**Partnership Type Management API**

**Conformance Profile**

**Document Number TMF668B**

**March 2017**

|  |  |
| --- | --- |
| **Release: Frameworx Release 17.0** | **Status: Member Evaluation** |
| **Version: 1.1.0** | **IPR Mode: RAND** |

# NOTICE

Copyright © TM Forum 2017. 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](http://www.tmforum.org/IPRPolicy/11525/home.html), 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](#_Toc497381645)

[Table of Contents 3](#_Toc497381646)

[Introduction - API DESCRIPTION 4](#_Toc497381647)

[RESOURCE MODEL CONFORMANCE 5](#_Toc497381648)

[API MANDATORY AND OPTIONAL RESOURCES 5](#_Toc497381649)

[Partnership Type MANDATORY AND OPTIONAL ATTRIBUTES 5](#_Toc497381650)

[NOTIFICATION MODEL CONFORMANCE 6](#_Toc497381651)

[API MANDATORY AND OPTIONAL NOTIFICATIONS 6](#_Toc497381652)

[API OPERATIONS CONFORMANCE 7](#_Toc497381653)

[API MANDATORY AND OPTIONAL OPERATIONS 7](#_Toc497381654)

[API GET OPERATION CONFORMANCE 8](#_Toc497381655)

[/partnershipType?fields=...&{filtering} 8](#_Toc497381656)

[/partnershipType/{id}?fields=...&{filtering} 8](#_Toc497381657)

[API POST OPERATION CONFORMANCE 9](#_Toc497381658)

[/partnershipType 9](#_Toc497381659)

[API PATCH OPERATION CONFORMANCE 10](#_Toc497381660)

[/partnershipType/{id} 10](#_Toc497381661)

[API DELETE OPERATION CONFORMANCE 11](#_Toc497381662)

[/partnershipType/{id} 11](#_Toc497381663)

[API CONFORMANCE TEST SCENARIOS 12](#_Toc497381664)

[PartnershipType resource TEST CASES 12](#_Toc497381665)

[Release History 15](#_Toc497381666)

# Introduction - API DESCRIPTION

The Partnership Type API provides standardized mechanisms for creating partnership types in the context of an onboarding process.

The following resources are managed by this API:

* **PartnershipType**: Identifies a type of a partnership between parties, including the list of roles types that are permitted (i.e Buyer, Seller, Developper). Role types may refer to agreement specifications to be signed by parties playing the role.

The API allows the retrieval, creation, update and deletion of partnership type and its owned sub-resources.

# 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 |
| PartnershipType | M |  |

## Partnership Type MANDATORY AND OPTIONAL ATTRIBUTES

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

|  |  |  |
| --- | --- | --- |
| **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 |
| name | M (for resource creation) O (otherwise) |  |
| roleType | 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 the following table indicates which ones are mandatory and which ones are optional.

|  |  |  |
| --- | --- | --- |
| Notification Name | Mandatory or Optional | Comments |
| PartnershipTypeCreationNotification | M |  |
| PartnershipTypeRemoveNotification | 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

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 resources:  PartnershipType | POST must be used to create a new resource |
| PATCH | M for resources:  PartnershipType | PATCH must be used to partially update a resource |
| DELETE | M for resources:  PartnershipType | 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

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

|  |  |  |
| --- | --- | --- |
| GET | M |  |
| Response Status Code 200 | M |  |
| Other Status Codes | NA |  |

## /partnershipType?fields=...&{filtering}

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

## /partnershipType/{id}?fields=...&{filtering}

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

# API POST OPERATION CONFORMANCE

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

|  |  |  |
| --- | --- | --- |
| POST | M |  |
| Status Code 201 | M |  |
| Other Status Codes | NA |  |

## /partnershipType

This operation creates a partnership type entity.

**Mandatory and Non Mandatory Attributes**

The following tables provides the list of mandatory and non mandatory attributes when creating a PartnershipType, 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 |  |

|  |  |  |
| --- | --- | --- |
| Non Mandatory Attributes | Default Value | Rule |
| description |  |  |
| roleType |  |  |

**Additional Rules**

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

|  |  |
| --- | --- |
| Context | Mandatory Sub-Attributes |
| roleType | name |

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 (e.g.: status may be differently set by the server after processing than the one requested by the requestor)

# API PATCH OPERATION CONFORMANCE

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

|  |  |  |
| --- | --- | --- |
| PATCH | O |  |
| Status Code 201 | M |  |
| Other Status Codes | NA |  |

## /partnershipType/{id}

This operation allows partial updates of a partnership type 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 |
| name |  |
| description |  |
| roleType |  |

|  |  |
| --- | --- |
| Non Patchable Attributes | Rule |
| href |  |
| id |  |

# API DELETE OPERATION CONFORMANCE

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

|  |  |  |
| --- | --- | --- |
| DELETE | O |  |
| Status Code 200 | M |  |
| Other Status Codes | NA |  |

## /partnershipType/{id}

This operation deletes a partnership type 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}/partnershipType/v1**, where {serverRoot} defines the certification endpoint

## PartnershipType resource TEST CASES

**Nominal Scenarios**

**TC\_Trou\_N1 – Create new PartnershipType with minimum required information**

* Send a POST message to {apiRoot}/partnershipType/ with the following contents in the BODY

{

"name": "MyPartnership",

}

* 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/{IDtt1} where {IDtt1} indicates the identifier assigned by the server to the new partnershipType 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}/partnershipType/
* Wait for a response from the server with the following characteristics
  + Response Code 200-OK
  + The body of the response includes one PartnershipType 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}/partnershipType/{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 PartnershipType resource structure that matches the values in the original request

**TC\_Trou\_N2 – Search for PartnershipTypes with specific characteristics**

* Send a GET message to /{apiRoot}/partnershipType
* Wait for a response from the server with the following characteristics
  + Response Code 200-OK
  + The body of the response includes at least two partnershipType 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}/ partnershipType?roleType.name=developer
* Wait for a response from the server with the following characteristics
  + Response Code 200-OK
  + The body of the response includes one PartnershipType resource referring to {IDtt1} and there is no reference to PartnershipType 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\_N3 – Filtered retrieval of PartnershipTypes**

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

**TC\_Trou\_N4 – Filtered Search and Filtered data response**

* Send a GET message to /{apiRoot}/partnershipType?roleType.name=developer&fields=name
* Wait for a response from the server with the following characteristics
  + Response Code 200-OK
  + The body of the response includes one PartnershipType resource referring to {IDtt1} and there is no reference to PartnershipType resource {IDtt2}
  + 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 ”description” to filter the data included in the response but any other parameter could be used

**Error Scenarios**

**TC\_Trou\_E1 – Unknown Trouble PartnershipType identifier**

* Send a GET message to /{apiRoot}/partnershipType/{IDtt3}, where {IDtt3} does not match any of the identifiers previously creted 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}/partnershipType/ with the following contents in the BODY.

{

}

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

## Release History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release Number** | **Date** | **Release led by:** | **Description** |
| Release 1.0 | 07/15/2015 | Pierre Gauthier  TM Forum  pgauthier@tmforum.orf | First Release of Draft Version of the Document. |
| Release 1.1 | 03/15/2017 |  | Updated version including Test scenarios |