



Framework Specification

Service Inventory API Conformance Profile

TMF638B
Release 18.0.0
February 2018

Release: TM Forum Release 18.0	Member Evaluation
Version: 1.0.0	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2018. 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
List of Tables.....	4
Introduction	5
API DESCRIPTION.....	6
RESOURCE MODEL CONFORMANCE	7
Service Inventory API MANDATORY AND OPTIONAL RESOURCES	7
Service Inventory resource MANDATORY AND OPTIONAL ATTRIBUTES	7
API OPERATIONS CONFORMANCE	9
Service Inventory MANDATORY AND OPTIONAL OPERATIONS	9
API GET FILTERING OPERATION CONFORMANCE	10
Filtering in Service Inventory resource	10
GET /serviceInventory/service.....	12
GET /serviceInventory/service/{serviceId}	12
API CONFORMANCE TEST SCENARIOS	13
Service Inventory resource TEST CASES.....	13
Acknowledgements.....	18
Release History.....	18
Contributors to Document	18

LIST OF TABLES

N/A.

INTRODUCTION

The following document is the REST API Conformance for the Service Inventory API.

API DESCRIPTION

The intent of this API is to provide a consistent/standardized mechanism to query and manipulate the Service inventory.

This API allows the following operations

- Retrieve a list of Service stored in a server filtered by a given criteria
- Retrieve a specific Service in the inventory

RESOURCE MODEL CONFORMANCE

Service Inventory API MANDATORY AND OPTIONAL RESOURCES

Resource Name	Mandatory or Optional	Comments
Service	M	Only GET operation

Because POST, PATCH and DELETE operation are only for admin user in very limited UC there are no mandatory for this API

Service Inventory resource MANDATORY AND OPTIONAL ATTRIBUTES

The following table list service attribute rule for the GET response (attribute stated as mandatory must be available in GET response)

Attribute Name	Mandatory or Optional	Comments
id	M	
href	M	
name	M	
description	M	
type	O	
state	M	
category	M	
isServiceEnabled	O	
hasStarted	O	
startMode	O	
isStateful	O	
serviceDate	O	
startDate	M	
endDate	O	
@type	O	
@schemaLocation	O	
@baseType	O	
serviceOrder	O	Array of serviceOrder

Attribute Name		Mandatory or Optional	Comments
serviceRelationship		M	Array of service relationship(s)
	type	M	
	serviceRef	M	id and href are mandatory
serviceCharacteristic		M	Array of service characteristic
	name	O	
	valueType	M	
	value	M	
supportingService		M	Array of supporting service
	id	M	
	href	M	
	name	O	
	category	O	
supportingResource		M	Array of supporting resource
	id	M	
	href	M	
	role	O	
	name	O	
	category	O	
serviceSpecification		M	A service specification
	id	M	
	href	M	
	name	O	
	version	M	
	targetServiceSchema	O	
note		O	Array of Note
relatedParty		M	Array of related party
	id	M	
	href	M	
	role	M	
	name	O	
	validFor	O	
place		M	Array of place
	id	M	
	href	M	
	name	O	
	role	O	

API OPERATIONS CONFORMANCE

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

Service Inventory MANDATORY AND OPTIONAL OPERATIONS

Uniform API Operation	Mandatory/Optional	Comments
GET	M	GET must be used to retrieve a representation of a resource

API GET FILTERING OPERATION CONFORMANCE

Definitions

Filtered Search: A filtered search can be applied using query parameters in order to obtain only the resource entities that meet the criteria defined by the filtering parameters included in the query request. Several elements can be applied to the filtered search. In that case logic, a logical AND is applied to combine the criteria (e.g.:?name=<value> &owner.id=<value>)

Filtered Response Data (Response Attribute selection): In order to apply a filter and limit the number of attributes included in the response, the GET request can include the “?fields=” query parameter. Several elements can be applied to the filter. In that case, a logical AND is applied to combine the values (e.g.:?fields=name,description will provide in the response only the values assigned to attributes name and description). Attribute selection capabilities are the same for collections retrieval and individual resource queries

Filtering in Service Inventory resource

Attribute name	Filtered search	Filtered search	Response Attribute Selection	Response Attribute Selection
	First Level	N Level	First Level	N Level
id	M	M	M	M
href	NA	NA	O	O
name	O	NA	M	NA
description	O	NA	M	NA
type	M	NA	M	NA
state	M	NA	M	NA
category	O	NA	O	NA
isServiceEnabled	O	NA	O	NA
hasStarted	O	NA	O	NA
startMode	O	NA	O	NA
isStateful	O	NA	O	NA
serviceDate	O	NA	O	NA
startDate	O	NA	O	NA
endDate	O	NA	O	NA
serviceOrder				
id	O	NA	O	O
href	NA	NA	O	NA
serviceOrderItem	O	NA	O	NA
serviceRelationship				
type	O	NA	O	NA

Attribute name	Filtered search First Level	Filtered search N Level	Response Attribute Selection First Level	Response Attribute Selection N Level
serviceRef	O	NA	O	NA
characteristic				
name	O	NA	O	NA
valueType	O	NA	O	NA
value	O	NA	O	NA
supportingService				
id	O	NA	O	NA
href	NA	NA	O	NA
name	O	NA	O	NA
category	O	NA	O	NA
supportingResource				
id	O	NA	O	NA
role	O	NA	O	NA
href	NA	NA	O	NA
name	O	NA	O	NA
serviceSpecification				
id	O	NA	O	NA
href	NA	NA	NA	NA
version	O	NA	O	NA
name	O	NA	O	NA
note				
date	NA	NA	NA	NA
author	NA	NA	NA	NA
text	NA	NA	NA	NA
relatedParty				
id	O	NA	O	O
href	NA	NA	NA	NA
role	O	NA	O	O
name	O	NA	O	NA
validFor	NA	NA	NA	NA
place				
id	O	NA	O	NA
href	NA	NA	NA	NA
role	O	NA	O	NA
name	O	NA	O	NA

GET /serviceInventory/service

Providing filtering criteria is mandatory to avoid too many answers retrieved. See example below:

- description
- relatedParty.id
- category + state
- supportingResource.id
- ...

Filtered Response Data: A filtered response can be requested for the attributes using the “?fields=” query parameter

- Any of the attributes in the first level of ServiceInventory resource definition

GET /serviceInventory/service/{serviceId}

Filtered Response Data: A filtered response can be requested for the following attributes using the “?fields=” query parameter

- Any of the attributes in the first level of Service Inventory resource definition

API CONFORMANCE TEST SCENARIOS

This section describes the test scenarios required for the basic CONNECT certification of Service Inventory API.

Test Cases must be executed in the order defined for each resource because the result from one of the scenarios will be input for the next one.

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as **{serverRoot}/location**, being {serverRoot} the certification endpoint

Service Inventory resource TEST CASES

Nominal Scenarios

TC_ServiceInventory_N1 – Retrieve a service from its id

This test case needs to have a service pre-existing already on the server side

```
{
  "id": "56",
  "href": "http://.../service/56",
  "name": "NiceService",
  "description": "Service inventory sample",
  "@type": "vCPE",
  "state": "Active",
  "type": "CFS",
  "category": "Cloud",
  "isServiceEnabled": true,
  "hasStarted": true,
  "startMode": "1",
  "isStateful": true,
  "serviceDate": "2018-01-15T12:26:11.747Z",
  "startDate": "2018-01-15T12:26:11.747Z",
  "serviceSpecification": {
    "id": "12",
    "href": "http://...:serviceSpecification/12",
    "name": "vCPE",
    "version": "1",
    "@type": "vCPE",
    "@schemaLocation": "http..."
  },
  "serviceCharacteristic": [
    {
      "name": "vCPE_IP",
      "valueType": "String",
      "value": {
```

```

    "@type":"IPAddress",
    "@schemaLocation":"http....",
    "vCPE_IP":"193.218.289.11"
  }
}
],
"serviceRelationship":[
  {
    "type":"ReliesOn",
    "service":{
      "id":"49",
      "href":"http://.../service/49"
    }
  }
]
}

```

- Send a GET message to `{apiRoot}/serviceInventory/service/56`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one service resource referring to {56}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {56} matches the values described above

TC_ServiceInventory_N2 – Search for Service with specific characteristics

This test case needs to have another service pre-existing already on the server side:

```

{
  "id":"11",
  "href":"http://.../service/11",
  "name":"NiceServiceToo",
  "description":"Service inventory sample",
  "@type":"vCPE",
  "state":"feasibilityChecked",
  "type":"CFS",
  "category":"Cloud",
  "serviceDate":"2018-01-15T12:26:11.747Z",
  "startDate":"2018-01-15T12:26:11.747Z",
  "serviceSpecification":{

```

```

    "id": "12",
    "href": "http://...:serviceSpecification/12",
    "name": "vCPE"
  },
  "serviceCharacteristic": [
    {
      "name": "vCPE_IP",
      "valueType": "String",
      "value": {
        "@type": "IPAddress",
        "@schemaLocation": "http...",
        "vCPE_IP": "193.218.236.21"
      }
    }
  ],
  "serviceRelationship": [
    {
      "type": "ReliesOn",
      "service": {
        "id": "19",
        "href": "http://.../service/19"
      }
    }
  ],
  "supportingService": [
    {
      "id": "17",
      "href": "http://.../service/17",
      "name": "supporting service vxwn",
      "@referredType": "vxwn"
    }
  ]
}

```

- Send a GET message to `/{apiRoot}/serviceInventory/service/?category=Cloud`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes two service resources, referring to {56} and {11}
 - The body of the response for the resource with each identifier matches the values in the corresponding original request
- Send a GET message to `/{apiRoot}/serviceInventory/service/?state=Active`
- Wait for a response from the server with the following characteristics

- Response Code 200-OK
 - The body of the response includes one service resource, referring to {56}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {56} matches the values in the original request
- Send a GET message to `/{apiRoot}/serviceInventory/service/?description=NiceServiceToo`
 - Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one service resource, referring to {11}
 - The response message includes all mandatory parameters
 - The body of the response for the resource with identifier {11} matches the values in the original request

TC_ServiceInventory_N3 – Filtered retrieval of service data

- Send a GET message to `/{apiRoot}/serviceInventory/service/{11}?fields=name,state`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one service resource, referring to {11} and including only attributes *name* and *state*, matching the values in the original request

Notice that this test case is using parameters "name" and "state" to filter the data included in the response but any other parameter could be used

TC_ServiceInventory_N4 – Filtered Search and Filtered data response

- Send a GET message to `/{apiRoot}/serviceInventory/service//?state=Active&fields=id,href`
- Wait for a response from the server with the following characteristics
 - Response Code 200-OK
 - The body of the response includes one service resource, referring to {56}

- The body of the response for the resource with each identifier includes only attributes *id* and *href*, matching the values in the corresponding original request

Notice that this test case is using parameters "id" and "href" to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_ServiceInventory_E1 – Unknown Service

- Send a GET message to `{apiRoot}/serviceInventory/service/{8}`, where {8} does not match any of the identifiers previously created in the server
- Wait for a response from the server with the following characteristics
 - Response Code 404-Not Found

ACKNOWLEDGEMENTS

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 18.0.0 Version 1.0.0	15-Jan-2018	Ludovic Robert	First Release of Draft version of the Document

CONTRIBUTORS TO DOCUMENT

Ludovic Robert	Orange
----------------	--------