND OPTIONAL OPERATIONS 9
API GET FILTERING OPERATION CONFORMANCE 10
GET /partnershipSpecification?fields=…&{filtering} 10
GET /partnershipSpecification/{ID}?fields=…&{filtering} 10
GET /partnership?fields=…&{filtering} 10
GET /partnership/{ID}?fields=…&{filtering} 10
API POST OPERATION CONFORMANCE 11
POST /partnershipSpecification/ 11
POST /partnership/ 11
API PATCH OPERATION CONFORMANCE 12
PATCH /partnershipSpecification/ 12
PATCH /partnership/ 12
API DELETE OPERATION CONFORMANCE 13
DELETE /partnershipSpecification/ 13
DELETE /partnership/ 13
API CONFORMANCE TEST SCENARIOS 14
PartnershipSpecification and Partnership resource TEST CASES 14
Document History 18
Version History 18
Release History 18

INTRODUCTION - API DESCRIPTION

The following document is the REST API Conformance for the Partnership Management API (TMF668).

Basically a partnership specification is defined by declaring a list of role specifications that the partners can play. A role specification is typically characterized by an agreement specification. In addition some role specifications imply the necessity to be linked to billing accounts. A Partnership instance in the other hand refers to its partnership specification and declares the list of partners – that is to say – the parties playing a given role in the partnership.

The Partnership Management API allows to:

- Create, update, delete or retrieve partnership specifications

- Create, update, delete or retrieve partnership instances

For the details of this API see the specification document Partnership Management API R19.0.0.

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
PartnershipSpecification	M	
Partnership	M	

PartnershipSpecification MANDATORY AND OPTIONAL ATTRIBUTES

The table below reminds mandatory and optional attributes for resource "PartnershipSpecification". For the list of mandatory and optional attributes of all contained entities see the specification document Partnership 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	
roleSpecification	O	An Array of role specifications contained by value
name	M	

Attribute Name			Mandatory or Optional	Comments
	description		O	
	requiresBilling		O	
	requiresSettlement		O	
	mimeType		O	
	name		O	
	url		O	
	agreementSpecification		O	An array of references to agreement specifications
		id	M	
		href	O	
		name	O	
		description	O	
		@referredType	M (Only if polymorphism is in use) O (Otherwise)	

Partnership MANDATORY AND OPTIONAL ATTRIBUTES

The table below reminds mandatory and optional attributes for resource "Partnership". For all the details on the attributes of sub-entities or referred entities being used see the specification document Partnership 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	
specification	M	
partner	O	An Array of partner structures contained by value. At least one partner is expected at partnership is completed.

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
PartnershipSpecificationCreateEvent	O	
PartnershipSpecificationAttributeValueChangeEvent	O	
PartnershipSpecificationStateChangeEvent	O	
PartnershipSpecificationDeleteEvent	O	
PartnershipCreateEvent	O	
PartnershipAttributeValueChangeEvent	O	
PartnershipStateChangeEvent	O	
PartnershipDeleteEvent	O	

API OPERATIONS CONFORMANCE

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

PartnershipSpecification MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for PartnershipSpecification 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

Partnership MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for Partnership 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 /partnershipSpecification?fields=…&{filtering}

GET /partnershipSpecification/{ID}?fields=…&{filtering}

GET /partnership?fields=…&{filtering}

GET /partnership/{ID}?fields=…&{filtering}

These operations allow retrieval of PartnershipSpecifications and Partnership 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 partnership specifications and partnership resources in the server.

POST /partnershipSpecification/

POST /partnership/

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 Partnership 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 partnership specifications and partnership resources in the server.

PATCH /partnershipSpecification/

PATCH /partnership/

The table below shows the expected return codes.

PATCH	M	
Response Status Code 200	M	
Other Status Codes	NA	

See the Partnership 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 partnership specifications and partnership resources in the server.

DELETE /partnershipSpecification/

DELETE /partnership/

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 Partnership 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}/partnershipManagement/v4**, being {serverRoot} defines the certification endpoint

PartnershipSpecification and Partnership resource TEST CASES

Pre-requisites of nominal scenarios

The nominal scenarios assume that there are some organizations and Individuals that already exist in the system. The following three resources need to be created using the Party Management API.

```
{ “@type”:”Organization”, "name": "SuperdataCompany"}
{ “@type”:”Organization”, "name": "BestCloudCompany"}
{ “@type”:”Individual”, "name": "Doctor Hack"}
```

The nominal scenarios assume that there are some agreements specifications that already exist in the system. The following two resources need to be created using the Agreement Management API.

```
{ “@type”:”AgreementSpecification”, "name": " ContentLicenseAgreement "}
{ “@type”:” AgreementSpecification”, "name": " ProfitShareAgreement "}
```

Nominal Scenarios

TC_Partnership_N1 – Create new Partnership Specification with some Role specifications

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

```
{
  "description": "This  partnership specification …",
  "name": "Dream Partnership template",
  "roleSpecification": [
    [
      {
            "name": "ContentProvider",
            "agreementSpecification": [{
              "name": "ContentLicenseAgreement"
             }]
      },
      {
            "name" : "CloudProvider"
      },
      {
            "name" : "Developer",
            "agreementSpecification": [{
              "name": "ProfitShareAgreement"
```

```
                }]
            },
            {
                "name" : "Tester"
            }
        ]
    ]
}
```

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

TC_ Partnership_N2 – retrieve the created partnership specification

- Send a GET message to {apiRoot}/partnershipSpecification/{ID1} (ID1 comes from TC_ Partnership_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 Partnership Specification resource structure that matches the values in the original request

TC_ Partnership_N3 – create a partnership instance cthat complies with the partnership specification

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

```
{
  "@type": "Partnership",
  "description": "This  partnership ...",
  "name": "Dream Partnership",
  "specification" : {
            "href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047",
            "name": "Dream Partnership template"
  },
  "partner": [
    {
            "engagedParty" : {
              "@referredType": "Organization",
              "name": "SuperdataCompany",
               "role": "ContentProvider"
               }
    },
    {
            "engagedParty" : {
               "@referredType": "Organization",
               "name": "BestCloudCompany",
               "role" : "CloudProvider"
              }
    }
  ]
}
```

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

TC_ Partnership_N4 – retrieve the created partnership instance

- Send a GET message to {apiRoot}/partnership/{ID3} (ID3 comes from TC_ Partnership_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 Partnership resource structure that matches the values in the original request

TC_Partnership_N5 – Add new partner in the created partnership instance

- Send a PATCH message to {apiRoot}/partnership/{ID3} (ID3 comes from TC_ Partnership_N3)
  with the following header for json-merge:
  with the following contents in the BODY:

```
{
  "partner": [
    {
            "engagedParty" : {
              "@referredType": "Organization",
              "name": "SuperdataCompany",
               "role": "ContentProvider"
               }
    },
    {
            "engagedParty" : {
               "@referredType": "Organization",
               "name": "BestCloudCompany",
               "role" : "CloudProvider"
              }
   },
    {
            "engagedParty" : {
               "@referredType": "Individual",
               "name": "Doctor Hack",
               "role" : "Developer"
             }
    }
  ]
}
```

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

- Response Code 200-Updated
- The response message includes all mandatory parameters
- The body of the response matches the new values set in the original request

Error Scenarios

TC_Partnership _E1 – Unknown PartnershipSpecification identifier

- Send a GET message to /{apiRoot}/partnershipSpecification/{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_Partnership _E2 – Unknown Partnership identifier

- Send a GET message to /{apiRoot}/partnership/{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_Partnership_E3 – Invalid Request on Partnership Specification – Missing mandatory parameter

- Send a POST message to {apiRoot}/partnershipSpecification 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_ Partnership_E4 – Invalid Request on Partnership – Missing mandatory parameter

- Send a POST message to {apiRoot}/partnershipSpecification 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

Version History

Version Number	Date	Release led by:	Description
2.0	27-Jun-2018	Mariano Belaunde	First publication of the Partnership Type API as a standalone API in R18.0
4.0	18-July-2019	Mariano Belaunde mariano.belaunde@orange.com Pierre Gauthier TM Forum pgauthier@tmforum.org	Renaming and refactoring for R19.0 *Nota: the version 4.0 is assigned to all the APIs delivered in the release 19.0*

Release History

Release Number	Date	Release led by:	Description
Release 18.0.0	15-July-2018	Orange Team	Initial Release
Release 19.0.0	18-July-2019	Mariano Belaunde - Orange mariano.laurent@orange.com	Version 4.0 of the API