



Frameworkx Specification

Service Test Management API REST Specification

TMF653
Release 16.5.1
April 2017

Latest Update: Frameworkx Release 16.5	TM Forum Approved
Version 1.0.1	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.

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

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

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

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE	2
Table of Contents.....	4
List of FIGURES	6
Introduction	7
SAMPLE USE CASES	8
Service Test for Customer Experience Assurance.....	8
Planned Service Test for Importance Service	8
RESOURCE MODEL.....	9
Managed Entity and Task Resource Models	9
Service Test Specification Resource	9
Services Test Resource.....	13
<i>Notification Resource Models</i>	19
Service Test Specification Create Notification	20
Service Test Specification Attribute Value Change Notification.....	21
Service Test Specification Remove Notification	21
Service Test Creation Notification.....	21
Service Test Attribute Value Change Notification.....	22
Service Test RemOVE Notification	22
API OPERATIONS.....	23
Operations on Service Test Specification	23
List service test specifications.....	23
Retrieve service test specification.....	25
Create service test specification	26
Patch service test specification.....	28
Delete service test specification.....	29
Operations on Service Test	29

List service tests.....	30
Retrieve service test.....	31
Create service test.....	32
Patch service test	34
Delete service test	35
API NOTIFICATIONS.....	36
Register listener	36
Unregister listener	36
Publish Event to listener	37
Acknowledgements.....	39
Release History	39

LIST OF FIGURES

Fig.1: Service Test Specification Resource	9
Fig.2: Service Test Resource	14
Fig. 3: State machine diagram for a service test	18

INTRODUCTION

The following document is the specification of the REST API for Service Test Management. It includes the model definition as well as all available operations. Possible actions are creating, updating and retrieving Service Test.

The Service Test API provides a standardized mechanism for placing a service test with all of the necessary test parameters. The API consists of a simple set of operations that interact with CRM/Service Management systems in a consistent manner. A service test is a procedure intended to check the quality, performance, or reliability of a service.

Service Test Management API manages service test resource and service specification test resource:

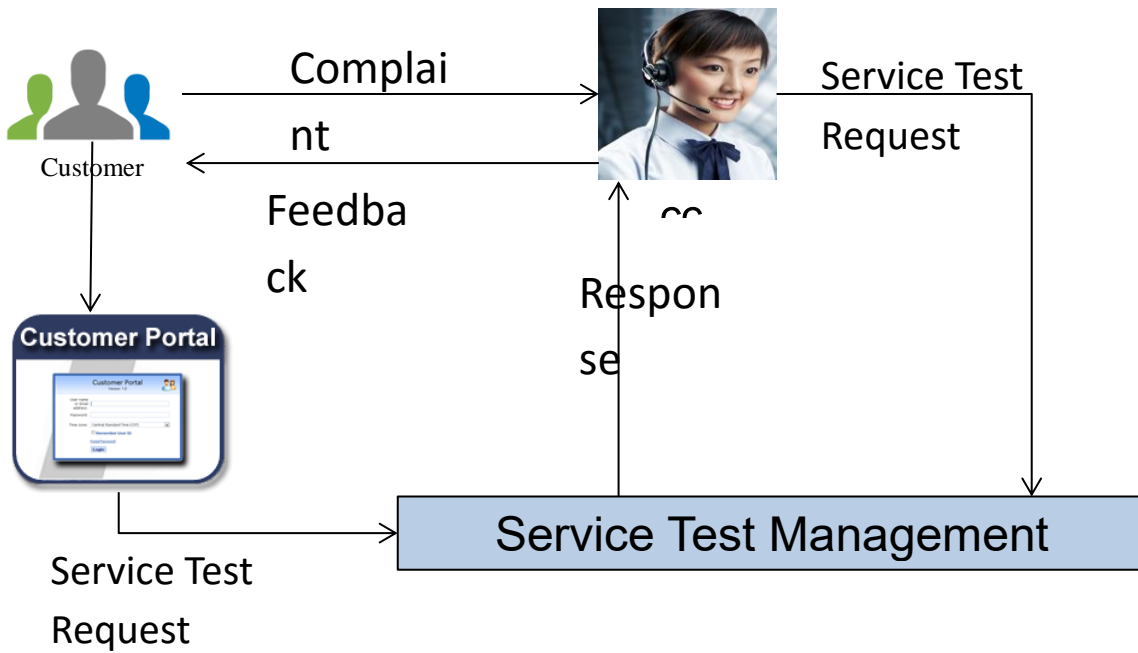
- A service test specification describes the service test in terms of parameters to be configured and measures to be taken.
- A service exists for a controlled test invocation on a service. The service test is executed according to a schedule, and contains service test configuration parameters that are to be applied at execution time, and service test measures that result.

SAMPLE USE CASES

SERVICE TEST FOR CUSTOMER EXPERIENCE ASSURANCE

The Service Test can be used for customer experience assurance while:

- The subscriber can launch service test from Self Service Portal to make sure the service quality while his/her experience is not good.
- The Call Center operator can execute service test to check the quality and analyze root cause for response while one customer has complaint



It could improve customer experience and increase FCR(First Call Resolution) much.

PLANNED SERVICE TEST FOR IMPORTANCE SERVICE

The test job can be defined for some important services that the test will be executed against them periodically to assurance the service quality and find potential service downgrading in advance to avoid massive impact to customers.

RESOURCE MODEL

MANAGED ENTITY AND TASK RESOURCE MODELS

SERVICE TEST SPECIFICATION RESOURCE

A service test specification describes the service test in terms of parameters to be configured and measures to be taken.

Resource Model

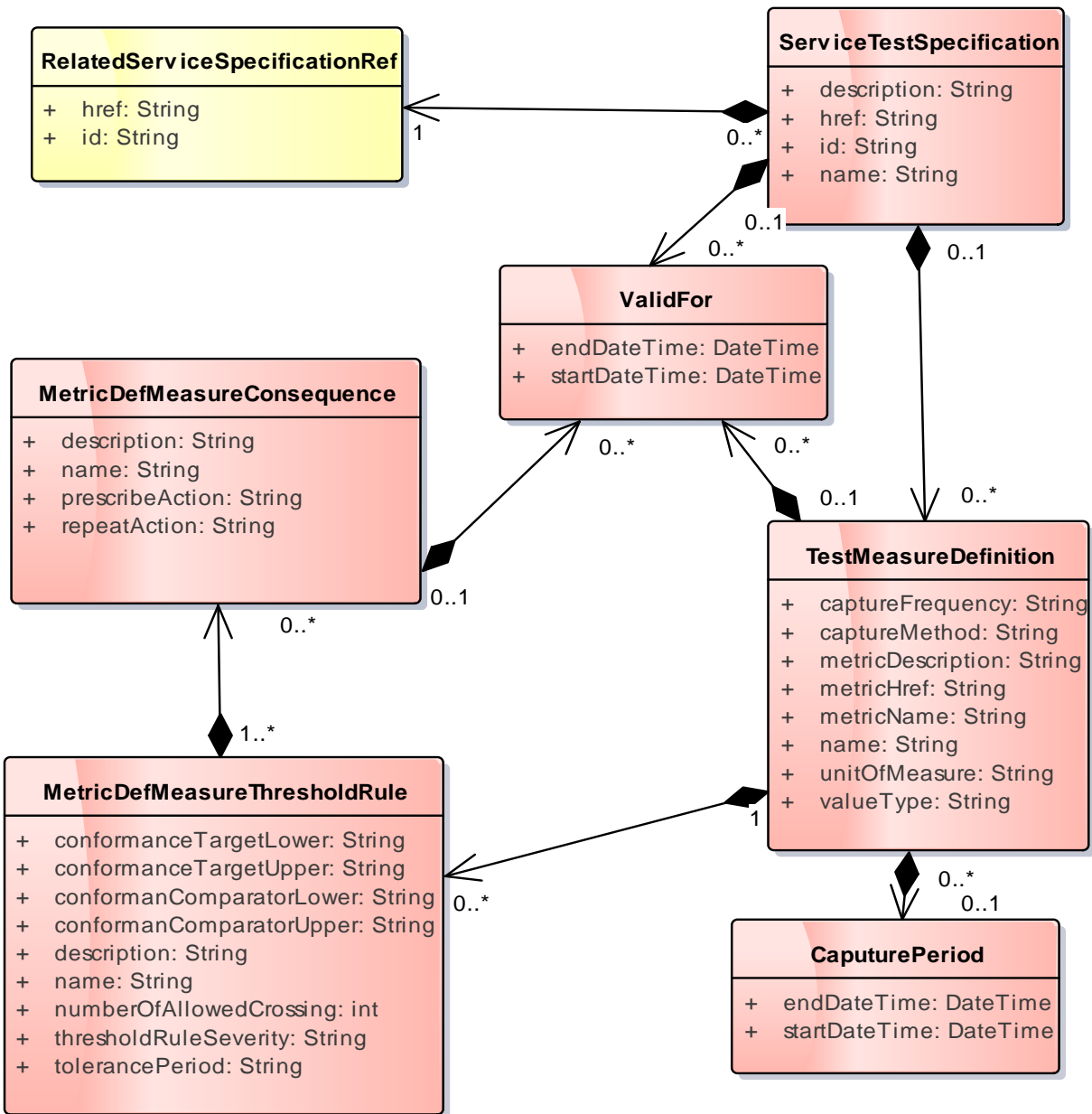


Fig.1: Service Test Specification Resource

Field descriptions

CapturePeriod fields

A period of the capture.

endTime	A DateTime. The end date and time.
startTime	A DateTime. The start date and time

MetricDefMeasureConsequence fields

A MetricDefMeasureConsequence defines the action (prescribed action or notification) to take when a MetricDefMeasureThresholdRule is crossed.

description	A String. A narrative that explains in detail what the consequence is.
name	A String. A word, term, or phrase by which a MetricDefMeasureConsequence is known and distinguished from other MetricDefMeasureConsequences.
prescribeAction	A String. Recommended remedy for a violated threshold. This could be the hyperlink to the action.
repeatAction	A String. An indicator use to specify that a consequence should cease being applied if a value is in the same range as the previous value or continue being applied if a value is in the same range as the previous value. If the repeatAction is True, if the consequence is always applied as soon as the MetricMeasure value is in the range of values and if the repeatAction is False, the consequence is applied only if the previous MetricMeasure value wasn't in the same range.
validFor	A period of a thing.

MetricDefMeasureThresholdRule fields

A MetricDefMeasureThresholdRule is a rule that defines the condition (raise or clear) to achieve to apply consequences when a threshold is crossed or ceased to be crossed. It also defines the severity of the raise or clear of the threshold.

conformanceTargetLower	A String. A value used to determine if the threshold is crossed or ceases to be crossed. It represents the lower limit. The value should be less than the conformanceTargetUpper. The conformance comparators should also be logically defined so as to not lead to a logically impossible condition.
conformanceTargetUpper	A String. A value used to determine if the threshold is crossed or ceases to be crossed. It represents the Upper limit. The value should be greater than the conformanceTargetLower. The conformance comparators should also be logically defined so as to not lead to a logically impossible condition.
conformanceComparatorLower	A String. An operator that when applied on a value specifies whether a threshold is crossed or ceased to be crossed. This operator is used to compare with the conformanceTargetLower.
conformanceComparatorUpper	A String. An operator that when applied on a value specifies whether a threshold is crossed or ceased to be crossed. This operator is used to

	compare with the conformanceTargetUpper.
description	A String. Description for the MetricDefMeasureThresholdRule .
name	A String. Name for the MetricDefMeasureThresholdRule .
numberOfAllowedCrossing	A int. The number of allowed crossing occurrences in reference to the tolerancePeriod without a consequence being initiated.
thresholdRuleSeverity	A String. A threshold can be generated in different severity levels. A crossing for each level may require a different condition and possibly trigger a different consequence.
tolerancePeriod	A String. An interval of time of allowed crossing occurrences before a consequence being initiated.
metricDefMeasureConsequence	A MetricDefMeasureConsequence defines the action (prescribed action or notification) to take when a MetricDefMeasureThresholdRule is crossed.

RelatedServiceSpecificationRef fields

The related service specification.

href	A String. Hyperlink to access service specification.
id	A String. The identifier of a service specification.

ServiceTestSpecification fields

The service test specification describes the service test in terms of parameters to be configured and measures to be taken.

description	A String. Description of a service test specification.
href	A String. Hyperlink to access a service test specification.
id	A String. Identifier of a service test specification.
name	A String. Name of a service test specification.
validFor	A period of a thing.
testMeasureDefinition	A TestMeasureDefinition specifies a measure of a specific aspect of a product, service, or resource test, such as lost packets or connectivity status.
relatedServiceSpecificationRef	The related service specification.

TestMeasureDefinition fields

A TestMeasureDefinition specifies a measure of a specific aspect of a product, service, or resource test, such as lost packets or connectivity status.

captureFrequency	A String. The frequency of capture for the metric. Note: This may be replaced by a set of entities similar to the Performance Monitoring.
captureMethod	A String. The method used to capture the Metric. Note: This may be replaced by a set of entities similar to the Performance Monitoring.
metricDescription	A String. Brief description of the metric.
metricHref	A String. Hyperlink to access a metric for detail information.
metricName	A String. The name of a metric that in the test measure.
name	A String. The name of the TestMeasureDefinition.

unitOfMeasure	A String. Name of a service test specification.
valueType	A String. A kind of value that the Metric value can take on, such as numeric, text, and so forth.
capturePeriod	A period of the capture.
metricDefMeasureThresholdRule	A MetricDefMeasureThresholdRule is a rule that defines the condition (raise or clear) to achieve to apply consequences when a threshold is crossed or ceased to be crossed. It also defines the severity of the raise or clear of the threshold.
validFor	A period of a thing.

ValidFor fields

A period of a thing.

endTime	A DateTime. The end date and time.
startTime	A DateTime. The start date and time

JSON representation sample

We provide below the JSON representation of an example of Service Test Specification Resource object:

```
{
  "description": "a service description",
  "href": "https://host:port/serviceTestManagement/serviceTestSpecification/12",
  "id": "12",
  "name": "monkey test",
  "validFor": {
    "endTime": "2017-03-01T00:00:00",
    "startTime": "2016-03-01T00:00:00"
  },
  "testMeasureDefinition": [
    {
      "captureFrequency": "once",
      "captureMethod": "automatic capture by instrument",
      "metricDescription": "This is the flow speed in a service",
      "metricHref": "https://host:port/metricManagement/metric/21",
      "metricName": "Flow speed",
      "name": "Definition of Flow speed",
      "unitOfMeasure": "A second",
      "valueType": "bits",
      "capturePeriod": {
        "endTime": "T00:00:00",
        "startTime": "T00:00:00"
      },
      "metricDefMeasureThresholdRule": [
        {
          "conformanceTargetUpper": "2500",
          "conformanceComparatorUpper": "greater than ",
          "description": "This speed is grater than the target upper",
          "name": "tooMuchTraffic",
          "numberOfAllowedCrossing": "0",
          "thresholdRuleSeverity": "1",

```

```
"tolerancePeriod": "15 minutes",
"metricDefMeasureConsequence": [
  {
    "appliedAction": "sendWarning",
    "description": "send a warning to the system",
    "name": "sendWarning",
    "repeatAction": "False",
    "validFor": {
      "endDateTime": "2017-03-01T00:00:00",
      "startDateTime": "2016-03-01T00:00:00"
    }
  },
  {
    "appliedAction": "generateReport",
    "description": "generate a warning report ",
    "name": "generateReport",
    "repeatAction": "False",
    "validFor": {
      "endDateTime": "2017-03-01T00:00:00",
      "startDateTime": "2016-03-01T00:00:00"
    }
  }
]
},
"validFor": {
  "endDateTime": "2017-03-01T00:00:00",
  "startDateTime": "2016-03-01T00:00:00"
}
},
"relatedServiceSpecificationRef": {
  "href": "https://host:port/serviceManagement/serviceSpecification/31",
  "id": "31",
}
}
```

SERVICES TEST RESOURCE

A service test is an entity that exists that exists for a controlled test invocation on a service. The service test is executed according to a schedule. and contains service test configuration parameters that are to be applied at execution time, and service test measures that result.

Resource Model

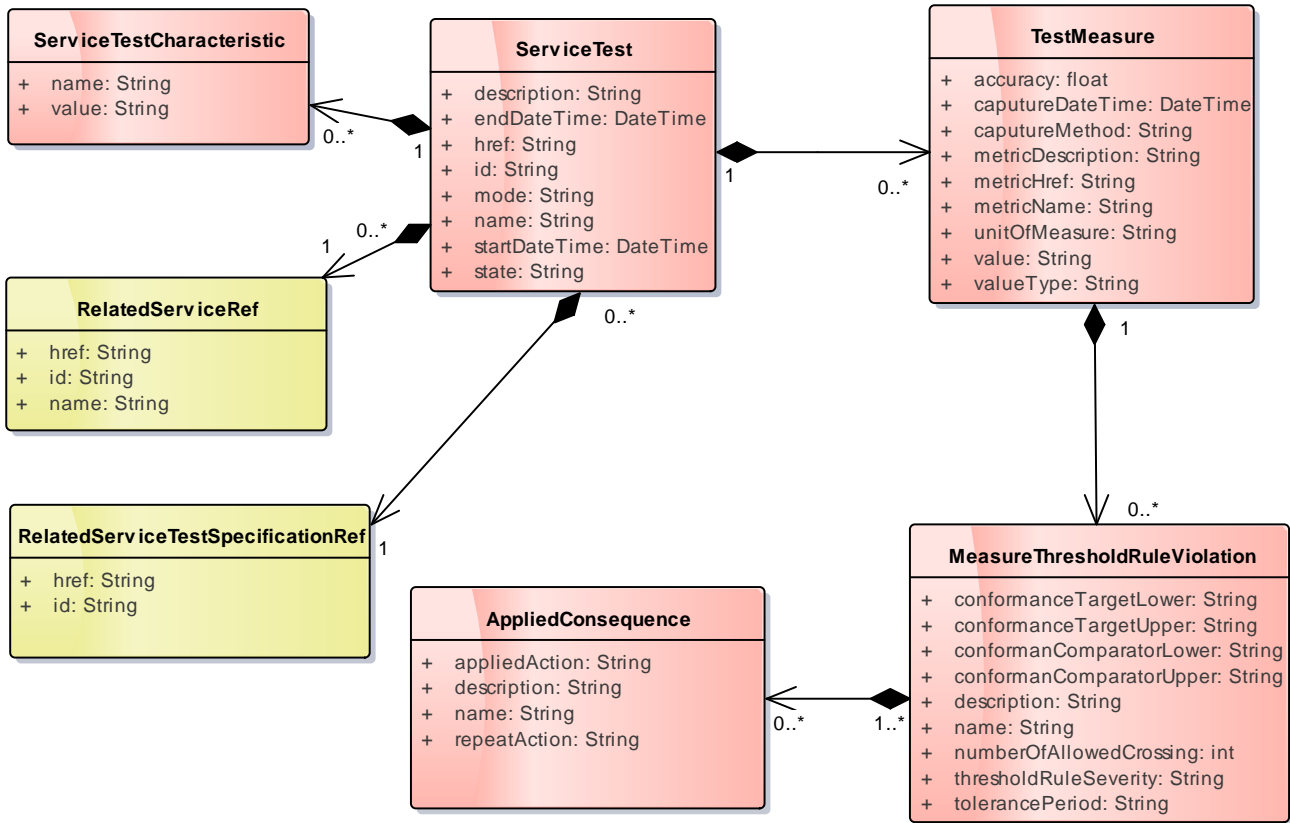


Fig.2: Service Test Resource

Field descriptions

AppliedConsequence fields

An Applied Consequence defines the action (prescribed action or notification) to take when a MeasureThresholdRuleViolation occurs.

appliedAction	A String. The action for a violated threshold. This could be a hyperlink to the action.
description	A String. A narrative that explains in detail what the consequence is.
name	A String. A word, term, or phrase by which Consequence is known and distinguished from other MetricDefMeasureConsequences.
repeatAction	A String. An indicator use to specify that a consequence should cease being applied if a value is in the same range as the previous value or continue being applied if a value is in the same range as the previous value. If the repeatAcation is True, if the consequence is always applied as soon as the MetricMeasure value is in the range of values and if the repeatAcation is False, the consequence is applied only if the previous MetricMeasure value wasn't in the same range.

MeasureThresholdRuleViolation fields

A `measureThresholdRuleViolation` is a violation of a rule that defines the in the `MetricDefMeasureThresholdRule`.

<code>conformanceTargetLower</code>	A String. A value used to determine if the threshold is crossed or ceases to be crossed. It represents the lower limit. The value should be less than the <code>conformanceTargetUpper</code> . The conformance comparators should also be logically defined so as to not lead to a logically impossible condition.
<code>conformanceTargetUpper</code>	A String. A value used to determine if the threshold is crossed or ceases to be crossed. It represents the Upper limit. The value should be greater than the <code>conformanceTargetLower</code> . The conformance comparators should also be logically defined so as to not lead to a logically impossible condition.
<code>conformanceComparatorLower</code>	A String. An operator that when applied on a value specifies whether a threshold is crossed or ceased to be crossed. This operator is used to compare with the <code>conformanceTargetLower</code> .
<code>conformanceComparatorUpper</code>	A String. An operator that when applied on a value specifies whether a threshold is crossed or ceased to be crossed. This operator is used to compare with the <code>conformanceTargetUpper</code> .
<code>description</code>	A String. Description for the <code>MetricDefMeasureThresholdRule</code> .
<code>name</code>	A String. Name for the <code>MetricDefMeasureThresholdRule</code> .
<code>numberOfAllowedCrossing</code>	A int. The number of allowed crossing occurrences in reference to the <code>tolerancePeriod</code> without a consequence being initiated.
<code>thresholdRuleSeverity</code>	A String. A threshold can be generated in different severity levels. A crossing for each level may require a different condition and possibly trigger a different consequence.
<code>tolerancePeriod</code>	A String. An interval of time of allowed crossing occurrences before a consequence being initiated.
<code>appliedConsequence</code>	An Applied Consequence defines the action (prescribed action or notification) to take when a <code>MeasureThresholdRuleViolation</code> occurs.

RelatedServiceRef fields

The service of the service test

<code>href</code>	A String. Hyperlink to access a service.
<code>id</code>	A String. Identifier of a service.
<code>name</code>	A String. A name to describe a service.

RelatedServiceTestSpecificationRef fields

The service test specification used by the service test.

<code>href</code>	A String. Hyperlink to access a service test specification.
<code>id</code>	A String. Identifier of a service test specification.

ServiceTest fields

A service test is an entity that exists that exists for a controlled test invocation on a service. The service test is executed according to a schedule and contains service test configuration parameters that are to be applied at execution time, and service test measures that result.

description	A String. Description of the service test
endTime	A DateTime. The end date and time of the service test.
href	A String. Hyperlink to access the service test
id	A String. The identifier of the service test
mode	A String. An indication of whether the service test is running in "PROACTIVE" or "ONDEMAND" mode
name	A String. The name of the service test
startTime	A DateTime. The start date and time of the service test.
state	A String. The actual state the service test is in
relatedServiceTestSpecificationRef	The service test specification used by the service test.
serviceTestCharacteristic	The configuration of the service test is set by the service test characteristic.
relatedServiceRef	The service of the service test
testMeasure	A TestMeasureDefinition specifies a measure of a specific aspect of a product, service, or resource test, such as lost packets or connectivity status.

ServiceTestCharacteristic fields

The configuration of the service test is set by the service test characteristic.

name	A String. The name of the characteristic.
value	A String. The value of characteristic.

TestMeasure fields

A TestMeasureDefinition specifies a measure of a specific aspect of a product, service, or resource test, such as lost packets or connectivity status.

accuracy	A float. The number of digits of accuracy captured for Metrics associated
captureDateTime	A DateTime. The date and time that capture the metric.
captureMethod	A String. "The method used to capture the Metrics. Note: This may be replaced by a set of entities similar to the Performance Monitoring Ref."
metricDescription	A String. Brief description of the metric.
metricHref	A String. Hyperlink to access a metric for detail information.
metricName	A String. The name of metric.
unitOfMeasure	A String. The unit of measure for the metric values, such as meters, cubic yards, kilograms [ISO 1000].
value	A String. The value of Metric in the test. This also could be a documents to record all values got during the service test.
valueType	A String. A kind of value that the Metric value associated with a metricValue can take on, such as numeric, text, and so forth.
measureThresholdRuleViolation	A measureThresholdRuleViolation is a violation of a rule that defines the in the MericDefMeasureThresholdRule.

JSON representation sample

We provide below the JSON representation of an example of Service Test Resource object:

```
{
  "href": "https://host:port/serviceTestManagement/serviceTest/3112",
  "id": "3112",
  "description": "This service test is a simple one",
  "endTime": "2016-03-02T24:00:00",
  "mode": "PROACTIVE",
  "startTime": "2016-03-02T00:00:00",
  "state": "Completed",
  "relatedServiceRef": {
    "href": "https://host:port/serviceManagement/service/21",
    "id": "21",
    "name": "gameService"
  },
  "relatedServiceTestSpecificationRef": {
    "href": "https://host:port/serviceTestManagement/serviceTest/22",
    "id": "22"
  },
  "serviceTestCharacteristic": [
    {
      "name": "number",
      "value": "3"
    },
    {
      "name": "restart",
      "value": "True"
    }
  ],
  "testMeasure": [
    {
      "accuracy": 0.0001,
      "captureDateTime": "2016-03-02T11:12:00",
      "captureMethod": "automatic capture by instrument",
      "metricDescription": "This is the flow speed in a service",
      "metricHref": "https://host:port/metricManagement/metric/21",
      "metricName": "Flow speed",
      "unitOfMeasure": "A second",
      "value": "3000",
      "valueType": "bits",
      "measureThresholdRuleViolation": [
        {
          "conformanceTargetUpper": "2500",
          "conformanceComparatorUpper": "greater than ",
          "description": "This speed is greater than the target upper",
          "name": "tooMuchTraffic",
          "numberOfAllowedCrossing": "0",
          "thresholdRuleSeverity": "1",
          "appliedConsequence": [
            {
              "appliedAction": "sendWarning",
              "description": "send a warning to the system",
              "name": "sendWarning",
              "repeatAction": "False"
            }
          ]
        }
      ]
    }
  ]
}
```

```

{
  "appliedAction": "generateReport",
  "description": "generate a waring report ",
  "name": "generateReport",
  "repeatAction": "False"
}
]
}
]
}
}
}
}
}

```

Here is the state machine diagram for a service test. Each test states are described in the tab below.

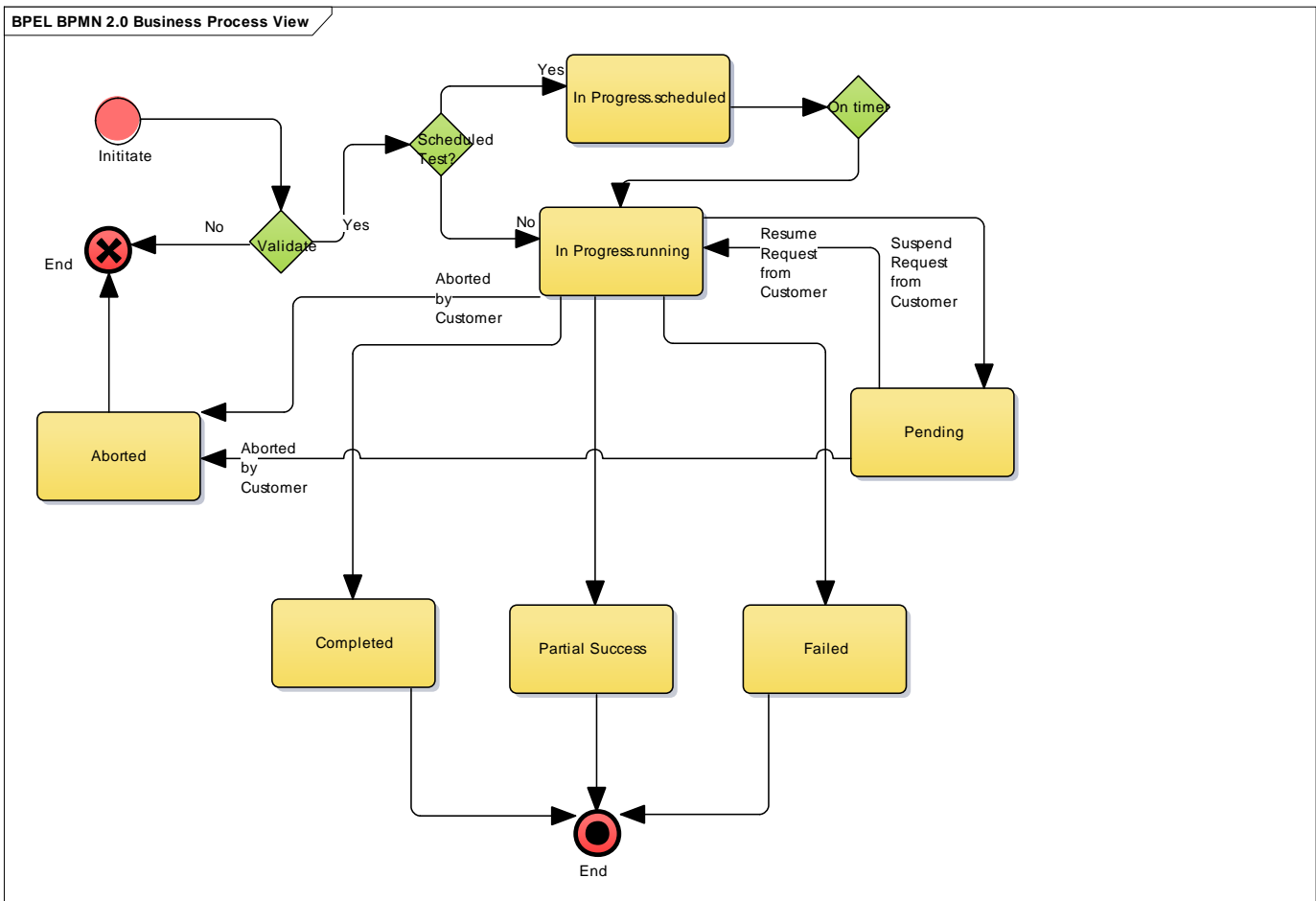


Fig. 3: State machine diagram for a service test

In Progress	The <i>In Progress</i> state is where a test has been started.
Aborted	The <i>Aborted</i> state is where a test execution has been successfully cancelled.
Completed	The <i>Completed</i> state is where the execution of test has been finished without error.

Pending	<p>The <i>Pending</i> state is used when the customer needs to suspend the test execution or some more information needs to be provided.</p> <p>A pending stage can lead into auto cancellation of a test, if no action is taken within the defined timeframes to be described under the Agreement.</p>
Failed	The execution of a test has <i>Failed</i> .
Partial Success	Some test items have failed and some have succeeded so the entire test is in a <i>Partial Success</i> state.

NOTIFICATION RESOURCE MODELS

6 notifications are defined for this API

Notifications related to ServiceTestSpecification:

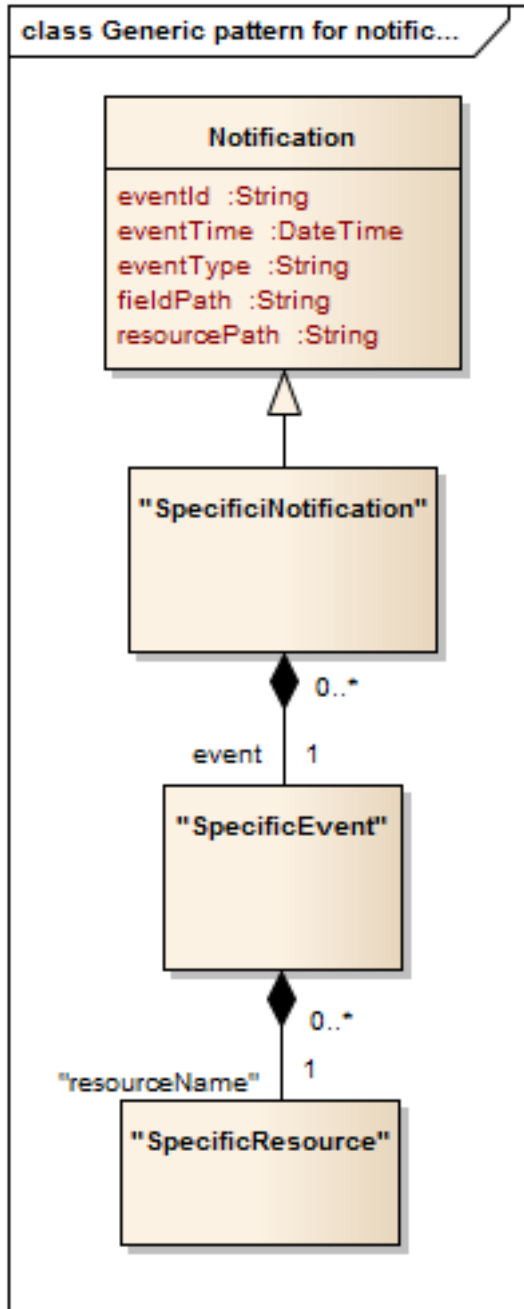
- ServiceTestSpecificationCreationNotification
- ServiceTestSpecificationAttributeValueChangeNotification
- ServiceTestSpecificationRemoveNotification

Notifications related to ServiceTest:

- ServiceTestCreationNotification
- ServiceTestAttributeValueChangeNotification
- ServiceTestRemoveNotification

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 TEST SPECIFICATION CREATE NOTIFICATION

Notification sent when a new ServiceTestSpecification resource is created.

Json representation sample

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

```

{
  "eventId":"00001",
  "eventTime":"2016-03-01T16:42:25-45:00",
  "eventType":"ServiceTestSpecificationCreationNotification",
  "event": {

```

```

    "ServiceTestSpecification" :
      {-- SEE ServiceTestSpecification RESOURCE SAMPLE --}
    }
  }

```

SERVICE TEST SPECIFICATION ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when removing a ServiceTestSpecification resource.

Json representation sample

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

```

{
  "eventId":"00002",
  "eventTime":"2016-03-01T16:48:00",
  "eventType":" ServiceTestSpecificationAttributeChangeNotification ",
  "event": {
    "ServiceTestSpecification" :
      {-- SEE ServiceTestSpecification RESOURCE SAMPLE --}
  }
}

```

SERVICE TEST SPECIFICATION REMOVE NOTIFICATION

Notification sent when removing a ServiceTestSpecification resource.

Json representation sample

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

```

{
  "eventId":"00003",
  "eventTime":"2016-03-05T16:48:00",
  "eventType":"ServiceTestSpecificationRemoveNotification",
  "event": {
    "ServiceTestSpecification" :
      {-- SEE ServiceTestSpecification RESOURCE SAMPLE --}
  }
}

```

SERVICE TEST CREATION NOTIFICATION

Notification sent when a new ServiceTest resource is created.

Json representation sample

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

```
{
  "eventId": "00004",
  "eventTime": "2016-03-01T16:42:25",
  "eventType": "ServiceTestCreationNotification",
  "event": {
    "serviceTest":
      {-- SEE ServiceTest RESOURCE SAMPLE --}
  }
}
```

SERVICE TEST ATTRIBUTE VALUE CHANGE NOTIFICATION

Notification sent when changing an attribute of a ServiceTest resource.

Json representation sample

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

```
{
  "eventId": "00005",
  "eventTime": "2016-03-02T16:42:25",
  "eventType": "ServiceTest AttributeValueChangeNotification",
  "event": {
    "serviceTest":
      {-- SEE ServiceTest RESOURCE SAMPLE --}
  }
}
```

SERVICE TEST REMOVE NOTIFICATION

Notification sent when changing an attribute of a ServiceTest resource.

Json representation sample

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

```
{
  "eventId": "00005",
  "eventTime": "2016-03-02T16:42:25",
  "eventType": "ServiceTestRemoveNotification",
  "event": {
    "serviceTest":
      {-- SEE ServiceTest 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 TEST SPECIFICATION

LIST SERVICE TEST SPECIFICATIONS

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

Description

This operation list service test 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 ServiceTestSpecification resources.

Request
<pre>GET /serviceTestManagement/serviceTestSpecification Accept: application/json</pre>
Response
<pre>200 [{ "description": "a service description", "href": "https://host:port/serviceTestManagement/serviceTestSpecification/12", "id": "12", "name": "monkey test", "validFor": { "endDateTime": "2017-03-01T00:00:00", "startDateTime": "2016-03-01T00:00:00" }, "testMeasureDefinition": [{ "captureFrequency": "once", "captureMethod": "automatic capture by instrument", "metricDescription": "This is the flow speed in a service", "metricHref": "https://host:port/metricManagement/metric/21", "metricName": "Flow speed", "name": "Definition of Flow speed", "unitOfMeasure": "A second", "valueType": "bits", "capturePeriod": { "endDateTime": "T00:00:00", "startDateTime": "T00:00:00" }, "metricDefMeasureThresholdRule": [{ "conformanceTargetUpper": "2500", "conformanComparatorUpper": "greater than ", "description": "This speed is greater than the target upper", "name": "tooMuchTraffic",</pre>


```

    "numberOfAllowedCrossing": "0",
    "thresholdRuleSeverity": "1",
    "tolerancePeriod": "15 minutes",
    "metricDefMeasureConsequence": [
      {
        "appliedAction": "sendWarning",
        "description": "send a warning to the system",
        "name": "sendWarning",
        "repeatAction": "False",
        "validFor": {
          "endDateTime": "2017-03-01T00:00:00",
          "startDateTime": "2016-03-01T00:00:00"
        }
      },
      {
        "appliedAction": "generateReport",
        "description": "generate a waring report ",
        "name": "generateReport",
        "repeatAction": "False",
        "validFor": {
          "endDateTime": "2017-03-01T00:00:00",
          "startDateTime": "2016-03-01T00:00:00"
        }
      }
    ]
  },
  "validFor": {
    "endDateTime": "2017-03-01T00:00:00",
    "startDateTime": "2016-03-01T00:00:00"
  }
},
"relatedServiceSpecificationRef": {
  "href": "https://host:port/serviceManagement/serviceSpecification/31",
  "id": "31",
}
}
]

```

RETRIEVE SERVICE TEST SPECIFICATION

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

Description

This operation retrieves a service test 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 ServiceTestSpecification resource.

Request
<pre>GET /serviceTestManagement/serviceTestSpecification/12?fields=description,href,id,name,validFor,relatedServiceSpecificationRef Accept: application/json</pre>
Response
<pre>200 { "description": "a service description", "href": "https://host:port/serviceTestManagement/serviceTestSpecification/12", "id": "12", "name": "monkey test", "validFor": { "endDateTime": "2017-03-01T00:00:00", "startDateTime": "2016-03-01T00:00:00" }, "relatedServiceSpecificationRef": { "href": "https://host:port/serviceManagement/serviceSpecification/31", "id": "31", } }</pre>

CREATE SERVICE TEST SPECIFICATION

POST /serviceTestSpecification

Note: this operation is available only to ADMIN API users

Description

This operation creates a service test specification entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceTestSpecification, including any possible rule conditions and applicable default values.

Mandatory Attributes	Rule
name	
relatedServiceSpecification	
id	Automatically generated
href	Automatically generated

Additional Rules

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

Context	Mandatory Sub-Attributes
RelatedServiceSpecificationRef	href, id
ValidFor	endDateTime, startDateTime
TestMeasureDefinition	metricName, metricHref, name
CaputurePeriod	endDateTime, startDateTime
MetircDefMeasureThresholdRule	conformanceTargetLower, conformanceTargetUpper, conformanComparatorLower, conformanComparatorUpper, name

Usage Samples

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

Request
POST /serviceTestManagement/ServiceTestSpecification Content-Type: application/json <pre>{ "description": "a service description", "name": "monkey test", "validFor": { "endDateTime": "2017-03-01T00:00:00", "startDateTime": "2016-03-01T00:00:00" }, "relatedServiceSpecificationRef": { "href": "https://host:port/serviceManagement/serviceSpecification/31", "id": "31", } }</pre>
Response
201 <pre>{ "description": "a service description", "href": "https://host:port/serviceTestManagement/serviceTestSpecification/13", "id": "13", "name": "monkey test", "validFor": {</pre>

```

    "endDateTime": "2017-03-01T00:00:00",
    "startDateTime": "2016-03-01T00:00:00"
  },
  "relatedServiceSpecificationRef": {
    "href": "https://host:port/serviceManagement/serviceSpecification/31",
    "id": "31",
  }
}

```

PATCH SERVICE TEST SPECIFICATION

PATCH /serviceTestSpecification/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a service test 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.

Non Patchable Attributes	Rule
href	
id	

Usage Samples

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

Request

```

PATCH /serviceTestManagement/ServiceTestSpecification/13
Content-Type: application/merge-patch+json

```

```

{
  "name": "new name"
}

```

Response

```

201
{
  "description": "a service description",
  "href": "https://host:port/serviceTestManagement/serviceTestSpecification/13",
  "id": "13",
  "name": "new name",
  "validFor": {
    "endTime": "2017-03-01T00:00:00",
    "startTime": "2016-03-01T00:00:00"
  },
  "relatedServiceSpecificationRef": {
    "href": "https://host:port/serviceManagement/serviceSpecification/31",
    "id": "31",
  }
}

```

DELETE SERVICE TEST SPECIFICATION**DELETE /serviceTestSpecification/{id}**

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service test specification type entity.

Usage Samples

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

Request

```
DELETE /serviceTestManagement/serviceTestSpecification/12
```

Response

```
204
```

OPERATIONS ON SERVICE TEST

LIST SERVICE TESTS

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

Description

This operation list service test 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 ServiceTest resources.

Request
<pre>GET /serviceTestManagement/serviceTest Accept: application/json</pre>
Response
<pre>200 [{ "href": "https://host:port/serviceTestManagement/serviceTest/3112", "id": "3112", "description": "This service test is a simple one", "endTime": "2016-03-02T24:00:00", "mode": "PROACTIVE", "startTime": "2016-03-02T00:00:00", "state": "Completed", "relatedServiceRef": { "href": "https://host:port/serviceManagement/service/21", "id": "21", "name": "gameService" }, "relatedServiceTestSpecificationRef": { "href": "https://host:port/serviceTestManagement/serviceTest/22", "id": "22" }, "serviceTestCharacteristic": [{ "name": "number", "value": "3" }, { "name": "restart", "value": "Ture" }] }],</pre>

```

"testMeasure": [
  {
    "accuracy": 0.0001,
    "captureDateTime": "2016-03-02T11:12:00",
    "captureMethod": "automatic capture by instrument",
    "metricDescription": "This is the flow speed in a service",
    "metricHref": "https://host:port/metricManagement/metric/21",
    "metricName": "Flow speed",
    "unitOfMeasure": "A second",
    "value": "3000",
    "valueType": "bits",
    "measureThresholdRuleViolation": [
      {
        "conformanceTargetUpper": "2500",
        "conformanComparatorUpper": "greater than ",
        "description": "This speed is greater than the target upper",
        "name": "tooMuchTraffic",
        "numberOfAllowedCrossing": "0",
        "thresholdRuleSeverity": "1",
        "appliedConsequence": [
          {
            "appliedAction": "sendWarning",
            "description": "send a warning to the system",
            "name": "sendWarning",
            "repeatAction": "False"
          },
          {
            "appliedAction": "generateReport",
            "description": "generate a waring report ",
            "name": "generateReport",
            "repeatAction": "False"
          }
        ]
      }
    ]
  }
]

```

RETRIEVE SERVICE TEST

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

Description

This operation retrieves a service test 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 ServiceTest resource.

Request
GET /serviceTestManagement/serviceTest/3112?fields=href,id,description Accept: application/json
Response
200 { "href": "https://host:port/serviceTestManagement/serviceTest/3112", "id": "3112", "description": "This service test is a simple one", }

CREATE SERVICE TEST

POST /serviceTest

Note: this operation is available only to ADMIN API users

Description

This operation creates a service test entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceTest, including any possible rule conditions and applicable default values.

Mandatory Attributes	Rule
name	
id	Automatically generated
href	Automatically generated
relatedService	
relatedServiceTestSpecification	

Additional Rules

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

Context	Mandatory Sub-Attributes
RelatedServiceRef	href, id
RelatedServiceTestSpecificationRef	href, id
ServiceTestCharacteristic	name, value

MeasureThresholdRuleViolation	name, conformanceTargetLower, conformanceTargetUpper, conformanComparatorLower, conformanComparatorUpper
TetMeasure	metricHref, metricName

Usage Samples

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

Request
<pre> POST /serviceTestManagement/serviceTest Content-Type: application/json { "description": "This service test is a simple one", "endDateTime": "2016-03-02T24:00:00", "mode": "PROACTIVE", "startDateTime": "2016-03-02T00:00:00", "state": " In Progress", "relatedServiceRef": { "href": "https://host:port/serviceManagement/service/21", "id": "21", "name": "gameService" }, "relatedServiceTestSpecificationRef": { "href": "https://host:port/serviceTestManagement/serviceTest/22", "id": "22" }, } </pre>
Response
<pre> 201 { "href": "https://host:port/serviceTestManagement/serviceTest/89757", "id": "89757", "description": "This service test is a simple one", "endDateTime": "2016-03-02T24:00:00", "mode": "PROACTIVE", "startDateTime": "2016-03-02T00:00:00", "state": "In Progress", "relatedServiceRef": { "href": "https://host:port/serviceManagement/service/21", "id": "21", "name": "gameService" }, "relatedServiceTestSpecificationRef": { "href": "https://host:port/serviceTestManagement/serviceTest/22", "id": "22" }, } </pre>

PATCH SERVICE TEST

PATCH /serviceTest/{id}

Description

This operation allows partial updates of a service test 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.

Non Patchable Attributes	Rule
id	
href	

Usage Samples

Here's an example of requests for patching a ServiceTest resource.

Changing the status to 'prospective' (using json-merge)

Request
PATCH /serviceTestManagement/serviceTest/42 Content-Type: application/merge-patch+json <pre>{ "status": "Completed" }</pre>
Response
201 <pre>{ "href": "https://host:port/serviceTestManagement/serviceTest/89757", "id": "89757", "description": "This service test is a simple one", "endTime": "2016-03-02T24:00:00", }</pre>

```

"mode": "PROACTIVE",
"startDateTime": "2016-03-02T00:00:00",
"state": "Completed",
"relatedServiceRef": {
  "href": "https://host:port/serviceManagement/service/21",
  "id": "21",
  "name": "gameService"
},
"relatedServiceTestSpecificationRef": {
  "href": "https://host:port/serviceTestManagement/serviceTest/22",
  "id": "22"
},
}

```

DELETE SERVICE TEST

DELETE /serviceTest/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a service test entity.

Usage Samples

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

Request
DELETE /serviceTestManagement/serviceTest/89757
Response

API NOTIFICATIONS

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

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

REGISTER LISTENER

POST /hub

Description

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

Behavior

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

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

Usage Samples

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

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

UNREGISTER LISTENER

DELETE /hub/{id}

Description

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

Behavior

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

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

Usage Samples

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

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

PUBLISH EVENT TO LISTENER

POST /client/listener

Description

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

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

Behavior

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

Usage Samples

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

Request
POST /client/listener

Accept: application/json

```
{
  "event": {
    EVENT BODY
  },
  "eventType": "EVENT_TYPE"
}
```

Response

201

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

ACKNOWLEDGEMENTS

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	11/28/2016	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of Draft Version of the Document.