***Frameworx Specification***

**Agreement Management  
API REST Specification**

**TMF651**

**Release 16.0.1**

**October 2016**

|  |  |
| --- | --- |
| **Latest Update: Frameworx Release 16** | **TM Forum Approved** |
| **Version 1.0.2** | **IPR Mode: RAND** |

# NOTICE

Copyright © TM Forum 2016. 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.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website, but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,

East Tower – 10th Floor,

Morristown, NJ  07960 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](#_Toc453600945)

[Table of Contents 4](#_Toc453600946)

[List of Tables 6](#_Toc453600947)

[Introduction 7](#_Toc453600948)

[SAMPLE USE CASES 8](#_Toc453600949)

[RESOURCE MODEL 9](#_Toc453600950)

[Managed Entity and Task Resource Models 9](#_Toc453600951)

[Agreement resource 9](#_Toc453600952)

[Agreement Specification resource 14](#_Toc453600953)

[Notification Resource Models 19](#_Toc453600954)

[Agreement Creation Notification 20](#_Toc453600955)

[Agreement Attribute Value Change Notification 21](#_Toc453600956)

[Agreement State Change Notification 21](#_Toc453600957)

[Agreement Remove Notification 21](#_Toc453600958)

[API OPERATIONS 23](#_Toc453600959)

[Operations on Agreement 23](#_Toc453600960)

[List agreements 23](#_Toc453600961)

[Retrieve agreement 24](#_Toc453600962)

[Create agreement 26](#_Toc453600963)

[Patch agreement 30](#_Toc453600964)

[Delete agreement 31](#_Toc453600965)

[Operations on Agreement Specification 32](#_Toc453600966)

[List agreement specifications 32](#_Toc453600967)

[Retrieve agreement specification 33](#_Toc453600968)

[Create agreement specification 35](#_Toc453600969)

[Patch agreement specification 36](#_Toc453600970)

[Delete agreement specification 38](#_Toc453600971)

[API NOTIFICATIONS 39](#_Toc453600972)

[Register listener 39](#_Toc453600973)

[Unregister listener 40](#_Toc453600974)

[Publish Event to listener 40](#_Toc453600975)

[Acknowledgments 42](#_Toc453600976)

[Version History 42](#_Toc453600977)

[Release History 42](#_Toc453600978)

[Contributors to Document 42](#_Toc453600979)

# List of Tables

N/A

# Introduction

The Agreement API provides a 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.

# SAMPLE USE CASES

Reader will find example of use cases using Agreement API in “Open Digital Business Scenarios and Use Cases” document.

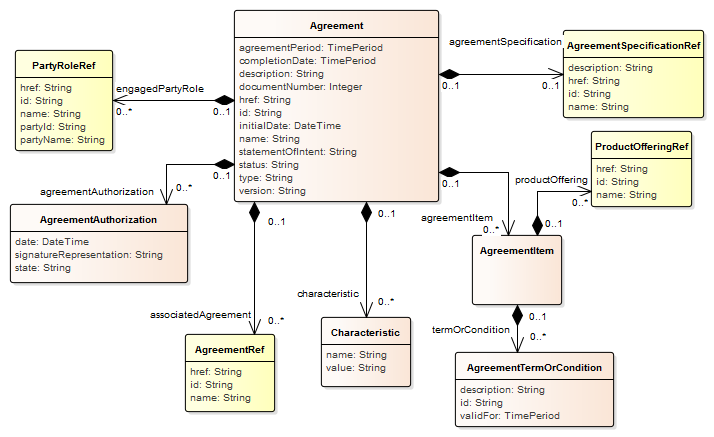
# RESOURCE MODEL

## Managed Entity and Task Resource Models

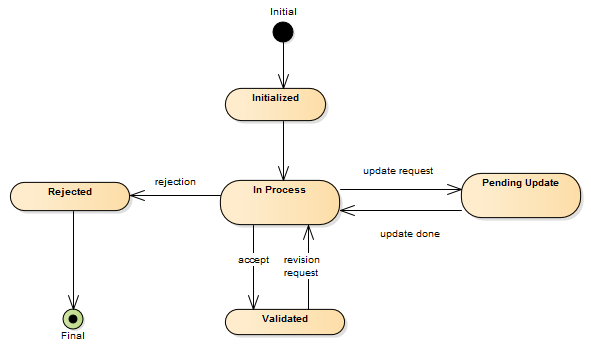
### Agreement resource

An agreement represents a contract or arrangement, either written or verbal and sometimes enforceable by law, such as a service level agreement or a customer price agreement. An agreement involves a number of other business entities, such as products, services, and resources and/or their specifications.

**Resource model**



**Lifecycle**

The Agreement lifecycle is tracked using the <i>status</i> field. The typical lifecycle values that can be taken are: Initialized, In process, Pending Update, validated and Rejected. The state machine specifying the typical state change transitions is provided below.  
  


**Field descriptions**

*Agreement* fields

|  |  |
| --- | --- |
| agreementPeriod | A time period. The time period during which the Agreement is in effect. |
| completionDate | A time period. Date at which the agreement is completed. |
| description | A string. Narrative that explains the agreement and details about it, such as why the agreement is taking place. |
| documentNumber | An integer. A reference number assigned to an Agreement that follows a prescribed numbering system. |
| href | A string. Unique url identifying the agreement as a resource. |
| id | A string. Unique identifier for the agreement. |
| initialDate | A date time (DateTime). Date at which the agreement was initialized. |
| name | A string. A human-readable name for the agreement. |
| statementOfIntent | A string. An overview and goals of the Agreement. |
| status | A string. The current status of the agreement. Typical values are: in process, approved and rejected. |
| type | A string. The type of the agreement. For example "commercial". |
| version | A string. A string identifying the version of the agreement. |
| agreementSpecification | An agreement specification reference (AgreementSpecificationRef). An AgreementSpecification represents a template of an agreement that can be used when establishing partnerships. |
| agreementItem | A list of agreement items (AgreementItem [\*]). A part of the agreement expressed in terms of a product offering and possibly including specific terms and conditions. |
| engagedPartyRole | A list of party role references (PartyRoleRef [\*]). A party role represents the part played by a party in a given context. |
| agreementAuthorization | A list of agreement authorizations (AgreementAuthorization [\*]). A business participant that is responsible for approving the agreement. |
| characteristic | A list of characteristics (Characteristic [\*]). Describes a given characteristic of an object or entity through a name/value pair. |
| associatedAgreement | A list of agreement references (AgreementRef [\*]). An agreement represents a contract or arrangement, either written or verbal and sometimes enforceable by law, such as a service level agreement or a customer price agreement. An agreement involves a number of other business entities, such as products, services, and resources and/or their specifications. |

*AgreementItem* sub-resource

A part of the agreement expressed in terms of a product offering and possibly including specific terms and conditions.

|  |  |
| --- | --- |
| productOffering | A list of product offering references (ProductOfferingRef [\*]). A product offering represents entities that are orderable from the provider of the catalog, this resource includes pricing information. |
| termOrCondition | A list of agreement term or conditions (AgreementTermOrCondition [\*]). Aspects of the agreement not formally specified elsewhere in the agreement and that cannot be captured elsewhere in a formal notation, or automatically monitored and require a more human level of management. |

*AgreementAuthorization* sub-resource

A business participant that is responsible for approving the agreement.

|  |  |
| --- | --- |
| date | A date time (DateTime). The date associated with the authorization state. |
| signatureRepresentation | A string. Indication that represents whether the signature is a physical paper signature or a digital signature. |
| state | A string. Current status of the authorization, for example in process, approved, rejected. |

*Characteristic* sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

|  |  |
| --- | --- |
| name | A string. Name of the characteristic. |
| value | A string. Value of the characteristic. |

*AgreementSpecificationRef* relationship

AgreementSpecification reference. An AgreementSpecification represents a template of an agreement that can be used when establishing partnerships.

|  |  |
| --- | --- |
| description | A string. A narrative that explains in detail what the agreement specification is about. |
| href | A string. Reference URL of the agreement specification. |
| id | A string. Unique identifier of the agreement specification. |
| name | A string. Name of the agreement specification. |

*PartyRoleRef* relationship

Party role reference. A party role represents the part played by a party in a given context.

|  |  |
| --- | --- |
| href | A string. Reference of the product. |
| id | A string. Unique identifier of the product. |
| name | A string. The name of the referred party role. |
| partyId | A string. The identifier of the engaged party that is linked to the PartyRole object. |
| partyName | A string. The name of the engaged party that is linked to the PartyRole object. |

*AgreementRef* relationship

Agreement reference. An agreement represents a contract or arrangement, either written or verbal and sometimes enforceable by law, such as a service level agreement or a customer price agreement. An agreement involves a number of other business entities, such as products, services, and resources and/or their specifications.

|  |  |
| --- | --- |
| href | A string. Reference of the agreement. |
| id | A string. Identifier of the agreement. |
| name | A string. Name of the agreement. |

**Json representation sample**

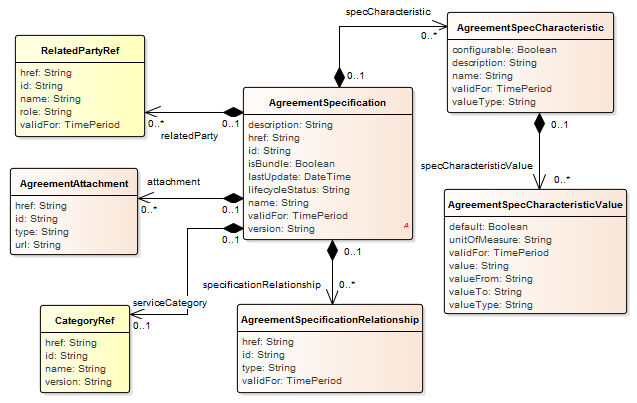
We provide below the json representation of an example of an 'Agreement' resource object

|  |
| --- |
| {  "agreementPeriod": {  "startDateTime": "2016-04-06T00:00",   "endDateTime": "2016-11-03T00:00"  },   "completionDate": "2016-10-16",   "description": "This agreement ...",   "documentNumber": 13,   "href": "https://host:port/onboardingManagement/agreement/9706",   "id": "9706",   "initialDate": "2015-10-16",   "name": "Summer Contract Agreement",   "statementOfIntent": "Agreement on minimum prices",   "status": "process",   "type": "commercial",   "version": "1.5",   "agreementSpecification": {  "description": "This agreement specification defines the ethic rules to be followed by each party.",   "href": "https://host:port/onboardingManagement/agreementSpecification/7399",   "id": "7399",   "name": "General Agreement Specification"  },   "agreementItem": [  {  "productOffering": [  {  "href": "https://host:port/productOffering/productOffering/4756",   "id": "4756",   "name": "Virtual Storage Medium",   "bundledProductOffering": [  {  "href": "https://host:port/productOffering/bundledProductOffering/1167",   "id": "1167",   "name": "Robust Offer",   "bundledProductOffering": []  }  ]  }  ],   "termOrCondition": [  {  "description": "This agreement term or condition ...",   "id": "5276",   "validFor": {  "startDateTime": "2016-04-09T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ]  }  ],   "engagedPartyRole": [  {  "href": "https://host:port/partyRole/partyRole/7770",   "id": "7770",   "name": "Supplier",   "partyId": "879",   "partyName": "Zero Bug LTD"  }  ],   "agreementAuthorization": [  {  "date": "2016-04-07T00:00",   "signatureRepresentation": "Mr Hyde",   "state": "rejected"  }  ],   "characteristic": [  [  {  "name": "country",   "value": "France"  },   {  "name": "confidentialLevel",   "value": "low"  }  ]  ],   "associatedAgreement": [  [  {  "name": "General Partnership Agreement",   "id": "98765453"  }  ]  ] } |

### Agreement Specification resource

A template of an agreement that can be used when establishing partnerships.

**Resource model**



**Field descriptions**

*AgreementSpecification* fields

|  |  |
| --- | --- |
| description | A string. A narrative that explains in detail what the agreement specification is about. |
| href | A string. Reference of the agreement specification. |
| id | A string. Unique identifier of the agreement specification. |
| isBundle | A boolean. Indicates that this agreement specification is a grouping of other agreement specifications. The list of bundled agreement specifications is provided via the specificationRelationship property. |
| lastUpdate | A date time (DateTime). Date and time of the last update. |
| lifecycleStatus | A string. Indicates the current lifecycle status. |
| name | A string. Name of the agreement specification. |
| validFor | A time period. The period for which the agreement specification is valid. |
| version | A string. Agreement specification version. |
| serviceCategory | A category reference (CategoryRef). The category resource is used to group product offerings, service and resource candidates in logical containers. Categories can contain other categories and/or product offerings, resource or service candidates. |
| specCharacteristic | A list of agreement spec characteristics (AgreementSpecCharacteristic [\*]). A characteristic quality or distinctive feature of an agreement. |
| relatedParty | A list of related party references (RelatedPartyRef [\*]). A related party defines party or party role linked to a specific entity. |
| attachment | A list of agreement attachments (AgreementAttachment [\*]). Represents a complementary piece of information to describe the agreement. Could be a document, picture, a video or any kind of multimedia content. |
| specificationRelationship | A list of agreement specification relationships (AgreementSpecificationRelationship [\*]). A relationship between agreement specifications. Typical relationships are substitution and dependency. |

*AgreementSpecCharacteristic* sub-resource

A characteristic quality or distinctive feature of an agreement.

|  |  |
| --- | --- |
| configurable | A boolean. If true, the Boolean indicates that the characteristic is configurable. |
| description | A string. A narrative that explains in detail what the characteristic is. |
| name | A string. Name of the characteristic being specified. |
| validFor | A time period. The period for which the specification characteristic is valid. |
| valueType | A string. A kind of value that the characteristic can take on, such as numeric, text and so forth. |
| specCharacteristicValue | A list of agreement spec characteristic values (AgreementSpecCharacteristicValue [\*]). A number or text that can be assigned to an agreement specification characteristic. |

*AgreementAttachment* sub-resource

Represents a complementary piece of information to describe the agreement. Could be a document, picture, a video or any kind of multimedia content.

|  |  |
| --- | --- |
| href | A string. Reference of the attachment. |
| id | A string. Unique identifier of the attachment. |
| type | A string. Attachment type such as video, picture. |
| url | A string. Uniform Resource Locator, is a web page address (a subset of URI). |

*AgreementSpecificationRelationship* sub-resource

A relationship between agreement specifications. Typical relationships are substitution and dependency.

|  |  |
| --- | --- |
| href | A string. Reference of an agreement specification. |
| id | A string. Unique identifier of the related agreement specification. |
| type | A string. Type of relationship such as, substitution or dependency. |
| validFor | A time period. The period for which the relationship is valid. |

*CategoryRef* relationship

Category reference. The category resource is used to group product offerings, service and resource candidates in logical containers. Categories can contain other categories and/or product offerings, resource or service candidates.

|  |  |
| --- | --- |
| href | A string. Unique reference of the category. |
| id | A string. Unique reference of the category. |
| name | A string. Name of the category. |
| version | A string. Category version. |

*RelatedPartyRef* relationship

RelatedParty reference. A related party defines party or party role linked to a specific entity.

|  |  |
| --- | --- |
| href | A string. Reference of the related party, could be a party reference or a party role reference. |
| id | A string. Unique identifier of a related party. |
| name | A string. Name of the related party. |
| role | A string. Role of the related party. |
| validFor | A time period. Validity period of the related party. |

**Json representation sample**

We provide below the json representation of an example of an 'AgreementSpecification' resource object

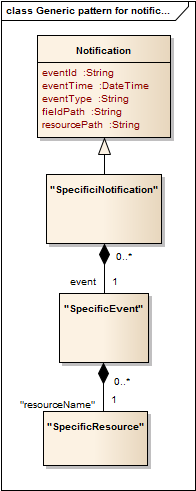
|  |
| --- |
| {  "description": "This agreement specification defines the ethic rules to be followed by each party.",   "href": "https://host:port/onboardingManagement/agreementSpecification/2376",   "id": "2376",   "isBundle": false,   "lastUpdate": "2016-04-07T00:00",   "lifecycleStatus": "initialized",   "name": "General Agreement Specification",   "validFor": {  "startDateTime": "2016-04-08T00:00",   "endDateTime": "2016-11-03T00:00"  },   "version": "3.0",   "serviceCategory": {  "href": "https://host:port/productOffering/category/7436",   "id": "7436",   "name": "Cloud",   "version": "3.2"  },   "specCharacteristic": [  {  "configurable": true,   "description": "This agreement spec characteristic ...",   "name": "Lights",   "validFor": {  "startDateTime": "2016-04-05T00:00",   "endDateTime": "2016-11-03T00:00"  },   "valueType": "integer",   "specCharacteristicValue": [  {  "default": true,   "unitOfMeasure": "MB",   "validFor": {  "startDateTime": "2016-04-10T00:00",   "endDateTime": "2016-11-03T00:00"  },   "value": "grey",   "valueFrom": "",   "valueTo": "",   "valueType": "string"  }  ]  }  ],   "relatedParty": [  {  "href": "https://host:port/partyManagement/organization/4043",   "id": "4043",   "name": "Richard Cole",   "role": "service provider",   "validFor": {  "startDateTime": "2016-04-07T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ],   "attachment": [  {  "href": "http://hostname:port/documentManagement/attachment/22",   "id": "22",   "type": "Document",   "url": "http://yyyyy"  }  ],   "specificationRelationship": [  {  "href": "https://host:port/agreements/agreementSpecificationRelationship/5661",   "id": "5661",   "type": "dependency",   "validFor": {  "startDateTime": "2016-04-05T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ] } |

## Notification Resource Models

4 notifications are defined for this API

Notifications related to Agreement:  
 - AgreementCreationNotification  
 - AgreementAttributeValueChangeNotification  
 - AgreementStateChangeNotification  
 - AgreementRemoveNotification

The notification structure for all notifications in this API follow the pattern depicted by the figure below.  
A notification resource (depicted by "SpecificNotification" placeholder) is a sub class of a generic Notification structure containing an id of the event occurence (eventId), an event timestamp (eventTime), and the name of the notification resource (eventType).   
This notification structure owns an event structure ("SpecificEvent" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



### Agreement Creation Notification

Notification sent when a new Agreement resource is created.

**Json representation sample**

We provide below the json representation of an example of an 'AgreementCreationNotification' notification object

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"AgreementCreationNotification",  "event": {  "agreement" :   {-- SEE Agreement RESOURCE SAMPLE --}  } } |

### Agreement Attribute Value Change Notification

Notification sent when changing an attribute of an Agreement resource.

**Json representation sample**

We provide below the json representation of an example of an 'AgreementAttributeValueChangeNotification' notification object

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"AgreementAttributeValueChangeNotification",  "event": {  "agreement" :   {-- SEE Agreement RESOURCE SAMPLE --}  } } |

### Agreement State Change Notification

Notification sent when changing the state of an Agreement resource.

**Json representation sample**

We provide below the json representation of an example of an 'AgreementStateChangeNotification' notification object

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"AgreementStateChangeNotification",  "event": {  "agreement" :   {-- SEE Agreement RESOURCE SAMPLE --}  } } |

### Agreement Remove Notification

Notification sent when removing an Agreement resource.

**Json representation sample**

We provide below the json representation of an example of an 'AgreementRemoveNotification' notification object

|  |
| --- |
| {  "eventId":"00001",  "eventTime":"2015-11-16T16:42:25-04:00",  "eventType":"AgreementRemoveNotification",  "event": {  "agreement" :   {-- SEE Agreement RESOURCE SAMPLE --}  } } |

# API OPERATIONS

Remember the following Uniform Contract:

|  |  |  |
| --- | --- | --- |
| Operation on Entities | Uniform API Operation | Description |
| Query Entities | GET Resource | GET must be used to retrieve a representation of a resource. |
| Create Entity | POST Resource | POST must be used to create a new resource |
| Partial Update of an Entity | PATCH Resource | PATCH must be used to partially update a resource |
| Complete Update of an Entity | PUT Resource | PUT must be used to completely update a resource identified by its resource URI |
| Remove an Entity | DELETE Resource | DELETE must be used to remove a resource |
| Execute an Action on an Entity | POST on TASK Resource | POST must be used to execute Task Resources |
| Other Request Methods | POST on TASK Resource | GET and POST must not be used to tunnel other request methods. |

Filtering and attribute selection rules are described in the TMF REST Design Guidelines.

Notifications are also described in a subsequent section.

## Operations on Agreement

### List agreements

**GET /agreement?fields=...&{filtering}**

**Description**

This operation list agreement entities.  
Attribute selection is enabled for all first level attributes.  
Filtering may be available depending on the compliance level supported by an implementation.

**Usage Samples**

Here's an example of a request for retrieving Agreement resources.

Retrieving all approved agreements of engaged party 'So Magic Ltd'. The result items are shrunk to show only the id and name(fields=id,name)

|  |
| --- |
| **Request** |
| GET /agreementManagement/agreement?fields=id,name&status=approved&engagedParty.name="So Magic Ltd" Accept: application/json |
| **Response** |
| 200  [  {  "id": "8756",   "name": "Employment Quota"  },   {  "id": "9435",   "name": "Zero Bug"  } ] |

### Retrieve agreement

**GET /agreement/{id}?fields=...&{filtering}**

**Description**

This operation retrieves an agreement entity.  
Attribute selection is enabled for all first level attributes.  
Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

**Usage Samples**

Here's an example of a request for retrieving an Agreement resource.

|  |
| --- |
| **Request** |
| GET /agreementManagement/agreement/9706 Accept: application/json |
| **Response** |
| 200  {  "agreementPeriod": {  "startDateTime": "2016-04-06T00:00",   "endDateTime": "2016-11-03T00:00"  },   "completionDate": "2016-10-16",   "description": "This agreement ...",   "documentNumber": 13,   "href": "https://host:port/onboardingManagement/agreement/9706",   "id": "9706",   "initialDate": "2015-10-16",   "name": "Summer Contract Agreement",   "statementOfIntent": "Agreement on minimum prices",   "status": "process",   "type": "commercial",   "version": "1.5",   "agreementSpecification": {  "description": "This agreement specification defines the ethic rules to be followed by each party.",   "href": "https://host:port/onboardingManagement/agreementSpecification/7399",   "id": "7399",   "name": "General Agreement Specification"  },   "agreementItem": [  {  "productOffering": [  {  "href": "https://host:port/productOffering/productOffering/4756",   "id": "4756",   "name": "Virtual Storage Medium",   "bundledProductOffering": [  {  "href": "https://host:port/productOffering/bundledProductOffering/1167",   "id": "1167",   "name": "Robust Offer",   "bundledProductOffering": []  }  ]  }  ],   "termOrCondition": [  {  "description": "This agreement term or condition ...",   "id": "5276",   "validFor": {  "startDateTime": "2016-04-09T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ]  }  ],   "engagedPartyRole": [  {  "href": "https://host:port/partyRole/partyRole/7770",   "id": "7770",   "name": "Supplier",   "partyId": "879",   "partyName": "Zero Bug LTD"  }  ],   "agreementAuthorization": [  {  "date": "2016-04-07T00:00",   "signatureRepresentation": "Mr Hyde",   "state": "rejected"  }  ],   "characteristic": [  [  {  "name": "country",   "value": "France"  },   {  "name": "confidentialLevel",   "value": "low"  }  ]  ],   "associatedAgreement": [  [  {  "name": "General Partnership Agreement",   "id": "98765453"  }  ]  ] } |

### Create agreement

**POST /agreement**

*Note: this operation is available only to ADMIN API users*

**Description**

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.

|  |  |
| --- | --- |
| Mandatory Attributes | Rule |
| name |  |
| type |  |
| engagedPartyRole |  |
| agreementItem |  |

|  |  |  |
| --- | --- | --- |
| Non Mandatory Attributes | Default Value | Rule |
| agreementPeriod |  |  |
| completionDate | Current date |  |
| description |  |  |
| 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 |

**Usage Samples**

Here's an example of a request for creating an Agreement resource. In this example the request only passes mandatory attributes.

|  |
| --- |
| **Request** |
| POST /agreementManagement/agreement Content-Type: application/json  {  "name": "Summer Contract Agreement",   "type": "commercial",   "agreementItem": [  {  "productOffering": [  {  "href": "https://host:port/productOffering/productOffering/4756",   "id": "4756",   "name": "Virtual Storage Medium",   "bundledProductOffering": [  {  "href": "https://host:port/productOffering/bundledProductOffering/1167",   "id": "1167",   "name": "Robust Offer",   "bundledProductOffering": []  }  ]  }  ],   "termOrCondition": [  {  "description": "This agreement term or condition ...",   "id": "5276",   "validFor": {  "startDateTime": "2016-04-09T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ]  }  ],   "engagedPartyRole": [  {  "href": "https://host:port/partyRole/partyRole/7770",   "id": "7770",   "name": "Supplier",   "partyId": "879",   "partyName": "Zero Bug LTD"  }  ] } |
| **Response** |
| 201  {  "completionDate": "2016-10-16",   "href": "https://host:port/onboardingManagement/agreement/9706",   "id": "9706",   "name": "Summer Contract Agreement",   "type": "commercial",   "version": "0",   "agreementItem": [  {  "productOffering": [  {  "href": "https://host:port/productOffering/productOffering/4756",   "id": "4756",   "name": "Virtual Storage Medium",   "bundledProductOffering": [  {  "href": "https://host:port/productOffering/bundledProductOffering/1167",   "id": "1167",   "name": "Robust Offer",   "bundledProductOffering": []  }  ]  }  ],   "termOrCondition": [  {  "description": "This agreement term or condition ...",   "id": "5276",   "validFor": {  "startDateTime": "2016-04-09T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ]  }  ],   "engagedPartyRole": [  {  "href": "https://host:port/partyRole/partyRole/7770",   "id": "7770",   "name": "Supplier",   "partyId": "879",   "partyName": "Zero Bug LTD"  }  ] } |

### Patch agreement

**PATCH /agreement/{id}**

**Description**

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 |  |

**Usage Samples**

Here's an example of a request for patching an Agreement resource.

Changing the status of the agreement as rejected (using json-patch)

|  |
| --- |
| **Request** |
| PATCH /agreementManagement/agreement/42 Content-Type: application/json-patch+json  {  "path": "/status",   "value": "rejected",   "op": "replace" } |
| **Response** |
| 201  { Similar JSON as in GET response with bank account domiciliation changed } |

### Delete agreement

**DELETE /agreement/{id}**

*Note: this operation is available only to ADMIN API users*

**Description**

This operation deletes an agreement entity.

**Usage Samples**

Here's an example of a request for deleting an Agreement resource.

|  |
| --- |
| **Request** |
| DELETE /agreementManagement/agreement/42 |
| **Response** |
| 204 |

## Operations on Agreement Specification

### List agreement specifications

**GET /agreementSpecification?fields=...&{filtering}**

**Description**

This operation list agreement specification entities.  
Attribute selection is enabled for all first level attributes.  
Filtering may be available depending on the compliance level supported by an implementation.

**Usage Samples**

Here's an example of a request for retrieving AgreementSpecification resources.

Retrieving all agreement specifications with the government army. The result items are shrunk to show only the id and name (fields=id,name)

|  |
| --- |
| **Request** |
| GET /agreementManagement/agreementSpecification?fields=id,name&relatedParty.name="Army" Accept: application/json |
| **Response** |
| 200  [  {  "id": "5434",   "name": "General Maintenance"  },   {  "id": "9080",   "name": "Secret"  } ] |

### Retrieve agreement specification

**GET /agreementSpecification/{id}?fields=...&{filtering}**

**Description**

This operation retrieves an agreement specification entity.  
Attribute selection is enabled for all first level attributes.  
Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

**Usage Samples**

Here's an example of a request for retrieving an AgreementSpecification resource.

|  |
| --- |
| **Request** |
| GET /agreementManagement/agreementSpecification/2376 Accept: application/json |
| **Response** |
| 200  {  "description": "This agreement specification defines the ethic rules to be followed by each party.",   "href": "https://host:port/onboardingManagement/agreementSpecification/2376",   "id": "2376",   "isBundle": false,   "lastUpdate": "2016-04-07T00:00",   "lifecycleStatus": "initialized",   "name": "General Agreement Specification",   "validFor": {  "startDateTime": "2016-04-08T00:00",   "endDateTime": "2016-11-03T00:00"  },   "version": "3.0",   "serviceCategory": {  "href": "https://host:port/productOffering/category/7436",   "id": "7436",   "name": "Cloud",   "version": "3.2"  },   "specCharacteristic": [  {  "configurable": true,   "description": "This agreement spec characteristic ...",   "name": "Lights",   "validFor": {  "startDateTime": "2016-04-05T00:00",   "endDateTime": "2016-11-03T00:00"  },   "valueType": "integer",   "specCharacteristicValue": [  {  "default": true,   "unitOfMeasure": "MB",   "validFor": {  "startDateTime": "2016-04-10T00:00",   "endDateTime": "2016-11-03T00:00"  },   "value": "grey",   "valueFrom": "",   "valueTo": "",   "valueType": "string"  }  ]  }  ],   "relatedParty": [  {  "href": "https://host:port/partyManagement/organization/4043",   "id": "4043",   "name": "Richard Cole",   "role": "service provider",   "validFor": {  "startDateTime": "2016-04-07T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ],   "attachment": [  {  "href": "http://hostname:port/documentManagement/attachment/22",   "id": "22",   "type": "Document",   "url": "http://yyyyy"  }  ],   "specificationRelationship": [  {  "href": "https://host:port/agreements/agreementSpecificationRelationship/5661",   "id": "5661",   "type": "dependency",   "validFor": {  "startDateTime": "2016-04-05T00:00",   "endDateTime": "2016-11-03T00:00"  }  }  ] } |

### Create agreement specification

**POST /agreementSpecification**

*Note: this operation is available only to ADMIN API users*

**Description**

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.

|  |  |
| --- | --- |
| 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 |

**Usage Samples**

Here's an example of a request for creating an AgreementSpecification resource. In this example the request only passes mandatory attributes.

|  |
| --- |
| **Request** |
| POST /agreementManagement/agreementSpecification Content-Type: application/json  {  "name": "General Agreement Specification",   "attachment": [  {  "href": "http://hostname:port/documentManagement/attachment/22",   "id": "22",   "type": "Document",   "url": "http://yyyyy"  }  ] } |
| **Response** |
| 201  {  "href": "https://host:port/onboardingManagement/agreementSpecification/2376",   "id": "2376",   "isBundle": false,   "name": "General Agreement Specification",   "attachment": [  {  "href": "http://hostname:port/documentManagement/attachment/22",   "id": "22",   "type": "Document",   "url": "http://yyyyy"  }  ] } |

### Patch agreement specification

**PATCH /agreementSpecification/{id}**

*Note: this operation is available only to ADMIN API users*

**Description**

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 |  |

**Usage Samples**

Here's an example of a request for patching an AgreementSpecification resource.

Adding an attachment document to the agreement specification (using json-patch)

|  |
| --- |
| **Request** |
| PATCH /agreementManagement/agreementSpecification/42 Content-Type: application/json-patch+json  {  "path": "/attachment",   "value": {  "url": "http://www.allmydocs.fr/agreement\_standard.pdf",   "type": "Document"  },   "op": "add" } |
| **Response** |
| 201  { Similar JSON as in GET response with an attachment added } |

### Delete agreement specification

**DELETE /agreementSpecification/{id}**

*Note: this operation is available only to ADMIN API users*

**Description**

This operation deletes an agreement specification entity.

**Usage Samples**

Here's an example of a request for deleting an AgreementSpecification resource.

|  |
| --- |
| **Request** |
| DELETE /agreementManagement/agreementSpecification/42 |
| **Response** |
| 204 |

# API NOTIFICATIONS

For every single of operation on the entities use the following templates and provide sample REST notification POST calls.

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines reproduced below.

## Register listener

**POST /hub**

**Description**

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

**Behavior**

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

**Usage Samples**

Here's an example of a request for registering a listener.

|  |
| --- |
| **Request** |
| POST /api/hub  Accept: application/json  {"callback": "http://in.listener.com"} |
| **Response** |
| 201  Content-Type: application/json  Location: /api/hub/42  {"id":"42","callback":"http://in.listener.com","query":null} |

## Unregister listener

**DELETE /hub/{id}**

**Description**

Clears the communication endpoint address that was set by creating the Hub.

**Behavior**

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

**Usage Samples**

Here's an example of a request for un-registering a listener.

|  |
| --- |
| **Request** |
| DELETE /api/hub/42  Accept: application/json |
| **Response** |
| 204 |

## Publish Event to listener

**POST /client/listener**

**Description**

Clears the communication endpoint address that was set by creating the Hub.

Provides to a registered listener the description of the event that was raised. The /client/listener url is the callback url passed when registering the listener.

**Behavior**

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

**Usage Samples**

Here's an example of a notification received by the listener. In this example “EVENT TYPE” should be replaced by one of the notification types supported by this API (see Notification resources Models section) and EVENT BODY refers to the data structure of the given notification type.

|  |
| --- |
| **Request** |
| POST /client/listener  Accept: application/json  {  "event": {  EVENT BODY  },  "eventType": "EVENT\_TYPE"  } |
| **Response** |
| 201 |

For detailed examples on the general TM Forum notification mechanism, see the TMF REST Design Guidelines.

# Acknowledgments

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Modified by** | **Description** |
| Version 1.0.0 | 15/04/2016 | Pierre Gauthier  TM Forum  [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Mariano Belaunde Orange [mariano.belaunde@orange.com](mailto:mariano.belaunde@orange.com) | Final version |
| Version 1.0.1 | 13/06/2016 | Alicja Kawecki  TM Forum | Updated cover; minor formatting/style corrections prior to publishing for Fx16 |
| Version 1.0.2 | 05/10/2016 | Alicja Kawecki  TM Forum | Updated cover and Notice to reflect TM Forum Approved status |

## Release History

|  |  |  |  |
| --- | --- | --- | --- |
| **Release Number** | **Date** | **Release led by:** | **Description** |
| Release 1.0 | 15/04/2016 | Pierre Gauthier TM Forum [pgauthier@tmforum.org](mailto:pgauthier@tmforum.org)  Mariano Belaunde Orange [mariano.belaunde@orange.com](mailto:mariano.belaunde@orange.com) | First Release of the Document. Generated from the API Data Model. |

## Contributors to Document

|  |  |
| --- | --- |
| Veronique Mauneau | Orange |
| Jean-Luc Tymen | Orange |
| Mariano Belaunde | Orange |
| Elaine Haher | Ericsson |
| August-Wilhelm Jagau | Ericsson |
| Liuyiling (Sammy) | Huawei |
| Sunruinan | Huawei |
| Jiang Yisong | Huawei |
| George Glass | BT |
| Pierre Gauthier | TM Forum |
| Andreas Polz | Infonova |
| Takayuki Nakamura | NTT |