



Service Catalog Management API REST Specification

Document Number: TMF633
August 2017

Latest Update: Release 17.5	Status: Member Evaluation
Version 2.0.0	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](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

4 Century Drive
Suite 100
Parsippany, NJ 07054
USA

Tel No. +1 973 944 5100

Fax No. +1 973 944 5110

TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE	2
Table of Contents.....	3
Introduction.....	6
SAMPLE USE CASES	7
Lifecycle Management Use Case	7
RESOURCE MODEL	9
Managed Entity and Task Resource Models	9
Service Catalog resource	11
Service Category resource	13
Service Candidate resource	17
Service Specification resource.....	20
Import Job resource.....	29
Export Job resource	31
Notification Resource Models	33
Service Catalog Creation Notification	34
Service Catalog Remove Notification.....	34
Service Catalog Batch Notification.....	35
Service Category Creation Notification	35
Service Category Remove Notification	36
Service Candidate Creation Notification	36
Service Candidate Remove Notification.....	36
Service Specification Creation Notification.....	37
Service Specification Remove Notification.....	37
API OPERATIONS.....	39
Operations on Service Catalog.....	40
List service catalogs	40

Retrieve service catalog	41
Create service catalog	42
Patch service catalog	44
Delete service catalog	45
Operations on Service Category	47
List service categories	47
Retrieve service category	48
Create service category	50
Patch service category	51
Delete service category	53
Operations on Service Candidate	54
List service candidates	54
Retrieve service candidate	61
Create service candidate	62
Patch service candidate	63
Delete service candidate	65
Operations on Service Specification	67
List service specifications	67
Retrieve service specification	69
Create service specification	72
Patch service specification	74
Delete service specification	77
Operations on Import Job	78
List import jobs	78
Retrieve import job	78
Create import job	79
Delete import job	80

Operations on Export Job	82
List export jobs	82
Retrieve export job	82
Create export job	83
Delete export job	84
API NOTIFICATIONS.....	86
Register listener	86
Unregister listener	87
Publish Event to listener	87
Lifecycle Management Extensions to Catalog.....	89
Query all versioned catalog resources.....	89
Query a specific versioned catalog resource	91
Query current version of a catalog resource	92
Create new version of a catalog resource.....	92
Modify an existing version of a catalog resource	93
Role based Access Control	95
Acknowledgements.....	96
Release History	96
Contributors to Document.....	96

INTRODUCTION

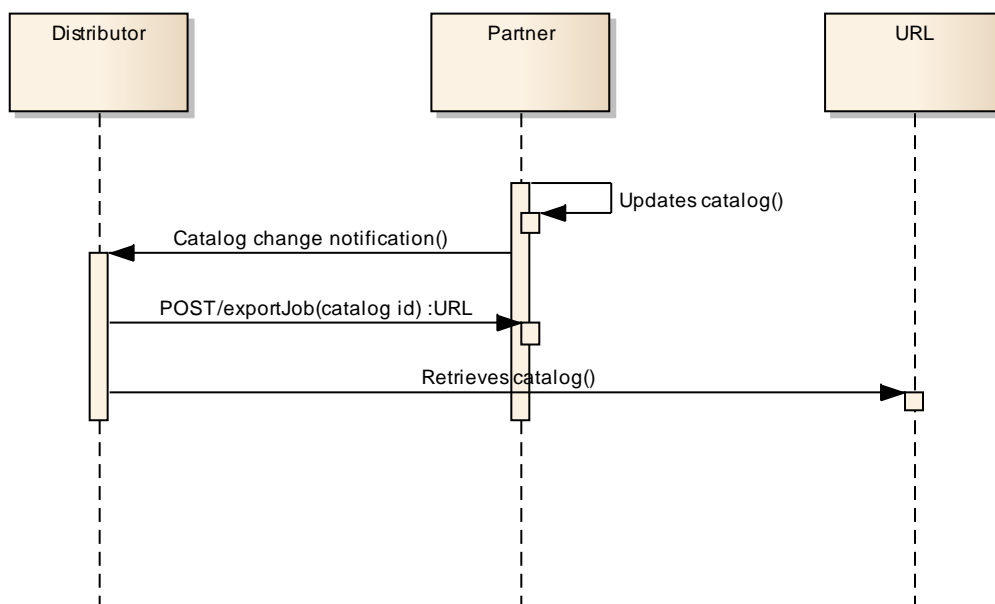
The Service Catalog Management API REST specification allows the management of the entire lifecycle of the Service Catalog elements and the consultation of service catalog elements during several processes such as ordering process.

SAMPLE USE CASES

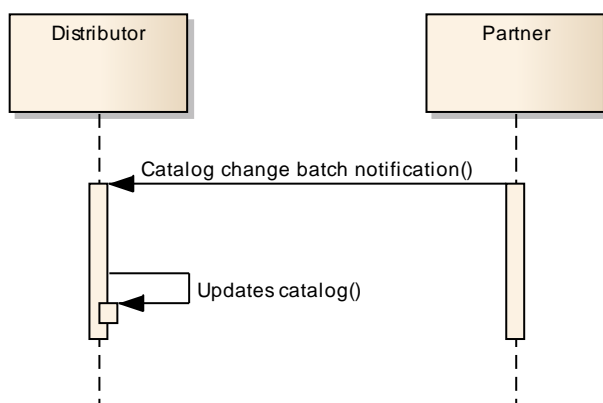
Lifecycle Management Use Case

The Service Catalog Management API REST Specification allows the management of the entire lifecycle of the service catalog elements. The followings are use case examples of a service catalog management. Please refer to Frameworkx guidebook GB978 for more information.

UC1: A partner updates his catalog. He notifies his distributor the catalog change. The distributor requests a catalog export. Then, he retrieves the catalog at the provided URL.



UC2: A partner updates his catalog. He notifies all catalog changes in detail to his distributor. This one updates his catalog copy.



UC3: A catalog administrator wants to retrieve effective duration of a resource candidate based on resource candidate identifier or other search criteria (GET /serviceCandidate/{ID}).

UC4: A catalog administrator wants to retrieve all service candidates in service catalog (GET /serviceCandidate).

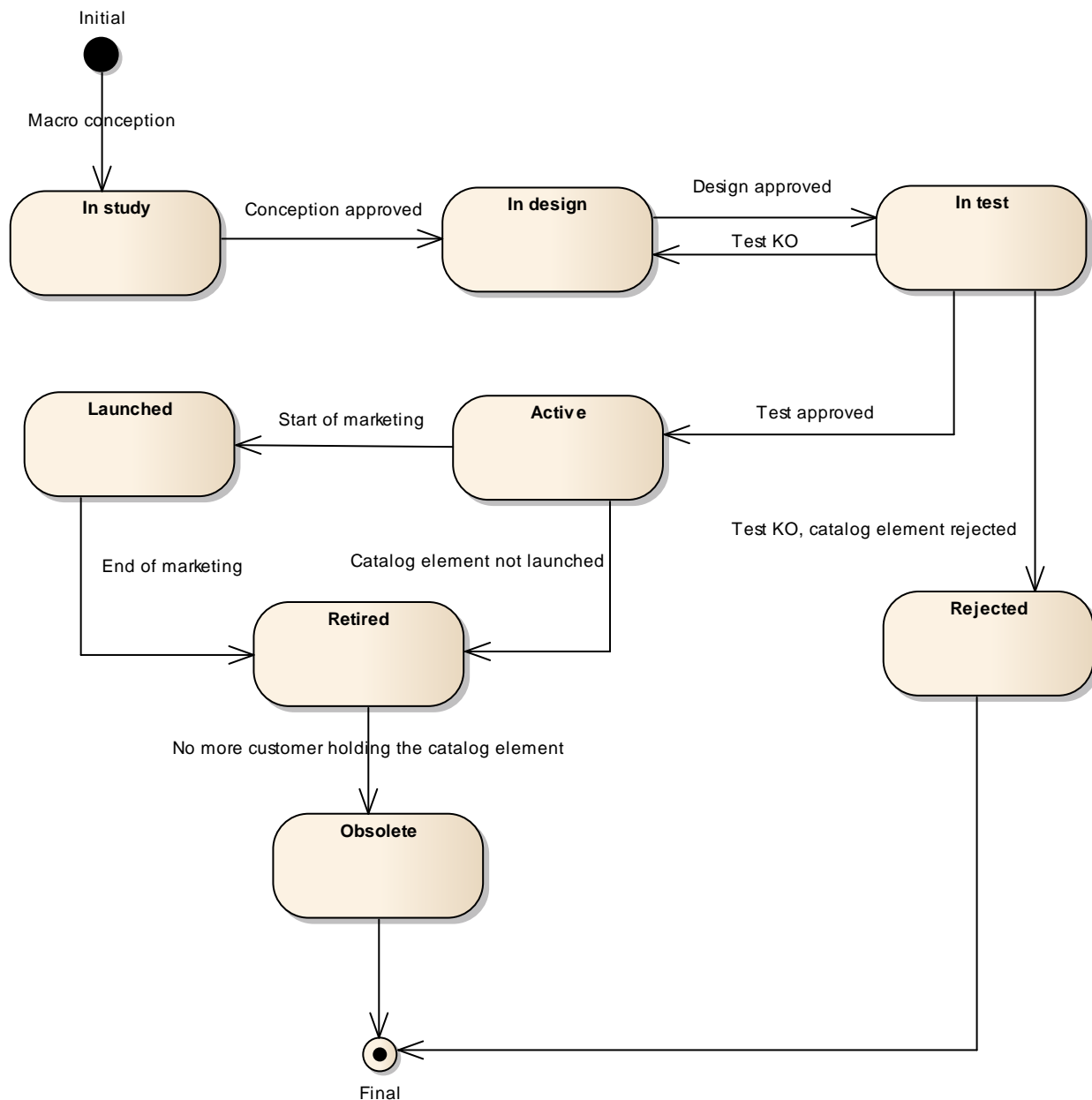
UC5: A catalog administrator wants to update the lifecycle status (from Launched to retired for example) of a service candidate (PATCH /serviceCandidate/{ID}).

RESOURCE MODEL

Managed Entity and Task Resource Models

Resource Lifecycle Management:

Resource Lifecycle Management is responsible for managing the entire lifecycle of the service catalog element and its underlying components. This include all the processes required to design, build, deploy, maintain and ultimately retire the catalog element.



When the macro conception of a catalog element is started the first status of the later is “In Study”.

When the conception of the catalog element is accepted, its status is changed to “In Design”.

If the design is approved its status is changed to “In Test”.

Then either the test is OK and then its status is changed to “Active” or the test is KO (failed) and its status is changed to “Rejected”. The Rejected status is a final status.

When a catalog element is in a “Active” status it means, it has been validated and tested, but it is still not available for customers.

When the beginning of marketing is reached, its status is changed to “Launched”. At this moment, customers can buy it.

If the catalog element is not launched, its status is changed to “Retired”.

The same status is achieved when a catalog element reaches the end of marketing.

The “Retired” status means it cannot be sold to any new customers, but previous customers can still have it.

When no more customer holds the catalog element, its status is changed to “Obsolete” meaning it can be removed from the catalog.

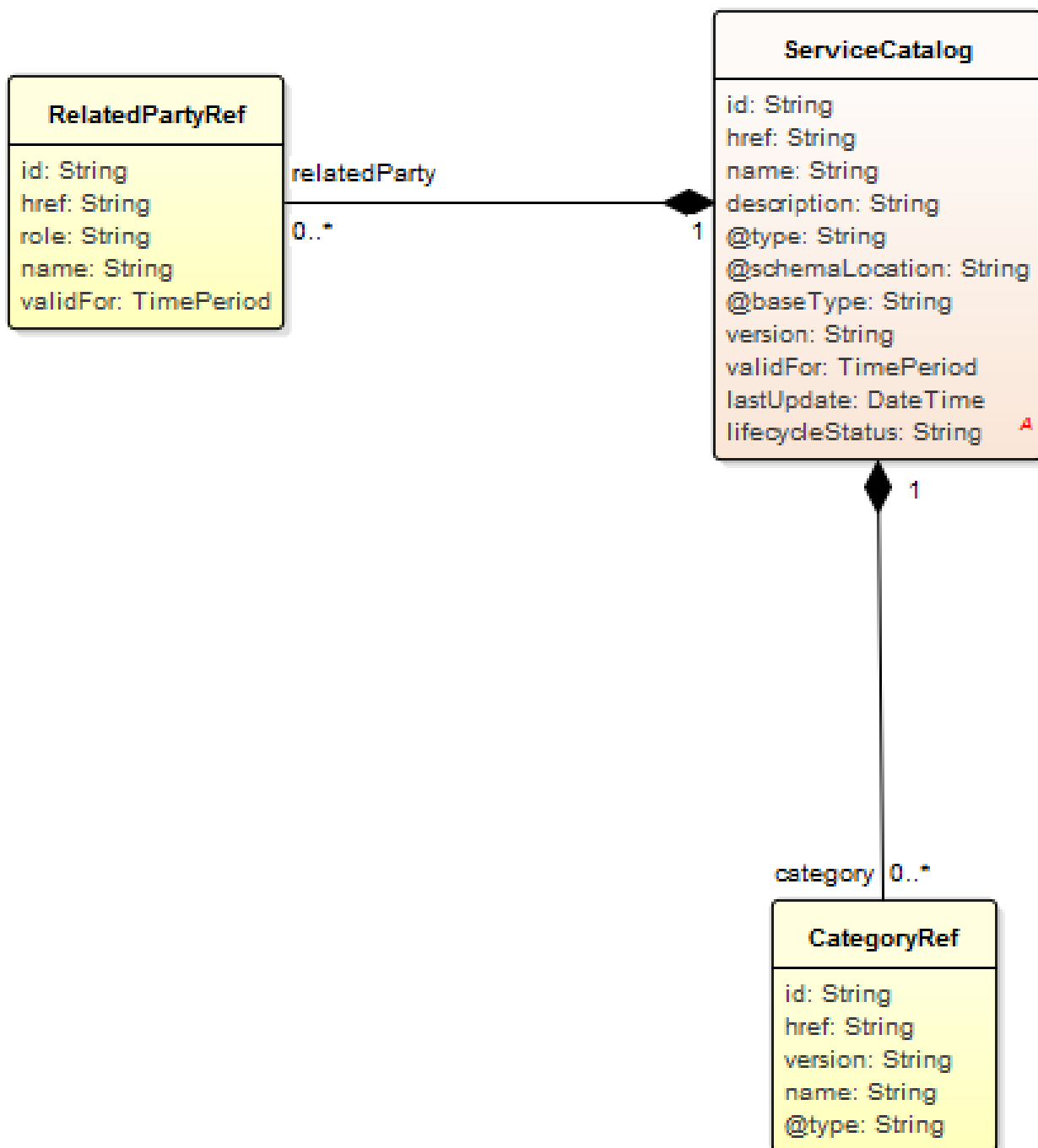
SERVICE CATALOG RESOURCE

The root entity for service catalog management.

A service catalog is a group of service specifications made available through service candidates that an organization provides to the consumers (internal consumers like its employees or B2B customers or B2C customers).

A service catalog typically includes name, description and time period that is valid for.

Resource model



Field descriptions

ServiceCatalog fields

id	A string. Unique identifier of the Catalog.
href	A string. Unique reference of the catalog.
name	A string. Name of the catalog.
description	A string. Description of this catalog.
@type	A string. Indicates the (class) type of catalog. For service catalogs, this will be 'ServiceCatalog'.
@schemaLocation	A string. This field provides a link to the schema describing this REST resource.
@baseType	A string. Indicates the base (class) type of this REST resource.
version	A string. Catalog version.
validFor	A time period. The period for which the catalog is valid.
lastUpdate	A date time (DateTime). Date and time of the last update.
lifecycleStatus	A string. Used to indicate the current lifecycle status.

Json representation sample

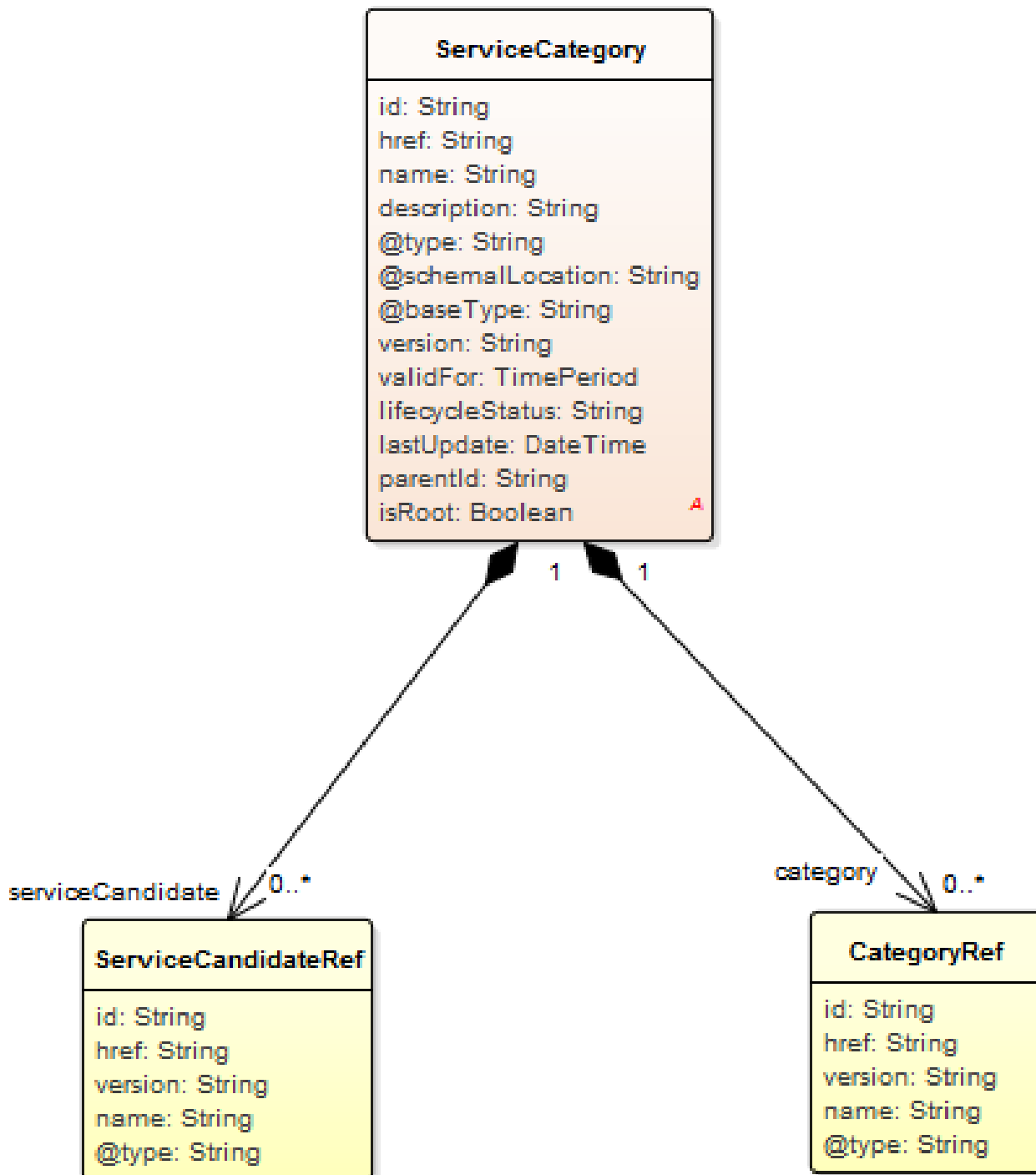
We provide below the json representation of an example of a 'ServiceCatalog' resource object

```
{
  "id": "3830",
  "href": "https://host:port/catalogManagement/serviceCatalog/3830",
  "name": "Catalog Wholesale Business",
  "description": "This service catalog ...",
  "@type": "ServiceCatalog",
  "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCatalog.yml ",
  "@baseType": "Catalog",
  "version": "1.0",
  "validFor": {
    "startDateTime": "2017-08-29T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lastUpdate": "2017-08-27T00:00",
  "lifecycleStatus": "Active"
}
```

SERVICE CATEGORY RESOURCE

The (service) category resource is used to group service candidates in logical containers. Categories can contain other categories.

Resource model



Field descriptions

ServiceCategory fields

id	A string. Unique identifier of the category.
href	A string. Hyperlink reference to the category.
name	A string. Name of the category.
description	A string. Description of the category.
@type	A string. The (class) type of this category.
@schemaLocation	A string. This field provides a link to the schema describing this REST resource.
@baseType	A string. Immediate base class type of this category.
version	A string. Category version.
validFor	A time period. The period for which the category is valid.
lifecycleStatus	A string. Used to indicate the current lifecycle status.
lastUpdate	A date time (DateTime). Date and time of the last update.
parentId	A string. Unique identifier of the parent category.
isRoot	A boolean. If true, this Boolean indicates that the category is a root of categories.
relatedParty	A list of related party references (RelatedPartyRef [*]). A related party defines party or party role linked to a specific entity.
serviceCandidate	A list of service candidate references (ServiceCandidateRef [*]). reference to ServiceCandidate object.
category	A list of category references (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.

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.

id	A string. Unique reference of the category.
href	A string. Unique reference of the category.

version A string. Category version.
 name A string. Name of the category.

RelatedPartyRef relationship

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

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

ServiceCandidateRef relationship

reference to ServiceCandidate object.

id A string. reference id to the target ServiceCandidate.
 href A string. Hyperlink reference to the target ServiceCandidate.
 version A string. ServiceCandidate version.
 name A string. Name given to the ServiceCandidate.
 @type A string. The (class) type of the ServiceCandidate.

Json representation sample

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

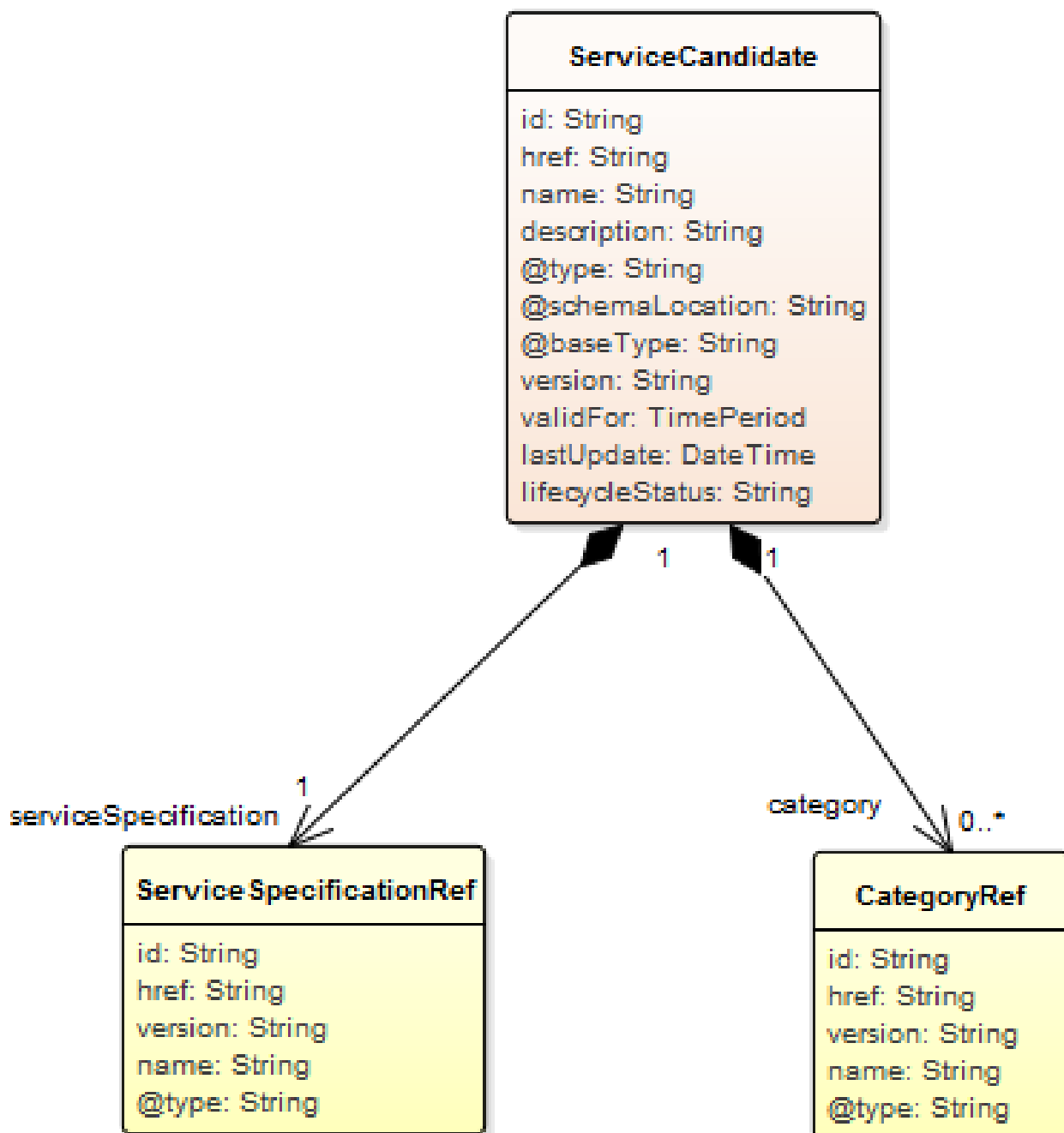
```
{
  "id": "1708",
  "href": "https://host:port/catalogManagement/serviceCategory/1708",
  "name": "Cloud Services",
  "description": "A category to hold all available cloud service offers",
  "@type": "ServiceCategory",
  "@schemalocation": "https://host:port/catalogManagement/schema/ServiceCategory.yml",
  "@baseType": "Category",
  "version": "1.0",
  "validFor": {
    "startDateTime": "2017-08-24T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lifecycleStatus": "Active",
  "lastUpdate": "2017-08-27T00:00",
  "parentId": "589",
  "isRoot": false,
```

```
"relatedParty": [  
  {  
    "id": "9555",  
    "href": "https://host:port/partyManagement/organization/9555",  
    "role": "seller",  
    "name": "John Doe",  
    "validFor": {  
      "startDateTime": "2017-08-24T00:00",  
      "endDateTime": "2018-03-25T00:00"  
    }  
  }  
],  
"serviceCandidate": [  
  {  
    "id": "5850",  
    "href": "https://host:port/catalogManagement/serviceCandidate/5850",  
    "version": "1.1",  
    "name": "Speed Max",  
    "@type": "ServiceCandidate"  
  }  
],  
"category": [  
  {  
    "id": "6086",  
    "href": "https://host:port/catalogManagement/category/6086",  
    "version": "1.5",  
    "name": "Cloud"  
  }  
]  
}
```


SERVICE CANDIDATE RESOURCE

ServiceCandidate is an entity that makes a service specification available to a catalog. A ServiceCandidate and its associated service specification may be published - made visible - in any number of service catalogs, or in none. One service specification can be composed of other service specifications.

Resource model



Field descriptions

ServiceCandidate fields

id	A string. Unique identifier of this REST resource.
href	A string. Hyperlink reference to this REST resource.
name	A string. Name given to this REST resource.
description	A string. Description of this REST resource.
@type	A string. Class type of this REST resource.
@schemaLocation	A string. This field provides a link to the schema describing this REST resource.
@baseType	A string. The (immediate) base class type of this REST resource.
version	A string. the version of service candidate.
validFor	A time period. The period for which this REST resource is valid.
lastUpdate	A date time (DateTime). Date and time of the last update of this REST resource.
lifecycleStatus	A string. Used to indicate the current lifecycle status of the service candidate.
category	A list of category references (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.
serviceSpecification	A service specification reference (ServiceSpecificationRef). reference to ServiceSpecification object.

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.

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

ServiceSpecificationRef relationship

reference to ServiceSpecification object.

id	A string. reference id to the target ServiceSpecification.
href	A string. Hyperlink reference to the target ServiceSpecification.
version	A string. ServiceSpecification version.
name	A string. Name given to the ServiceSpecification.
@type	A string. The (class) type of the ServiceSpecification.

Json representation sample

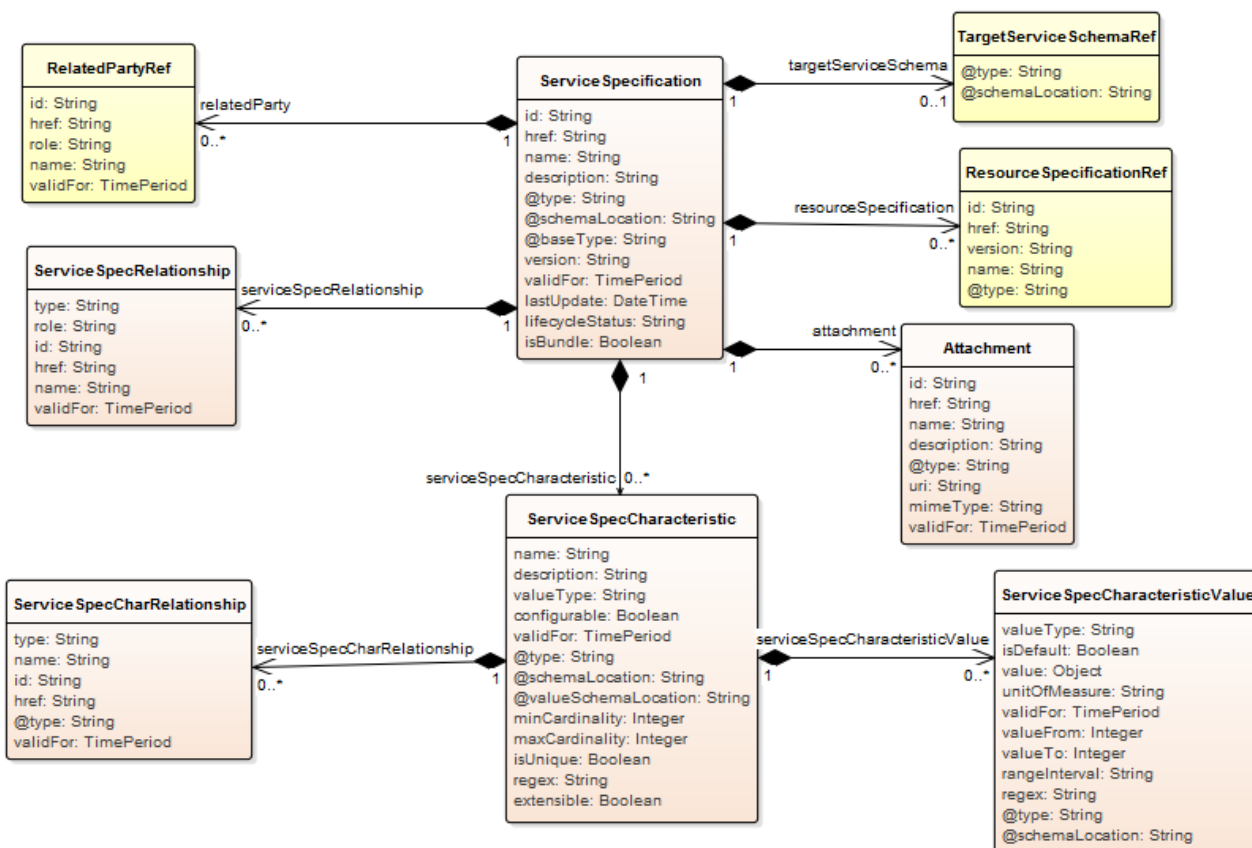
We provide below the json representation of an example of a 'ServiceCandidate' resource object

```
{
  "id": "4994",
  "href": "https://host:port/catalogManagement/serviceCandidate/4994",
  "name": "TVServiceCandidate",
  "description": "This service candidate ...",
  "@type": "ServiceCandidate",
  "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCandidate.yml ",
  "@baseType": "",
  "version": "2.1",
  "validFor": {
    "startDateTime": "2017-08-23T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lastUpdate": "2017-08-27T00:00",
  "lifecycleStatus": "Active",
  "category": [
    {
      "id": "5980",
      "href": "https://host:port/catalogManagement/category/5980",
      "version": "3.2",
      "name": "TV"
    }
  ],
  "serviceSpecification": {
    "id": "9600",
    "href": "https://host:port/catalogManagement/serviceSpecification/9600",
    "version": "2.1",
    "name": "CFSS_TV",
    "@type": "CustomerFacingServiceSpecification"
  }
}
```

SERVICE SPECIFICATION RESOURCE

ServiceSpecification is a class that offers characteristics to describe a type of service. Functionally, it acts as a template by which Services may be instantiated. By sharing the same specification, these services would therefore share the same set of characteristics. CustomerFacing ServiceSpecification(CFSS) and ResourceFacingServiceSpecification (RFSS) are two well-known extension of the ServiceSpecification class.

Resource model



Field descriptions

ServiceSpecification fields

id	A string. Unique identifier of this REST resource.
href	A string. Hyperlink reference to this REST resource.
name	A string. Name given to this REST resource.
description	A string. Description of this REST resource.
@type	A string. Class type of this REST resource.

@schemaLocation	A string. This field provides a link to the schema describing this REST resource.
@baseType	A string. The (immediate) base class type of this REST resource.
version	A string. Service specification version.
validFor	A time period. The period for which this REST resource is valid.
lastUpdate	A date time (DateTime). Date and time of the last update of this REST resource.
lifecycleStatus	A string. Used to indicate the current lifecycle status of the service specification.
isBundle	A boolean. A flag indicates that if this service specification is a bundled specification (true) or single (false).
resourceSpecification	A list of resource specification references (ResourceSpecificationRef [*]). The ResourceSpecification is required for a service specification with type ResourceFacingServiceSpecification (RFSS).
attachment	A list of attachments (Attachment [*]). Complements the description of an element (for instance a product) through video, pictures...
serviceSpecCharacteristic	A list of service spec characteristics (ServiceSpecCharacteristic [*]). This class represents the key features of this service specification. For example, bandwidth is a characteristic of many different types of services; if bandwidth is a relevant characteristic (e.g., from the point-of-view of a Customer obtaining this Service via a Product) then bandwidth would be a ServiceSpecCharacteristic for that particular Service.
relatedParty	A list of related party references (RelatedPartyRef [*]). A related party defines party or party role linked to a specific entity.
serviceSpecRelationship	A list of service spec relationships (ServiceSpecRelationship [*]). A migration, substitution, dependency or exclusivity relationship between/among service specifications.
targetServiceSchema	A target service schema reference (TargetServiceSchemaRef). The reference object to the schema and type of target service which is described by service specification.

Attachment sub-resource

Complements the description of an element (for instance a product) through video, pictures...

description	A string. A narrative text describing the content of the attachment.
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).

ServiceSpecCharRelationship sub-resource

An aggregation, migration, substitution, dependency or exclusivity relationship between/among Specification Characteristics.

type	A string. Type of relationship such as aggregation, migration, substitution, dependency, exclusivity.
name	A string. Name of the target characteristic.
id	A string. Unique identifier of the target specification.
href	A string. Hyperlink reference to the target specification.
@type	A string. class type of target specification.
validFor	A time period. The period for which the object is valid.

ServiceSpecCharacteristic sub-resource

This class represents the key features of this service specification. For example, bandwidth is a characteristic of many different types of services; if bandwidth is a relevant characteristic (e.g., from the point-of-view of a Customer obtaining this Service via a Product) then bandwidth would be a ServiceSpecCharacteristic for that particular Service.

name	A string. A word, term, or phrase by which this characteristic specification is known and distinguished from other characteristic specifications.
description	A string. A narrative that explains in detail what the ServiceSpecCharacteristic is.
valueType	A string. A kind of value that the characteristic can take on, such as numeric, text and so forth.
configurable	A boolean. If true, the Boolean indicates that the ServiceSpecCharacteristic is configurable.
validFor	A time period. The period for which the ServiceSpecCharacteristic is valid.
@type	A string. (Class) type of the ServiceSpecCharacteristic.
@schemaLocation	A string. A link to the schema describing this characteristic.

@valueSchemaLocation	A string. This (optional) field provides a link to the schema describing the value type.
minCardinality	An integer. The minimum number of instances a CharacteristicValue can take on. For example, zero to five phone numbers in a group calling plan, where zero is the value for the minCardinality.
maxCardinality	An integer. The maximum number of instances a CharacteristicValue can take on. For example, zero to five phone numbers in a group calling plan, where five is the value for the maxCardinality.
isUnique	A boolean. An indicator that specifies if a value is unique for the specification. Possible values are; "unique while value is in effect" and "unique whether value is in effect or not".
regex	A string. A rule or principle represented in regular expression used to derive the value of a characteristic value.
extensible	A boolean. An indicator that specifies that the values for the characteristic can be extended by adding new values when instantiating a characteristic for an Entity.
serviceSpecCharacteristicValue	A list of service spec characteristic values (ServiceSpecCharacteristicValue [*]). A ServiceSpecCharacteristicValue object is used to define a set of attributes, each of which can be assigned to a corresponding set of attributes in a ServiceSpecCharacteristic object. The values of the attributes in the ServiceSpecCharacteristicValue object describe the values of the attributes that a corresponding ServiceSpecCharacteristic object can take on.
serviceSpecCharRelationship	A list of service spec char relationships (ServiceSpecCharRelationship [*]). An aggregation, migration, substitution, dependency or exclusivity relationship between/among Specification Characteristics.

ServiceSpecCharacteristicValue sub-resource

A ServiceSpecCharacteristicValue object is used to define a set of attributes, each of which can be assigned to a corresponding set of attributes in a ServiceSpecCharacteristic object. The values of the attributes in the ServiceSpecCharacteristicValue object describe the values of the attributes that a corresponding ServiceSpecCharacteristic object can take on.

valueType	A string. A kind of value that the characteristic value can take on, such as numeric, text and so forth.
isDefault	A boolean. If true, the Boolean Indicates if the value is the default value for a characteristic.
value	An object (Object). the value that the characteristic can take on.

unitOfMeasure	A string. A length, surface, volume, dry measure, liquid measure, money, weight, time, and the like. In general, a determinate quantity or magnitude of the kind designated, taken as a standard of comparison for others of the same kind, in assigning to them numerical values, as 1 foot, 1 yard, 1 mile, 1 square foot.
validFor	A time period. The period of time for which a value is applicable.
valueFrom	An integer. The low range value that a characteristic can take on.
valueTo	An integer. The upper range value that a characteristic can take on.
rangeInterval	A string. An indicator that specifies the inclusion or exclusion of the valueFrom and valueTo attributes. If applicable, possible values are "open", "closed", "closedBottom" and "closedTop".
regex	A string. A regular expression constraint for given value.
@type	A string. The class type of the characteristic value.
@schemaLocation	A string. Hyperlink reference to schema describing this object.

ServiceSpecRelationship sub-resource

A migration, substitution, dependency or exclusivity relationship between/among service specifications.

type	A string. Type of relationship such as migration, substitution, dependency, exclusivity.
role	A string. The association role for this service specification.
id	A string. Unique identifier of target ServiceSpecification.
href	A string. Reference of the target ServiceSpecification.
name	A string. The name given to the target service specification instance.
validFor	A time period. The period for which the ServiceSpecRelationship is valid.

RelatedPartyRef relationship

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

id	A string. Unique identifier of a related party.
href	A string. Reference of the related party, could be a party reference or a party role reference.
role	A string. Role of the related party.
name	A string. Name of the related party.

validFor A time period. Validity period of the related party.

ResourceSpecificationRef relationship

Resource Specification reference: The ResourceSpecification is required for a ResourceFacingServiceSpecification (RFSS). This relationship should be NULL for a CustomerFacingServiceSpecification (CFSS).

id A string. Unique identifier of the resource specification.

href A string. Hyperlink reference of the resource specification.

name A string. Name of the ResourceSpecification.

version A string. Resource specification version.

TargetServiceSchemaRef relationship

The reference object to the schema and type of target service which is described by service specification.

@type A string. Class type of the target service.

@schemaLocation A string. This field provides a link to the schema describing the target service.

Json representation sample

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

```
{
  "id": "7655",
  "href": "https://host:port/catalogManagement/serviceSpecification/7655",
  "name": "Firewall Service",
  "description": "This service specification ...",
  "@type": "ResourceFacingServiceSpec",
  "@schemaLocation": "https://host:port/catalogManagement/schema/ResourceFacingServiceSpecification.yml",
  "@baseType": "ServiceSpecification",
  "version": "2.1",
  "validFor": {
    "startDateTime": "2017-08-23T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lastUpdate": "2017-08-27T00:00",
  "lifecycleStatus": "Active",
  "isBundle": false,
  "resourceSpecification": [
    {
      "id": "42",
      "href": "http://hostname:port/catalogManagement/resourceFunctionSpec/42",
      "name": "Firewall",
      "version": "1.0"
    }
  ],
  "attachment": [
    {
```

```

    "description": "This attachment ...",
    "href": "http://hostname:port/documentManagement/attachment/22",
    "id": "22",
    "type": "Document",
    "url": "http://xxxxx"
  }
],
"serviceSpecCharacteristic": [
  {
    "name": "OperatingSystem",
    "description": "This service spec characteristic ...",
    "valueType": "String",
    "@valueSchemaLocation": "",
    "configurable": true,
    "validFor": {
      "startDateTime": "2017-08-12T00:00",
      "endDateTime": "2018-03-07T00:00"
    },
    "@type": "ServiceSpecCharacteristic",
    "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceSpecCharacteristic.yml",
    "minCardinality": 0,
    "maxCardinality": 1,
    "isUnique": true,
    "regex": "",
    "extensible": false,
    "serviceSpecCharRelationship": [
      {
        "type": "string",
        "name": "OperatingSystem",
        "id": "4690",
        "href": "https://host:port/catalogManagement/serviceSpecification/4690",
        "@type": "CustomerFacingServiceSpec",
        "validFor": {
          "startDateTime": "2017-08-11T00:00",
          "endDateTime": "2018-03-07T00:00"
        }
      }
    ]
  },
],
"serviceSpecCharacteristicValue": [
  {
    "isDefault": true,
    "value": "Android KitKat",
    "validFor": {
      "startDateTime": "2017-08-06T00:00",
      "endDateTime": "2018-03-07T00:00"
    }
  }
]
},
{
  "name": "Scalability",
  "description": "Scalability parameters for this resource facing service spec ",
  "valueType": "CapabilityScalable",
  "@valueSchemaLocation": "https://host:port/catalogManagement/schema/CapabilityScalable.yml",
  "configurable": true,

```

```

"validFor": {
  "startDateTime": "2017-08-17T00:00",
  "endDateTime": "2018-03-12T00:00"
},
"@type": "ServiceSpecCharacteristic",
"@schemaLocation": "https://host:port/catalogManagement/schema/ServiceSpecCharacteristic.yml",
"minCardinality": 0,
"maxCardinality": 1,
"isUnique": true,
"extensible": true,
"serviceSpecCharRelationship": [
],
"serviceSpecCharacteristicValue": [
  {
    "valueType": "Object",
    "value": {
      "minInstances": 1,
      "maxInstances": 1000
    },
    "isDefault": true,
    "validFor": {
      "startDateTime": "2017-08-17T00:00",
      "endDateTime": "2018-03-12T00:00"
    },
    "@Type": "CapabilityScalable",
    "@schemaLocation": "https://host:port/catalogManagement/schema/CapabilityScalable.yml"
  }
]
},
],
"relatedParty": [
  {
    "id": "3643",
    "href": "https://host:port/partyManagement/organization/3643",
    "role": "Supplier",
    "name": "Firewall Express",
    "validFor": {
      "startDateTime": "2017-08-25T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
],
"serviceSpecRelationship": [
  {
    "type": "a string ...",
    "role": "a string ...",
    "id": "5563",
    "href": "https://host:port/catalogManagement/serviceSpecification/5563",
    "name": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-25T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
],
],

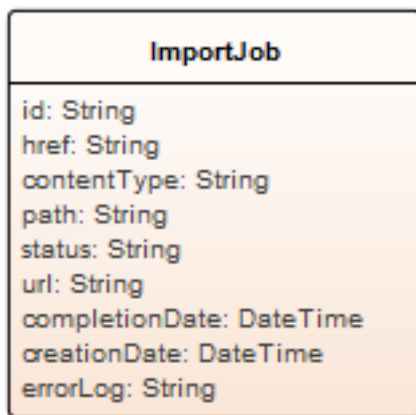
```

```
"targetServiceSchema": {  
  "@type": "RFS",  
  "@schemaLocation": "https://host:port/catalogManagement/schema/RFS.yml"  
}  
}
```

IMPORT JOB RESOURCE

Represents a task used to import resources from a file.

Resource model



Field descriptions

ImportJob fields

id	A string. Identifier of the import job.
href	A string. Reference of the import job.
contentType	A string. Indicates the format of the imported data.
path	A string. URL of the root resource where the content of the file specified by the import job must be applied.
status	A string. Status of the import job (not started, running, succeeded, failed).
url	A string. URL of the file containing the data to be imported.
completionDate	A date time (DateTime). Date at which the job was completed.
creationDate	A date time (DateTime). Date at which the job was created.
errorLog	A string. Reason for failure if status is failed.

Json representation sample

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

```

{
  "id": "7497",
  "href": "https://host:port/catalogManagement/importJob/7497",
  "contentType": "application/json",

```

```
"path": "/warning/system",  
"status": "completed",  
"url": "https://my/daily/job/NHCFD6",  
"completionDate": "2017-08-27T00:00",  
"creationDate": "2017-08-27T00:00",  
"errorLog": "http://my-platform/logging/errors.log"  
}
```

EXPORT JOB RESOURCE

Represents a task used to export resources to a file.

Resource model



Field descriptions

ExportJob fields

id	A string. Identifier of the export job.
href	A string. Reference of the export job.
query	A string. Used to scope the exported data.
path	A string. URL of the root resource acting as the source for streaming content to the file specified by the export job.
contentType	A string. The format of the exported data.
status	A string. Status of the export job (not started, running, succeeded, failed).
url	A string. URL of the file containing the data to be exported.
completionDate	A date time (DateTime). Data at which the job was completed.
creationDate	A date time (DateTime). Date at which the job was created.
errorLog	A string. Reason for failure.

Json representation sample

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

```
{
```

```
"id": "1866",  
"href": "https://host:port/catalogManagement/exportJob/1866",  
"query": "advancedCatalog",  
"path": "/warning/system",  
"contentType": "application/json",  
"status": "running",  
"url": "https://my/daily/job/NHCFD6",  
"completionDate": "2017-08-27T00:00",  
"creationDate": "2017-08-27T00:00",  
"errorLog": "http://my-platform/logging/errors.log"  
}
```


Notification Resource Models

9 notifications are defined for this API

Notifications related to ServiceCatalog:

- ServiceCatalogCreationNotification
- ServiceCatalogRemoveNotification
- ServiceCatalogBatchNotification

Notifications related to ServiceCategory:

- ServiceCategoryCreationNotification
- ServiceCategoryRemoveNotification

Notifications related to ServiceCandidate:

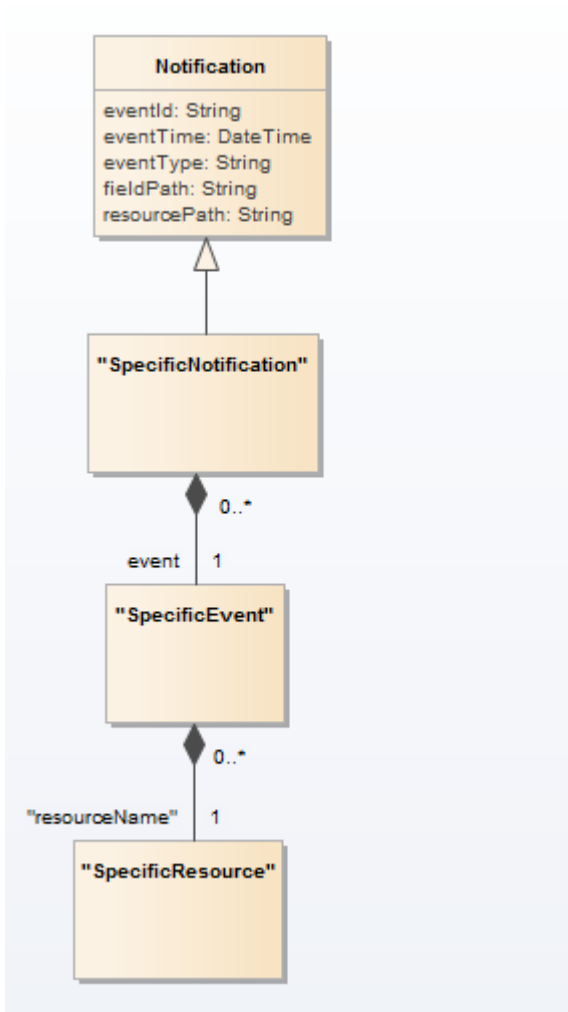
- ServiceCandidateCreationNotification
- ServiceCandidateRemoveNotification

Notifications related to ServiceSpecification:

- ServiceSpecificationCreationNotification
- ServiceSpecificationRemoveNotification

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 occurrence (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).



SERVICE CATALOG CREATION NOTIFICATION

Notification sent when a new ServiceCatalog resource is created.

Json representation sample

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

```

{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "ServiceCatalogCreationNotification",
  "event": {
    "serviceCatalog":
      {-- SEE ServiceCatalog RESOURCE SAMPLE --}
  }
}

```

SERVICE CATALOG REMOVE NOTIFICATION

Notification sent when removing a ServiceCatalog resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceCatalogRemoveNotification",
  "event": {
    "serviceCatalog" :
      [-- SEE ServiceCatalog RESOURCE SAMPLE --]
  }
}
```

SERVICE CATALOG BATCH NOTIFICATION

Notification sent when a batch job on resource ServiceCatalog changes

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceCatalogBatchNotification",
  "event": {
    "serviceCatalog" :
      [-- SEE ServiceCatalog RESOURCE SAMPLE --]
  }
}
```

SERVICE CATEGORY CREATION NOTIFICATION

Notification sent when a new ServiceCategory resource is created.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceCategoryCreationNotification",
  "event": {
```

```

    "serviceCategory" :
      {-- SEE ServiceCategory RESOURCE SAMPLE --}
  }
}

```

SERVICE CATEGORY REMOVE NOTIFICATION

Notification sent when removing a ServiceCategory resource.

Json representation sample

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

```

{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceCategoryRemoveNotification",
  "event": {
    "serviceCategory" :
      {-- SEE ServiceCategory RESOURCE SAMPLE --}
  }
}

```

SERVICE CANDIDATE CREATION NOTIFICATION

Notification sent when a new ServiceCandidate resource is created.

Json representation sample

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

```

{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceCandidateCreationNotification",
  "event": {
    "serviceCandidate" :
      {-- SEE ServiceCandidate RESOURCE SAMPLE --}
  }
}

```

SERVICE CANDIDATE REMOVE NOTIFICATION

Notification sent when removing a ServiceCandidate resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceCandidateRemoveNotification",
  "event": {
    "serviceCandidate" :
      [-- SEE ServiceCandidate RESOURCE SAMPLE --]
  }
}
```

SERVICE SPECIFICATION CREATION NOTIFICATION

Notification sent when a new ServiceSpecification resource is created.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceSpecificationCreationNotification",
  "event": {
    "serviceSpecification" :
      [-- SEE ServiceSpecification RESOURCE SAMPLE --]
  }
}
```

SERVICE SPECIFICATION REMOVE NOTIFICATION

Notification sent when removing a ServiceSpecification resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"ServiceSpecificationRemoveNotification",
  "event": {
    "serviceSpecification" :
      [-- SEE ServiceSpecification 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 SERVICE CATALOG

LIST SERVICE CATALOGS

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

Description

This operation list service catalog 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 ServiceCatalog resources.

Request
GET /catalogManagement/serviceCatalog Accept: application/json
Response
200 [{ "id": "2355", "href": "https://host:port/catalogManagement/serviceCatalog/2355", "name": "IOT Service Catalog", "description": "This service catalog ...", "@type": "ServiceCatalog", "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCatalog.yml", "@baseType": "Catalog", "version": "1.0", "validFor": { "startDateTime": "2017-08-17T00:00", "endDateTime": "2018-03-12T00:00" }, "lastUpdate": "2017-08-14T00:00", "lifecycleStatus": "Active", "relatedParty": [{ "id": "3426", "href": "https://host:port/partyManagement/organization/3426", "role": "vendor", "name": "Company ABC", "validFor": {


```

        "startDateTime": "2017-08-14T00:00",
        "endDateTime": "2018-03-12T00:00"
    }
}
],
"category": [
  {
    "id": "7752",
    "href": "https://host:port/catalogManagement/category/7752",
    "version": "1.0",
    "name": "IoT"
  }
]
}
]

```

RETRIEVE SERVICE CATALOG

GET /serviceCatalog/{id}?fields=...&{filtering}

Description

This operation retrieves a service catalog 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 a ServiceCatalog resource.

Request
<pre> GET /catalogManagement/serviceCatalog/2355 Accept: application/json </pre>
Response
<pre> 200 { "id": "2355", "href": "https://host:port/catalogManagement/serviceCatalog/2355", "name": "IOT Service Catalog", "description": "This service catalog ...", "@type": "ServiceCatalog", "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCatalog.yml", </pre>

```

"@baseType": "Catalog",
"version": "1.0",
"validFor": {
  "startDateTime": "2017-08-17T00:00",
  "endDateTime": "2018-03-12T00:00"
},
"lastUpdate": "2017-08-14T00:00",
"lifecycleStatus": "Active",
"relatedParty": [
  {
    "id": "3426",
    "href": "https://host:port/partyManagement/organization/3426",
    "role": "vendor",
    "name": "Company ABC",
    "validFor": {
      "startDateTime": "2017-08-14T00:00",
      "endDateTime": "2018-03-12T00:00"
    }
  }
],
"category": [
  {
    "id": "7752",
    "href": "https://host:port/catalogManagement/category/7752",
    "version": "1.0",
    "name": "IoT"
  }
]
}

```

CREATE SERVICE CATALOG

POST /serviceCatalog

Note: this operation is available only to ADMIN API users

Description

This operation creates a service catalog entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceCatalog, 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		
@type	ServiceCatalog	

@schemaLocation		
@baseType	Catalog	
version		
validFor		
lastUpdate		
lifecycleStatus		

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
@type	ServiceCatalog
@baseType	Catalog

Usage Samples

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

<p>Request</p> <pre>POST /catalogManagement/serviceCatalog Content-Type: application/json { "name": "IOT Service Catalog" }</pre>
<p>Response</p> <pre>201 { "id": "3830", "href": "https://host:port/catalogManagement/serviceCatalog/3830", "name": "IOT Service Catalog", "description": "", "@type": "ServiceCatalog", "@schemaLocation": "", "@baseType": "Catalog", "version": "1.0", "validFor": { "startDateTime": "2017-08-17T00:00", "endDateTime": "" }, "lastUpdate": "2017-08-17T00:00", "lifecycleStatus": "In Design",</pre>

```

"relatedParty": [
],
"category": [
]
}

```

PATCH SERVICE CATALOG

PATCH /serviceCatalog/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a service catalog 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
name	
description	
@schemaLocation	
@baseType	
version	
validFor	
lifecycleStatus	

Non Patchable Attributes	Rule
id	
href	
@type	
lastUpdate	

Usage Samples

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

Request

PATCH /catalogManagement/serviceCatalog/3830 Content-Type: application/merge-patch+json <pre>{ "name": "new name" }</pre>
Response
200 <pre>{ "id": "3830", "href": "https://host:port/catalogManagement/serviceCatalog/3830", "name": "new name", "description": "", "@type": "ServiceCatalog", "@schemaLocation": "", "@baseType": "Catalog", "version": "1.0", "validFor": { "startDateTime": "2017-08-17T00:00", "endDateTime": "" }, "lastUpdate": "2017-08-17T00:00", "lifecycleStatus": "In Design", "relatedParty": [], "category": [] }</pre>

DELETE SERVICE CATALOG

DELETE /serviceCatalog/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service catalog entity.

Usage Samples

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

Request
DELETE /catalogManagement/serviceCatalog/42
Response
204

OPERATIONS ON SERVICE CATEGORY

LIST SERVICE CATEGORIES

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

Description

This operation list service category 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 ServiceCategory resources.

Request
GET /catalogManagement/serviceCategory Accept: application/json
Response
200 [{ "id": "1708", "href": "https://host:port/catalogManagement/serviceCategory/1708", "name": "a string ...", "description": "This service category ...", "@type": "a string ...", "@schemaLocation": "a string ...", "@baseType": "a string ...", "version": "1.0", "validFor": { "startDateTime": "2017-08-24T00:00", "endDateTime": "2018-03-25T00:00" }, "lifecycleStatus": "a string ...", "lastUpdate": "2017-08-27T00:00", "parentId": "589", "isRoot": false, "relatedParty": [{ "id": "9555",

```

    "href": "https://host:port/partyManagement/organization/9555",
    "role": "seller",
    "name": "John Doe",
    "validFor": {
      "startDateTime": "2017-08-24T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  },
  "serviceCandidate": [
    {
      "id": "5850",
      "href": "https://host:port/catalogManagement/serviceCandidate/5850",
      "version": "1.1",
      "name": "Speed Max",
      "@type": "a string ..."
    }
  ],
  "category": [
    {
      "id": "6086",
      "href": "https://host:port/catalogManagement/category/6086",
      "version": "1.5",
      "name": "Cloud"
    }
  ]
}
]

```

RETRIEVE SERVICE CATEGORY

GET /serviceCategory/{id}?fields=...&{filtering}

Description

This operation retrieves a service category 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 a ServiceCategory resource.

Request

```

GET /catalogManagement/serviceCategory/1708
Accept: application/json

```


Response

200

```

{
  "id": "1708",
  "href": "https://host:port/catalogManagement/serviceCategory/1708",
  "name": "a string ...",
  "description": "This service category ...",
  "@type": "a string ...",
  "@schemaLocation": "a string ...",
  "@baseType": "a string ...",
  "version": "1.0",
  "validFor": {
    "startDateTime": "2017-08-24T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lifecycleStatus": "a string ...",
  "lastUpdate": "2017-08-27T00:00",
  "parentId": "589",
  "isRoot": false,
  "relatedParty": [
    {
      "id": "9555",
      "href": "https://host:port/partyManagement/organization/9555",
      "role": "seller",
      "name": "John Doe",
      "validFor": {
        "startDateTime": "2017-08-24T00:00",
        "endDateTime": "2018-03-25T00:00"
      }
    }
  ],
  "serviceCandidate": [
    {
      "id": "5850",
      "href": "https://host:port/catalogManagement/serviceCandidate/5850",
      "version": "1.1",
      "name": "Speed Max",
      "@type": "a string ..."
    }
  ],
  "category": [
    {
      "id": "6086",
      "href": "https://host:port/catalogManagement/category/6086",
      "version": "1.5",
      "name": "Cloud"
    }
  ]
}

```

CREATE SERVICE CATEGORY

POST /serviceCategory

Note: this operation is available only to ADMIN API users

Description

This operation creates a service category entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceCategory, 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		
@type	ServiceCategory	
@schemaLocation		
@baseType	Category	
version		
validFor		
lifecycleStatus		
lastUpdate		
parentId		
isRoot		
relatedParty		
serviceCandidate		
category		

Usage Samples

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

Request
POST /catalogManagement/serviceCategory Content-Type: application/json <pre>{ "name": "a string ..." }</pre>

Response
<pre> 201 { "id": "1708", "href": "https://host:port/catalogManagement/serviceCategory/1708", "name": "a string ..." ... } </pre>

PATCH SERVICE CATEGORY

PATCH /serviceCategory/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a service category 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
name	
description	
@schemaLocation	
@baseType	
version	
validFor	
lifecycleStatus	
parentId	
isRoot	
relatedParty	
serviceCandidate	
category	

Non Patchable Attributes	Rule
--------------------------	------

id	
href	
@type	
lastUpdate	

Usage Samples

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

Request
<pre>PATCH /catalogManagement/serviceCategory/1708 Content-Type: application/merge-patch+json { "name": "new name" }</pre>
Response
<pre>200 { "id": "1708", "href": "https://host:port/catalogManagement/serviceCategory/1708", "name": "new name", "description": "This service category ...", "@type": "a string ...", "@schemaLocation": "a string ...", "@baseType": "a string ...", "version": "1.0", "validFor": { "startDateTime": "2017-08-24T00:00", "endDateTime": "2018-03-25T00:00" }, "lifecycleStatus": "a string ...", "lastUpdate": "2017-08-27T00:00", "parentId": "589", "isRoot": false, "relatedParty": [{ "id": "9555", "href": "https://host:port/partyManagement/organization/9555", "role": "seller", "name": "John Doe", "validFor": { "startDateTime": "2017-08-24T00:00", "endDateTime": "2018-03-25T00:00" } }] }</pre>

```

],
"serviceCandidate": [
  {
    "id": "5850",
    "href": "https://host:port/catalogManagement/serviceCandidate/5850",
    "version": "1.1",
    "name": "Speed Max",
    "@type": "a string ..."
  }
],
"category": [
  {
    "id": "6086",
    "href": "https://host:port/catalogManagement/category/6086",
    "version": "1.5",
    "name": "Cloud"
  }
]
}

```

DELETE SERVICE CATEGORY

DELETE /serviceCategory/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service category entity.

Usage Samples

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

Request
DELETE /catalogManagement/serviceCategory/42
Response
204

OPERATIONS ON SERVICE CANDIDATE

LIST SERVICE CANDIDATES

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

Description

This operation list service candidate 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 ServiceCandidate resources.

Request
GET /catalogManagement/serviceCandidate Accept: application/json
Response
200 [{ "id": "4994", "href": "https://host:port/catalogManagement/serviceCandidate/4994", "name": "a string ...", "description": "This service candidate ...", "@type": "a string ...", "@schemaLocation": "a string ...", "@baseType": "a string ...", "version": "2.1", "validFor": { "startDateTime": "2017-08-23T00:00", "endDateTime": "2018-03-25T00:00" }, "lastUpdate": "2017-08-27T00:00", "lifecycleStatus": "a string ...", "category": [{ "id": "5980", "href": "https://host:port/catalogManagement/category/5980", "version": "3.2",

```

    "name": "TV"
  }
],
"serviceSpecification": {
  "id": "9600",
  "href": "https://host:port/catalogManagement/serviceSpecification/9600",
  "version": "2.1",
  "name": "Boabab1500",
  "@type": "a string ..."
}
}
]

```

GET/...depth

The depth parameter allows to list resources at different levels of depth. The resource properties are provided and the related resources are returned according to the depth value. This feature can exist with enabled filtering and attribute selection.

This operation can return a very large amount of data. The Content-Range header must be used to control the amount of data returned. The header is present in the request and control the minimum and maximum values returned. In the response, the content-Range header is used to indicate the presence of more elements in the collection and the current position of the elements in the overall collection.

Usage Samples:

<pre> GET /catalogManagement/serviceCandidate?depth=3&category.name=IOT Range:items=23-24 Accept: application/json </pre>
<p>RESPONSE</p> <pre> 200 Content-Range:items 23-24/50 [{ "id": "203", "href": "http://serverlocation:port/catalogManagement/serviceCandidate/203", "version": "2.0", "@type": "ServiceCandidate", "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCandidate.json", "@baseType": "", "lastUpdate": "2017-04-19T16:42:23-04:00", </pre>

```

"name": "SmartDeviceService",
"description": "A service candidate ...",
"lifecycleStatus": "Active",
"validFor": {
  "startDateTime": "2016-04-19T16:42:23-04:00",
  "endDateTime": "2016-06-19T00:00:00-04:00"
},
"category": [
  {
    "id": "8121",
    "href": "https://host:port/catalogManagement/serviceCategory/8121",
    "name": "IOT",
    "description": "This service category ...",
    "@type": "ServiceCategory",
    "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCategory.yml",
    "@baseType": "Category",
    "version": "1.0",
    "validFor": {
      "startDateTime": "2017-08-11T00:00",
      "endDateTime": "2018-03-07T00:00"
    },
    "lifecycleStatus": "Active",
    "lastUpdate": "2017-08-09T00:00",
    "parentId": "636",
    "isRoot": false,
    "category": [
      {
        "id": "5982",
        "href": "https://host:port/catalogManagement/category/5982",
        "version": "3.2",
        "name": "IOT Resources"
      }
    ]
  },
  "serviceCandidate": [
    {
      "id": "203",
      "href": "https://host:port/catalogManagement/serviceCandidate/203",
      "version": "1.0",
      "name": "SmartDeviceService"
    },
    {
      "id": "123",
      "href": "https://host:port/catalogManagement/serviceCandidate/123",
      "version": "1.0",
      "name": "Wireless HUB Service"
    }
  ],
  "relatedParty": [
    {
      "id": "4434",

```



```

        "href": "https://host:port/partyManagement/organization/4434",
        "role": "employee",
        "name": "Jimmy Doe",
        "validFor": {
            "startDateTime": "2017-08-10T00:00",
            "endDateTime": "2018-03-07T00:00"
        }
    }
}
],
"serviceSpecification": [
{
    "id": "8115",
    "href": "https://host:port/catalogManagement/customerFacingServiceSpecification/8115",
    "name": "Smart Device Service",
    "description": "This Service spec ...",
    "@type": "CustomerFacingServiceSpecification",
    "@schemaLocation":
"https://host:port/catalogManagement/schema/CustomerFacingServiceSpecification.json",
    "@baseType": "ServiceSpecification",
    "version": "1.0",
    "validFor": {
        "startDateTime": "2017-08-16T00:00",
        "endDateTime": "2018-03-12T00:00"
    },
    "lastUpdate": "2017-08-14T00:00",
    "lifecycleStatus": "Active",
    "isBundle": false,
    "targetServiceSchema": {
        "@type": "CFS",
        "@schemaLocation": "https://host:port/catalogManagement/schema/CFS.json",
    },
    "attachment": [],
    "relatedParty": [
        {
            "id": "3362",
            "href": "https://host:port/partyManagement/organization/3362",
            "role": "Distributor",
            "name": "ABC",
            "validFor": {
                "startDateTime": "2017-08-13T00:00",
                "endDateTime": "2018-03-12T00:00"
            }
        }
    ]
},
"serviceSpecCharacteristic": [
{
    "name": "memorySize",
    "description": "This service spec characteristic ...",

```

```

"valueType": "Integer",
"configurable": false,
"validFor": {
  "startDateTime": "2017-08-10T00:00",
  "endDateTime": "2018-03-12T00:00"
},
"minCardinality": 1,
"maxCardinality": 1,
"isUnique": true,
"serviceSpecCharRelationship": [
],
"serviceSpecCharacteristicValue": [
  {
    "valueType": "Integer",
    "isDefault": true,
    "value": 16,
    "unitOfMeasure": "GB",
    "validFor": {
      "startDateTime": "2017-08-16T00:00",
      "endDateTime": "2018-03-12T00:00"
    }
  }
]
},
"serviceSpecRelationship": [
  {
    "type": "Requires",
    "id": "5751",
    "href": "https://host:port/catalog/resourceFacingServiceSpecification/5751",
    "name": "SmartDevice RFS",
    "validFor": {
      "startDateTime": "2017-08-16T00:00",
      "endDateTime": "2018-03-12T00:00"
    }
  }
]
},
{
  "id": "123",
  "href": "http://serverlocation:port/catalogManagement/serviceCandidate/123",
  "@type": "ServiceCandidate",
  "@schemaLocation": "https://host:port/catalogManagement/schema/ServiceCandidate.Json",
  "@baseType": "",
  "version": "1.0",
  "lastUpdate": "2016-04-19T16:42:23-04:00",
  "name": "Wireless HUB Service",
  "description": " Wireless HUB Service ",

```

```

"lifecycleStatus": "Active",
"validFor": {
  "startDateTime": "2016-04-19T16:42:23-04:00",
  "endDateTime": "2016-06-19T00:00:00-04:00"
},
"category": [
  {
    "id": "8121",
    "href": "https://host:port/catalogManagement/serviceCategory/8121",
    "name": "IOT",
    "description": "This service category ...",
    "@type": "ServiceCategory",
    "@schemaLocation":
    "https://host:port/catalogManagement/schema/ServiceCategory.yml",
    "@baseType": "Category",
    "version": "1.0",
    "validFor": {
      "startDateTime": "2017-08-11T00:00",
      "endDateTime": "2018-03-07T00:00"
    },
    "lifecycleStatus": "Active",
    "lastUpdate": "2017-08-09T00:00",
    "parentId": "636",
    "isRoot": false,
    "category": [
      {
        "id": "5982",
        "href": "https://host:port/catalogManagement/category/5982",
        "version": "3.2",
        "name": "IOT Resources"
      }
    ],
    "serviceCandidate": [
      {
        "id": "203",
        "href": "https://host:port/catalogManagement/serviceCandidate/203",
        "version": "1.0",
        "name": "SmartDeviceService"
      },
      {
        "id": "123",
        "href": "https://host:port/catalogManagement/serviceCandidate/123",
        "version": "1.0",
        "name": "Wireless HUB Service"
      }
    ],
    "relatedParty": [
      {
        "id": "4434",
        "href": "https://host:port/partyManagement/organization/4434",

```

```

        "role": "employee",
        "name": "Jimmy Doe",
        "validFor": {
            "startDateTime": "2017-08-10T00:00",
            "endDateTime": "2018-03-07T00:00"
        }
    }
}
],
"serviceSpecification": [
    {
        "id": "405",
        "href":
"http://serverlocation:port/catalogManagement/customerFacingServiceSpecification/405",
        "version": "2.0",
        "name": "Wireless HUB CFSS ",
        "@type": "CustomerFacingServiceSpecification",
        "@schemaLocation":
"https://host:port/catalogManagement/schema/CustomerFacingServiceSpecification.
yml",
        "@baseType": "ServiceSpecification",
        "isBundle": false,
        "serviceSpecCharacteristic": [
            {
                "name": "technology",
                "description": "technology",
                "valueType": "string",
                "configurable": true,
                "minCardinality": 1,
                "maxCardinality": 2,
                "extensible": true,
                "@type": "ServiceSpecCharacteristic",
                "@schemaLocation":
"https://host:port/catalogManagement/schema/ServiceSpecCharacteristic.yml",
                "validFor": {
                    "startDateTime": "2017-04-19T16:42:23-04:00",
                    "endDateTime": ""
                },
                "serviceSpecCharacteristicValue": [
                    {
                        "valueType": "string",
                        "isDefault": "false",
                        "value": "3G",
                        "unitOfMeasure": "",
                        "valueFrom": "",
                        "valueTo": "",
                        "validFor": {
                            "startDateTime": "2017-04-19T16:42:23-04:00",

```

```
    "endDateTime": ""  
  },  
  },  
  {  
    "valueType": "string",  
    "isDefault": "false",  
    "value": "4G",  
    "unitOfMeasure": "",  
    "valueFrom": "",  
    "valueTo": "",  
    "validFor": {  
      "startDateTime": "2016-04-19T16:42:23-04:00",  
      "endDateTime": ""  
    }  
  }  
]  
}  
]
```

RETRIEVE SERVICE CANDIDATE

```
GET /serviceCandidate/{id}?fields=...&{filtering}
```

Description

This operation retrieves a service candidate 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 a ServiceCandidate resource.

Request

GET /catalogManagement/serviceCandidate/4994
 Accept: application/json

Response

```
200
{
  "id": "4994",
  "href": "https://host:port/catalogManagement/serviceCandidate/4994",
  "name": "a string ...",
  "description": "This service candidate ...",
  "@type": "a string ...",
  "@schemaLocation": "a string ...",
  "@baseType": "a string ...",
  "version": "2.1",
  "validFor": {
    "startDateTime": "2017-08-23T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lastUpdate": "2017-08-27T00:00",
  "lifecycleStatus": "a string ...",
  "category": [
    {
      "id": "5980",
      "href": "https://host:port/catalogManagement/category/5980",
      "version": "3.2",
      "name": "TV"
    }
  ],
  "serviceSpecification": {
    "id": "9600",
    "href": "https://host:port/catalogManagement/serviceSpecification/9600",
    "version": "2.1",
    "name": "Boabab1500",
    "@type": "a string ..."
  }
}
```

CREATE SERVICE CANDIDATE

POST /serviceCandidate

Note: this operation is available only to ADMIN API users

Description

This operation creates a service candidate entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceCandidate, 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		
@type	ServiceCandidate	
@schemaLocation		
@baseType		
version		
validFor		
lastUpdate		
lifecycleStatus		
category		
serviceSpecification		

Usage Samples

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

Request
POST /catalogManagement/serviceCandidate Content-Type: application/json <pre>{ "name": "a string ..." }</pre>
Response
201 <pre>{ "id": "4994", "href": "https://host:port/catalogManagement/serviceCandidate/4994", "name": "a string ..." }</pre>

PATCH SERVICE CANDIDATE

PATCH /serviceCandidate/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a service candidate 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
name	
description	
@schemaLocation	
@baseType	
version	
validFor	
lifecycleStatus	
category	
serviceSpecification	

Non Patchable Attributes	Rule
id	
href	
@type	
lastUpdate	

Usage Samples

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

Request
PATCH /catalogManagement/serviceCandidate/4994 Content-Type: application/merge-patch+json <pre>{ "name": "new name" }</pre>

}

Response

200

```

{
  "id": "4994",
  "href": "https://host:port/catalogManagement/serviceCandidate/4994",
  "name": "new name",
  "description": "This service candidate ...",
  "@type": "a string ...",
  "@schemaLocation": "a string ...",
  "@baseType": "a string ...",
  "version": "2.1",
  "validFor": {
    "startDateTime": "2017-08-23T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lastUpdate": "2017-08-27T00:00",
  "lifecycleStatus": "a string ...",
  "category": [
    {
      "id": "5980",
      "href": "https://host:port/catalogManagement/category/5980",
      "version": "3.2",
      "name": "TV"
    }
  ],
  "serviceSpecification": {
    "id": "9600",
    "href": "https://host:port/catalogManagement/serviceSpecification/9600",
    "version": "2.1",
    "name": "Boabab1500",
    "@type": "a string ..."
  }
}

```

DELETE SERVICE CANDIDATE**DELETE /serviceCandidate/{id}***Note: this operation is available only to ADMIN API users***Description**

This operation deletes a service candidate entity.

Usage Samples

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

Request
DELETE /catalogManagement/serviceCandidate/42
Response
204

OPERATIONS ON SERVICE SPECIFICATION

LIST SERVICE SPECIFICATIONS

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

Description

This operation list service 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 ServiceSpecification resources.

Request
GET /catalogManagement/serviceSpecification Accept: application/json
Response
200 [{ "id": "7655", "href": "https://host:port/catalogManagement/serviceSpecification/7655", "name": "Speed987", "description": "This service specification ...", "@type": "a string ...", "@schemaLocation": "a string ...", "@baseType": "a string ...", "version": "2.1", "validFor": { "startDateTime": "2017-08-23T00:00", "endDateTime": "2018-03-25T00:00" }, "lastUpdate": "2017-08-27T00:00", "lifecycleStatus": "a string ...", "isBundle": true, "resourceSpecification": [{ "id": "42", "href": "http://hostname:port/catalogManagement/resourceSpecification/42",

```

"name": "Robust600",
"version": "1.0"
}
],
"attachment": [
{
"description": "This attachment ...",
"href": "http://hostname:port/documentManagement/attachment/22",
"id": "22",
"type": "Document",
"url": "http://xxxxx"
}
],
"serviceSpecCharacteristic": [
{
"name": "a string ...",
"description": "This service spec characteristic ...",
"valueType": "a string ...",
"configurable": true,
"validFor": {
"startDateTime": "2017-08-29T00:00",
"endDateTime": "2018-03-25T00:00"
},
"@type": "a string ...",
"@schemaLocation": "a string ...",
"@valueSchemaLocation": "a string ...",
"minCardinality": 40,
"maxCardinality": 76,
"isUnique": true,
"regex": "a string ...",
"extensible": false,
"serviceSpecCharacteristicValue": [
{
"valueType": "a string ...",
"isDefault": false,
"value": "a Object ...",
"unitOfMeasure": "a string ...",
"validFor": {
"startDateTime": "2017-08-23T00:00",
"endDateTime": "2018-03-25T00:00"
},
"valueFrom": 1,
"valueTo": 48,
"rangeInterval": "a string ...",
"regex": "a string ...",
"@type": "a string ...",
"@schemaLocation": "a string ..."
}
],
"@type": "a string ...",
"@schemaLocation": "a string ..."
}
],
"serviceSpecCharRelationship": [
{
"type": "a string ...",
"name": "a string ...",
"id": "3682",
"href": "https://host:port/catalogManagement/serviceSpecification/3682",

```

```

    "@type": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-30T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
]
},
"relatedParty": [
  {
    "id": "3643",
    "href": "https://host:port/partyManagement/organization/3643",
    "role": "bill receiver",
    "name": "John Doe",
    "validFor": {
      "startDateTime": "2017-08-25T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
],
"serviceSpecRelationship": [
  {
    "type": "a string ...",
    "role": "a string ...",
    "id": "5563",
    "href": "https://host:port/catalogManagement/serviceSpecification/5563",
    "name": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-25T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
],
"targetServiceSchema": {
  "@type": "a string ...",
  "@schemaLocation": "a string ..."
}
}
]

```

RETRIEVE SERVICE SPECIFICATION

GET /serviceSpecification/{id}?fields=...&{filtering}

Description

This operation retrieves a service 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 a ServiceSpecification resource.

Request
GET /catalogManagement/serviceSpecification/7655 Accept: application/json
Response
200 <pre> { "id": "7655", "href": "https://host:port/catalogManagement/serviceSpecification/7655", "name": "Speed987", "description": "This service specification ...", "@type": "a string ...", "@schemaLocation": "a string ...", "@baseType": "a string ...", "version": "2.1", "validFor": { "startDateTime": "2017-08-23T00:00", "endDateTime": "2018-03-25T00:00" }, "lastUpdate": "2017-08-27T00:00", "lifecycleStatus": "a string ...", "isBundle": true, "resourceSpecification": [{ "id": "42", "href": "http://hostname:port/catalogManagement/resourceSpecification/42", "name": "Robust600", "version": "1.0" }], "attachment": [{ "description": "This attachment ...", "href": "http://hostname:port/documentManagement/attachment/22", "id": "22", "type": "Document", "url": "http://xxxxx" }], "serviceSpecCharacteristic": [{ "name": "a string ...", </pre>

```

"description": "This service spec characteristic ...",
"valueType": "a string ...",
"configurable": true,
"validFor": {
  "startDateTime": "2017-08-29T00:00",
  "endDateTime": "2018-03-25T00:00"
},
"@type": "a string ...",
"@schemaLocation": "a string ...",
"@valueSchemaLocation": "a string ...",
"minCardinality": 40,
"maxCardinality": 76,
"isUnique": true,
"regex": "a string ...",
"extensible": false,
"serviceSpecCharacteristicValue": [
  {
    "valueType": "a string ...",
    "isDefault": false,
    "value": "a Object ...",
    "unitOfMeasure": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-23T00:00",
      "endDateTime": "2018-03-25T00:00"
    },
    "valueFrom": 1,
    "valueTo": 48,
    "rangeInterval": "a string ...",
    "regex": "a string ...",
    "@type": "a string ...",
    "@schemaLocation": "a string ..."
  }
],
"serviceSpecCharRelationship": [
  {
    "type": "a string ...",
    "name": "a string ...",
    "id": "3682",
    "href": "https://host:port/catalogManagement/serviceSpecification/3682",
    "@type": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-30T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
]
},
"relatedParty": [
  {
    "id": "3643",
    "href": "https://host:port/partyManagement/organization/3643",
    "role": "bill receiver",
    "name": "John Doe",
    "validFor": {

```

```

        "startDateTime": "2017-08-25T00:00",
        "endDateTime": "2018-03-25T00:00"
    }
}
],
"serviceSpecRelationship": [
{
    "type": "a string ...",
    "role": "a string ...",
    "id": "5563",
    "href": "https://host:port/catalogManagement/serviceSpecification/5563",
    "name": "a string ...",
    "validFor": {
        "startDateTime": "2017-08-25T00:00",
        "endDateTime": "2018-03-25T00:00"
    }
}
],
"targetServiceSchema": {
    "@type": "a string ...",
    "@schemaLocation": "a string ..."
}
}
}

```

CREATE SERVICE SPECIFICATION

POST /serviceSpecification

Note: this operation is available only to ADMIN API users

Description

This operation creates a service specification entity.

Mandatory and Non Mandatory Attributes

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

Mandatory Attributes	Rule
name	
@type	

Non Mandatory Attributes	Default Value	Rule
description		
@schemaLocation		
@baseType		
version		
validFor		

lastUpdate		
lifecycleStatus		
isBundle	false	
resourceSpecification		
attachment		
serviceSpecCharacteristic		
relatedParty		
serviceSpecRelationship		
targetServiceSchema		

Additional Rules

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

Context	Mandatory Sub-Attributes
attachment	name
relatedParty	id or href
serviceSpecRelationship	type, id or href

Usage Samples

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

<p>Request</p> <p>POST /catalogManagement/serviceSpecification Content-Type: application/json</p> <pre>{ "name": "Speed987", "@type": "a string ..." }</pre>
<p>Response</p> <p>201</p> <pre>{ "id": "7655", "href": "https://host:port/catalogManagement/serviceSpecification/7655", "name": "Speed987", "@type": "a string ..." }</pre>

PATCH SERVICE SPECIFICATION

PATCH /serviceSpecification/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a service 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
name	
description	
@schemaLocation	
@baseType	
version	
validFor	
lifecycleStatus	
isBundle	
resourceSpecification	
attachment	
serviceSpecCharacteristic	
relatedParty	
serviceSpecRelationship	
targetServiceSchema	

Non Patchable Attributes	Rule
id	
href	
lastUpdate	
@type	

Usage Samples

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

Request

PATCH /catalogManagement/serviceSpecification/7655
 Content-Type: application/merge-patch+json

```
{
  "name": "new name"
}
```

Response

200

```
{
  "id": "7655",
  "href": "https://host:port/catalogManagement/serviceSpecification/7655",
  "name": "new name",
  "description": "This service specification ...",
  "@type": "a string ...",
  "@schemaLocation": "a string ...",
  "@baseType": "a string ...",
  "version": "2.1",
  "validFor": {
    "startDateTime": "2017-08-23T00:00",
    "endDateTime": "2018-03-25T00:00"
  },
  "lastUpdate": "2017-08-27T00:00",
  "lifecycleStatus": "a string ...",
  "isBundle": true,
  "resourceSpecification": [
    {
      "id": "42",
      "href": "http://hostname:port/catalogManagement/resourceSpecification/42",
      "name": "Robust600",
      "version": "1.0"
    }
  ],
  "attachment": [
    {
      "description": "This attachment ...",
      "href": "http://hostname:port/documentManagement/attachment/22",
      "id": "22",
      "type": "Document",
      "url": "http://xxxxx"
    }
  ],
  "serviceSpecCharacteristic": [
    {
      "name": "a string ...",
      "description": "This service spec characteristic ...",
      "valueType": "a string ...",

```

```

"configurable": true,
"validFor": {
  "startDateTime": "2017-08-29T00:00",
  "endDateTime": "2018-03-25T00:00"
},
"@type": "a string ...",
"@schemaLocation": "a string ...",
"@valueSchemaLocation": "a string ...",
"minCardinality": 40,
"maxCardinality": 76,
"isUnique": true,
"regex": "a string ...",
"extensible": false,
"serviceSpecCharacteristicValue": [
  {
    "valueType": "a string ...",
    "isDefault": false,
    "value": "a Object ...",
    "unitOfMeasure": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-23T00:00",
      "endDateTime": "2018-03-25T00:00"
    },
    "valueFrom": 1,
    "valueTo": 48,
    "rangeInterval": "a string ...",
    "regex": "a string ...",
    "@type": "a string ...",
    "@schemaLocation": "a string ..."
  }
],
"serviceSpecCharRelationship": [
  {
    "type": "a string ...",
    "name": "a string ...",
    "id": "3682",
    "href": "https://host:port/catalogManagement/serviceSpecification/3682",
    "@type": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-30T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
]
},
"relatedParty": [
  {
    "id": "3643",
    "href": "https://host:port/partyManagement/organization/3643",
    "role": "bill receiver",
    "name": "John Doe",
    "validFor": {
      "startDateTime": "2017-08-25T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
]

```

```

    }
  }
],
"serviceSpecRelationship": [
  {
    "type": "a string ...",
    "role": "a string ...",
    "id": "5563",
    "href": "https://host:port/catalogManagement/serviceSpecification/5563",
    "name": "a string ...",
    "validFor": {
      "startDateTime": "2017-08-25T00:00",
      "endDateTime": "2018-03-25T00:00"
    }
  }
],
"targetServiceSchema": {
  "@type": "a string ...",
  "@schemaLocation": "a string ..."
}
}
}

```

DELETE SERVICE SPECIFICATION

DELETE /serviceSpecification/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service specification entity.

Usage Samples

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

Request
DELETE /catalogManagement/serviceSpecification/42
Response
204

OPERATIONS ON IMPORT JOB

LIST IMPORT JOBS

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

Description

This operation list import job 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 ImportJob resources.

List all started active jobs. The result items are shrunk to show only the id (fields=id)

Request
<pre>GET /catalogManagement/importJob?fields=id&status="started" Accept: application/json</pre>
Response
<pre>200 [{ "id": "42" }, { "id": "43" }]</pre>

RETRIEVE IMPORT JOB

GET /importJob/{id}?fields=...&{filtering}

Description

This operation retrieves an import job 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 a ImportJob resource.

Request
GET /catalogManagement/importJob/7497 Accept: application/json
Response
200 <pre>{ "id": "7497", "href": "https://host:port/catalogManagement/importJob/7497", "contentType": "application/json", "path": "/warning/system", "status": "completed", "url": "https://my/daily/job/NHCFD6", "completionDate": "2017-08-27T00:00", "creationDate": "2017-08-27T00:00", "errorLog": "http://my-platform/logging/errors.log" }</pre>

CREATE IMPORT JOB

POST /importJob

Description

This operation creates an import job entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ImportJob, 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
url	

Non Mandatory Attributes	Default Value	Rule
contentType		
path		
status		
completionDate		
creationDate		
errorLog		

Usage Samples

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

Request
POST /catalogManagement/importJob Content-Type: application/json <pre>{ "url": "https://my/daily/job/NHCFD6" }</pre>
Response
201 <pre>{ "id": "7497", "href": "https://host:port/catalogManagement/importJob/7497", "url": "https://my/daily/job/NHCFD6" }</pre>

DELETE IMPORT JOB

DELETE /importJob/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes an import job entity.

Usage Samples

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

Request
DELETE /catalogManagement/importJob/42
Response
204

OPERATIONS ON EXPORT JOB

LIST EXPORT JOBS

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

Description

This operation list export job 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 ExportJob resources.

Request
GET /catalogManagement/exportJob Accept: application/json
Response
200 [{ "id": "1866", "href": "https://host:port/catalogManagement/exportJob/1866", "query": "advancedCatalog", "path": "/warning/system", "contentType": "application/json", "status": "running", "url": "https://my/daily/job/NHCFD6", "completionDate": "2017-08-27T00:00", "creationDate": "2017-08-27T00:00", "errorLog": "http://my-platform/logging/errors.log" }]

RETRIEVE EXPORT JOB

GET /exportJob/{id}?fields=...&{filtering}

Description

This operation retrieves an export job 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 a ExportJob resource.

Request
GET /catalogManagement/exportJob/1866 Accept: application/json
Response
200 <pre>{ "id": "1866", "href": "https://host:port/catalogManagement/exportJob/1866", "query": "advancedCatalog", "path": "/warning/system", "contentType": "application/json", "status": "running", "url": "https://my/daily/job/NHCFD6", "completionDate": "2017-08-27T00:00", "creationDate": "2017-08-27T00:00", "errorLog": "http://my-platform/logging/errors.log" }</pre>

CREATE EXPORT JOB

POST /exportJob

Description

This operation creates an export job entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ExportJob, 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
url	

Non Mandatory Attributes	Default Value	Rule
query		
path		
contentType		
status		
completionDate		
creationDate		
errorLog		

Usage Samples

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

Request
POST /catalogManagement/exportJob Content-Type: application/json <pre>{ "url": "https://my/daily/job/NHCFD6" }</pre>
Response
201 <pre>{ "id": "1866", "href": "https://host:port/catalogManagement/exportJob/1866", "url": "https://my/daily/job/NHCFD6" }</pre>

DELETE EXPORT JOB

DELETE /exportJob/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes an export job entity.

Usage Samples

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

Request
DELETE /catalogManagement/exportJob/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 the hub is shared and already exist and it doesn't support multiple listeners.

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 can 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.

LIFECYCLE MANAGEMENT EXTENSIONS TO CATALOG

In Lifecycle Management, there is a requirement to distinguish between entities existing with different life cycle version numbers and accessible via different ACL mechanisms. For example, the same Service Candidate may exist in a Catalog but with different version numbers.

It may be possible for an administrator to see all the existing versions or for a partner to see only a subset of all the existing versions.

The entity version number is not dependent on the version number of the API. For example, in PLM the same API (running at a specific version number) may be used to retrieve entities with different PLM version numbers.

In order to distinguish resources representing entities running with different version numbers and accessible through the same API version, the following directive can be used `/id:(version=x)` and the version attribute is added to each entity.

```
{
  "id": "VirtualStorageService",
  "href":
"http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorageService",
  "version": "1.0",
  "lastUpdate": "2017-04-19T16:42:23-04:00",
  "name": "Virtual Storage Service",
  "description": "Virtual Storage Service",
  "lifecycleStatus": "Active",
  .....
}
```

Note that the catalog resources in this case may have the same ID but may be distinguished by the inclusion of the version number in their ID i.e. `/VirtualStorageService:(version=1.0)`, `/VirtualStorageService:(version=2.0)`.

In the following examples, we will assume that two versions of the VirtualStorage Service Candidate exist in the Service Catalog. The Inactive and Active versions respectively version 1.0 and version 2.0.

Query all versioned catalog resources

Users with different roles may have access to different versions of the entities in the catalog. For example, user A may have access to only the version 1.0 of the entities while user B may have access to version 1.0 and version 2.0.

Admin user of Catalog have access to all the versions of the resources while non-admin users have by default access to only the latest version of the entities in the Catalog.

For example, the following request on the admin endpoint will return all the versioned resources matching a specific ID.

REQUEST
<pre>GET api/admin/catalogManagement/serviceCandidate/?id=VirtualStorageService Accept: application/json</pre>
RESPONSE
<pre>200 Content-Type: application/json [{ "id": "VirtualStorageService", "href": "http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorage Service", "version": "1.0", "lastUpdate": "2017-04-19T16:42:23-04:00", "name": "Virtual Storage Service", "description": "Virtual Storage Service", "lifecycleStatus": "Active", }, { "id": "VirtualStorageService", "href": "http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorage Service", "version": "2.0", "lastUpdate": "2017-04-19T16:42:23-04:00", "name": "Virtual Storage Service", "description": "Virtual Storage Service", "lifecycleStatus": "Active", }]</pre>

```

    }
]

```

Query a specific versioned catalog resource

In general, a non-admin API user only has visibility to the latest version number or visibility to a subset of versioned catalog resources.

It may be possible for an admin API user to retrieve a resource with a specific version number by using an ID and versioning filtering criteria.

REQUEST

```

GET api/admin/catalogManagement/serviceCandidate/?id=VirtualStorageService&version=1.0
Accept: application/json

```

RESPONSE

```

200
Content-Type: application/json
[
  {
    "id": "VirtualStorageMedium",
    "href": "http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorage
Service",
    "version": "1.0",
    "lastUpdate": "2017-01-19T16:42:23-04:00",
    "name": "Virtual Storage Service",
    "description": "Virtual Storage Service",
    "lifecycleStatus": "Active",
    .....
  },
  .....
]

```

Query current version of a catalog resource

By default, only the most current version is returned (for admin and non-admin).

REQUEST
GET api/admin/catalogManagement/serviceCandidate/VirtualStorageService Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{ "id": "VirtualStorageMedium", "href": "http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorage Service", "version": "2.0", "lastUpdate": "2017-04-19T16:42:23-04:00", "name": "Virtual Storage Service", "description": "Virtual Storage Service", "lifecycleStatus": "Active", }</pre>

Create new version of a catalog resource

POST is used to create a new version of a catalog resource.

The constraint is that the version numbers for the resource having the same ID must differ.

REQUEST
POST catalogManagement/serviceCandidate Content-type: application/json <pre>{</pre>

```

    "id": "VirtualStorage",
    "href": "http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorage
    Service",
    "version": "3.0",
    "name": "Virtual Storage Service",
    "description": "Virtual Storage Service",
    "lifecycleStatus": "Active",
    "validFor": {
        "startDateTime": "2017-08-19T16:42:23-04:00",
        "endDateTime": "2017-011-19T00:00:00-04:00"
    },
    ...
}

```

RESPONSE

```

201
Content-Type: application/json

{
    "id": "VirtualStorage",
    "href": "http://serverlocation:port/catalogManagement/serviceCandidate/VirtualStorage
    Service",
    "version": "3.0",
    "lastUpdate": "2017-08-19T16:42:23-04:00",
    "name": "Virtual Storage Service",
    "description": "Virtual Storage Service",
    "lifecycleStatus": "Active",
    "validFor": {
        "startDateTime": "2016-04-19T16:42:23-04:00",
        "endDateTime": "2016-06-19T00:00:00-04:00"
    },
    ...
}

```

Modify an existing version of a catalog resource

By default, PATCH or PUT will be acting only on the latest version of a catalog resource. For example, PATCH /.../serviceCandidate/VirtualStorageService will only update the VirtualStorageService ServiceCandidate at Version 2.0 (which is the most current).

To update a specific version of an entity the (Version=X) directive is added to the ID (i.e. /id:(version=x)).

Note that this capability is only available to API users having the proper authorizations to change the catalog entities with specific version numbers.

For example, to change the VirtualStorageService versioned at 1.0 we could use:
 /serviceCandidate/VirtualStorageService(Version=1.0)

REQUEST
PATCH /catalogManagement/serviceCandidate/VirtualStorageService(Version=1.0) Content-type: application/json-patch+json <pre>{ "lifecycleStatus": "Active" }</pre>
RESPONSE
200 Content-Type: application/json <pre>{ "id": "VirtualStorageService", "href": "http://serverlocation:port/catalogManagement/serviceCanddidate/VirtualStorageService", "version": "1.0", "lifecycleStatus": "Active", }</pre>

ROLE BASED ACCESS CONTROL

The user presents their credentials for authentication

If the credentials are valid

1. The user is given access to the catalog
2. As defined by their role(s)
3. As defined by their access rights
4. As defined by the access type: CRUD, discover
5. As defined by the pre-defined filter

For example, if they issue a get on a catalog that a party has no access they get an error response

Or if they try to modify an area of the catalog but do not have Write Access they get an error response

Normally we anticipate that the OAUTH2 or Open ID Connect are used as the authorization APIs and that ACL are established between authorized parties with regards to the content of the Catalog (i.e. GET but also enable of update operations on specific entities).

ACKNOWLEDGEMENTS

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	4/15/2013	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of Draft Version of the Document.
Release 1.2	8/15/2016	Frank Wong DGIT fwong@dgitsystems.com	Updated to resolve errors and reflect discussion following Team Action Week Vancouver 2016
Release 1.4	10/16/2016	Frank Wong DGIT fwong@dgitsystems.com	Incorporated comments from the review meeting
Release 2.0	8/27/2017	Kamal Maghsoudlou Ericsson kamal.maghsoudlou@ericsson.com	added extension pattern for REST resources

CONTRIBUTORS TO DOCUMENT

- Elaine Haher Ericsson
- Frank Wong FGIT
- Kamal Maghsoudlou Ericsson
- Michel Besson TM Forum
- Mariano Belaunde Orange
- Pierre Gauthier TM Forum