TM Forum Specification

Service Test Management API Conformance Profile

TMF653B

Release 18.0.0

June 2018

Latest Update: TM Forum Release 18.0.0	Member Evaluation
Version 2.0.1	IPR Mode: RAND

NOTICE

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 - API DESCRIPTION....4

RESOURCE MODEL CONFORMANCE....5

- Service Test Management API MANDATORY AND OPTIONAL RESOURCES.... 5
- Service Test Specification Resource MANDATORY AND OPTIONAL ATTRIBUTES.... 5
- Service Test Resource MANDATORY AND OPTIONAL ATTRIBUTES.... 7

NOTIFICATION MODEL CONFORMANCE....11

- ServiceTestSpecification API MANDATORY AND OPTIONAL NOTIFICATIONS.... 11
- ServiceTest API MANDATORY AND OPTIONAL NOTIFICATIONS.... 11

API OPERATIONS CONFORMANCE....12

- API MANDATORY AND OPTIONAL OPERATIONS.... 12

API GET OPERATION CONFORMANCE....13

- GET /serviceTest/{id}?fields=...&{filtering}.... 13
- GET /serviceTestSpecification?fields=...&{filtering}.... 13
- GET /serviceTest/{id}?fields=...&{filtering}.... 13
- GET /serviceTestSpecification?fields=...&{filtering}.... 13

API POST OPERATION CONFORMANCE....15

- POST /serviceTestSpecification.... 15
- POST /serviceTest.... 16

API PATCH OPERATION CONFORMANCE....18

- PATCH /serviceTestSpecification/{id}.... 18
- PATCH /serviceTest/{id}.... 19

API DELETE OPERATION CONFORMANCE....21

- DELETE /serviceTestSpecification/{id}.... 21
- DELETE /serviceTest/{id}.... 21

API CONFORMANCE TEST SCENARIOS....22

- ServiceTest Resource TEST CASES.... 22
- ServiceTestSpecification Resource TEST CASES.... 28

ACKNOWLEDGEMENTS....35

- Version HISTORY.... 35
- Release History.... 35

INTRODUCTION - API DESCRIPTION

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.

This document identifies the parameters that must be included in a request related to the operations above as well as the parameters expected in the response.

The test scenarios in this document are intended to create a set of resources (Service Test and Service Test Specification) and then retrieve the information stored in the server to confirm the resources created are stored with the values originally set. Additionally, some test scenarios are included to verify that the server replies with the corresponding error response in situations where a mandatory attribute is not included in the request.

RESOURCE MODEL CONFORMANCE

Service Test Management API MANDATORY AND OPTIONAL RESOURCES

For the Resources defined by the API, here the following table indicates which are mandatory and which ones are optional.

Resource Name	Mandatory / Optional	Comments
SERVICE TEST	M	
SERVICE TEST SPECIFICATION	M	

Service Test Specification Resource MANDATORY AND OPTIONAL ATTRIBUTES

Parameter	**Mandatory/Optional**	**Comments**
id	M (in response messages) O (otherwise)	Generated by the server and provided in the response upon resource creation. Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
href	M (in response messages) O (otherwise)	Value in response must be the same as the one set in Location header provided upon entity creation
@type	O	
@baseType	O	
@schemaLocation	O	
description	O	
name	M	
validFor	O	

testMeasureDefinition			***O***	A list of test measure definition. (TestMeasureDefinition[*])
	captureFrequency		O	
	captureMethod		O	
	capturePeriod		O	
	metricDescription		O	
	metricHref		M	
	metricName		M	
	name		M	
	unitOfMeasure		O	
	valueType		O	
	validFor		O	
	@type		O	
	@baseType		O	
	@schemaLocation		O	
	thresholdRule		O	A list of threshold rule . (MetricDefMeasureThresholdRule[*])
		conformanceTargetLower	M	
		conformanceTargetUpper	M	
		conformanceComparatorLower	M	
		conformanceComparatorUpper	M	
		description	O	
		name	M	
		numberOfAllowedCrossing	O	

		thresholdRuleSeverity		O	
		tolerancePeriod		O	
		@type		O	
		@baseType		O	
		@schemaLocation		O	
		consequence		O	A list of measure consequence. (MetricDefMeasureConsequence[*])
			description	O	
			name	O	
			prescribeAction	O	
			repeatAction	O	
			validFor	O	
			@type	O	
			@baseType	O	
			@schemaLocation	O	
RelatedServiceSpecification				M	
	id			M	
	href			M	
	@referredType			O	

Service Test Resource MANDATORY AND OPTIONAL ATTRIBUTES

For the resources defined by the API the following table indicates which ones are mandatory and which ones are optional.

Parameter	Mandatory/Optional	Comments
@type	O	
@baseType	O	
@schemaLocation	O	
id	M (in response messages) O (otherwise)	Generated by the server and provided in the response upon resource creation. Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
href	M (in response messages) O (otherwise)	Value in response must be the same as the one set in Location header provided upon entity creation
description	O	
endDateTime	O	
name	M	
mode	O	
startDateTime	O	
state	O	
testMeasure	***O***	A list of test measure.(TestMeasure[*])
@type	O	
@baseType	O	
@schemaLocation	O	
accuracy	O	

	captureDateTime				
	captureMethod			O	
	metricDescription			O	
	metricHref			M	
	metricName			M	
	unitOfMeasure			O	
	value				
	valueType			O	
	ruleViolation			O	A list of rule violation. (MeasureThresholdRuleViolation[*])
		@type		O	
		@baseType		O	
		@schemaLocation		O	
		conformanceTargetLower		M	
		conformanceTargetUpper		M	
		conformanComparatorLower		M	
		conformanComparatorUpper		M	
		description		O	
		name		M	
		numberOfAllowedCrossing		O	
		thresholdRuleSeverity		O	
		tolerancePeriod		O	
		appliedConsequence		O	A list of applied consequence.(AppliedConsequence[*])
			@type	O	

			@baseType	O	
			@schemaLocation	O	
			description	O	
			name	O	
			repeatAction	O	
			appliedAction	O	
characteristic				O	A list of service test characteristic. (ServiceTestCharacteristic)
	name			M	
	value			M	
relatedService				M	ServiceRef
	@referredType			O	
	href			M	
	id			M	
	name				
testSpecification				M	ServiceSpecificationRef
	@referredType			O	
	href			M	
	id			M	

NOTIFICATION MODEL CONFORMANCE

The Pub/Sub models are common and described in the TMF REST Design Guidelines. Use the following templates to describe the Hub Mandatory and Optional attributes and filtering support.

ServiceTestSpecification API MANDATORY AND OPTIONAL NOTIFICATIONS

For the Notifications defined by the API, it is filled in the following table to indicate which ones are mandatory and which ones are optional.

All notifications are optional.

Resource Name	Mandatory or Optional	Comments
ServiceTestSpecificationCreationNotification	O	
ServiceTestSpecificationRemoveNotification	O	
ServiceTestSpecificationAttributeChangeNotification	O	

ServiceTest API MANDATORY AND OPTIONAL NOTIFICATIONS

Resource Name	Mandatory or Optional	Comments
ServiceTestCreationNotification	O	
AttributeValueChangeNotificationNotification	O	
ServiceTestRemoveNotification	O	

API OPERATIONS CONFORMANCE

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

API MANDATORY AND OPTIONAL OPERATIONS

The following table indicates which ones are mandatory and which ones are optional for each one of the resources in the API (default is for all resources).

Uniform API Operation	Mandatory/Optional	Comments
GET	M for all resources	GET must be used to retrieve a representation of a resource
POST	M for all resources	POST must be used to create a new resource
PUT	O for all resources	PUT must be used to completely update a resource identified by its resource URI
PATCH	O for all resources	PATCH must be used to partially update a resource
DELETE	O for all resources	DELETE must be used to remove a resource

API GET OPERATION CONFORMANCE

For every single resource use the following template to specify the mandatory and optional features supported by the GET operation.

GET	M	
Response Status Code 200	M	
Other Status Codes	NA	

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

Filtered Data (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 will provide in the response only the values assigned to attributes category and channel. Attribute selection capabilities are the same for collections retrieval and individual resource queries

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

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

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

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

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

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

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

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

API POST OPERATION CONFORMANCE

For every single resource, use the following template to specify the mandatory and optional features supported by the POST operation.

This Uniform Contract operation is used to create an entity.

POST	M	
Status Code 201	M	
Other Status Codes	NA	

POST /serviceTestSpecification

Description

This operation is used to create a managed entity for serviceTestSpecification.

Mandatory and Non Mandatory Attributes

The following tables specifies the attributes required when an entity is created (and their default values if not) as well as attributes with special considerations:

Attribute name	Mandatory/Option	Rule
id	O	Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
@type	O	
@baseType	O	
@schemaLocation	O	
description	O	
name	M	

Attribute name	Mandatory/Option	Rule
validFor	O	
testMeasureDefinition	O	
relatedServiceSpecification	M	

Specify the attributes that must be included in the BODY of the response, even if they are not included in the request because they are mandatory in the definition of the resource to be created as per the resource model defined

- href
- id

Additional Rules

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

Object	Mandatory Sub-Attributes
ServiceSpecificationRef	href, id
TestMeasureDefinition	metricName, metricHref, name
MetricDefMeasureThresholdRule	conformanceTargetLower, conformanceTargetUpper, conformanceComparatorLower, conformanceComparatorUpper, name

POST /serviceTest

Description

This operation creates service level specification entity.

The following tables specifies the attributes required when an entity is created (and their default values if not) as well as attributes with special considerations:

Attribute name	Mandatory/Option	Rule
id	O	Accepted in entity-creation requests if the server supports the incoming identifier as the reference to create new resources
@type	O	

Attribute name	Mandatory/Option	Rule
@baseType	O	
@schemaLocation	O	
description	O	
endDateTime	O	
mode	O	
name	M	
startDateTime	O	
state	O	
relatedSpecification	M	
characteristic	O	
relatedService	M	
testMeasure	O	

Specify the attributes that must be included in the BODY of the response, even if they are not included in the request because they are mandatory in the definition of the resource to be created as per the resource model defined

- href
- id

Additional Rules

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

Object	Mandatory Sub-Attributes
ServiceRef	href, id
ServiceTestSpecificationRef	href, id
ServiceTestCharacteristic	name, value
MeasureThresholdRuleViolation	name, conformanceTargetLower, conformanceTargetUpper, conformanceComparatorLower, conformanceComparatorUpper
TestMeasure	metricHref, metricName

API PATCH OPERATION CONFORMANCE

This Uniform Contract operation is used to partially update the representation of a Resource.

PATCH	O	
Status Code 200	M	
Other Status Codes	NA	

PATCH /serviceTestSpecification/{id}

Description

This operation allows partial updates of a service level objective 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, 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.

Attribute Name	Patchable ? (Y/N)	Rule
id	N	
href	N	
@type	N	
@baseType	N	
@schemaLocation	N	
description	Y	

Attribute Name	Patchable ? (Y/N)	Rule
name	Y	
validFor	Y	
testMeasureDefinition	Y	
relatedServiceSpecification	Y	

PATCH /serviceTest/{id}

Description

This operation allows partial updates of a service level 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.

Attribute Name	Patchable ? (Y/N)	Rule
id	N	
href	N	
@type	N	
@baseType	N	
@schemaLocation	N	
description	Y	
endDateTime	Y	

Attribute Name	Patchable ? (Y/N)	Rule
mode	Y	
name	Y	
startDateTime	Y	
state	Y	
relatedSpecification	Y	
characteristic	Y	
relatedService	Y	
testMeasure	Y	

API DELETE OPERATION CONFORMANCE

This Uniform Contract operation is used to delete a resource.

DELETE	O	THIS PATCH OPERATION
Status Code 204	M	
Other Status Codes	NA	

DELETE /serviceTestSpecification/{id}

Description

This operation deletes a serviceTestSpecification entity.

For this operation, only the “id” field is mandatory for the client of the API. All other fields are not required.

DELETE /serviceTest/{id}

Description

This operation deletes a party serviceTest entity.

For this operation, only the “id” field is mandatory for the client of the API. All other fields are not required.

API CONFORMANCE TEST SCENARIOS

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

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

Requests must be addressed to the endpoint provided for certification, specifically they must be addressed to the URI defined by the concatenation of the {apiRoot} and the specific resource, where the {apiRoot} is defined as **{serverRoot}/serviceTestManagement/v2**, where {serverRoot} defines the certification endpoint

ServiceTest Resource TEST CASES

Nominal Scenarios

TC_ServiceTest_N1 – Create new Service Test with minimum required information

- Send a POST message to {apiRoot}/serviceTest/ with the following contents in the BODY

```
{

  "name": "test1",
  "relatedService":{
   "id":"service1",
   "href":"/service1"
  },
  "testSpecification":{
   "id":"spec1",
    "href":"/spec1"
  }
}
```

- Wait for a response from the server with the following characteristics

  - Response Code 201-Created

  - Include a location header in the body set to /{apiRoot}/serviceTest/{ID1} where {ID1} indicates the identifier assigned by the server to the new Service Test resource

  - The response message includes all mandatory parameters

  - The body of the response matches the values set in the original request

- Send a GET message to /{apiRoot}/serviceTest/
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource with ID set to {ID1}, the same identifier as assigned by the server to the new resource created
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID1} matches the values set in the original request
- Send a GET message to /{apiRoot}/serviceTest/{ID1}
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The response message includes all mandatory parameters
  - The body of the response includes a Service Test resource structure that matches the values in the original request

TC_ServiceTest_N2 – Create new Service Test with minimum set of parameters supported by server

- Send a POST message to {apiRoot}/serviceTest/ with the following contents in the BODY

```
{

  "id": "test2",
  "name": "test2",
  "relatedService":{
    "id":"service2",
    "href":"/service2"
  },
  "testSpecification":{
    "id":"spec2",
      "href":"/spec2"
  }
}
```

- Wait for a response from the server with the following characteristics
  - Response Code 201-Created
  - Include a location header in the body set to /{apiRoot}/serviceTest/{ID2} where {ID2} indicates the identifier assigned by the server to the new Service Test resource
  - The response message includes all mandatory parameters
  - The body of the response matches the values set in the original request
- Send a GET message to /{apiRoot}/serviceTest/
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource with ID set to {ID2}, the same identifier as assigned by the server to the new resource created
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID2} matches the values set in the original request
- Send a GET message to /{apiRoot}/serviceTest/{ID2}
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The response message includes all mandatory parameters
  - The body of the response includes a Service Test resource structure that matches the values in the original request

TC_ServiceTest_N3 – Search for ServiceTests with specific characteristics

- Send a GET message to /{apiRoot}/serviceTest
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes at least two Service Test resources referring to {ID1} and {ID2}
  - 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}/serviceTest?name=test1
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource referring to {ID1} and there is no reference to Service Test resource {ID2}
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID1} matches the values in the original request
- Send a GET message to /{apiRoot}/serviceTest? relatedService.id=service1
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource referring to {ID1} and there is no reference to Service Test resource {ID2}
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID1} matches the values in the original request

TC_ServiceTest_N4 – Filtered retrieval of ServiceTests

- Send a GET message to /{apiRoot}/serviceTest/{ID1}?fields= name
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource referring to {ID1} and including only attributes name, matching the values in the original request
- Send a GET message to /{apiRoot}/serviceTest/{ID2}?fields= relatedService, testSpecification
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource referring to {ID2} and including only attributes name and relatedService, matching the values in the original request

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

TC_ServiceTest_N5 – Filtered Search and Filtered data response

- Send a GET message to /{apiRoot}/serviceTest? name=test1& relatedService.id="service1"
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one Service Test resource referring to {ID1} and there is no reference to Service Test resource {ID2}
  - The body of the response for the resource with each identifier includes only attribute description, matching the values in the corresponding original request

Notice that this test case is using the parameter “name” and “relatedService.id” to filter the data included in the response but any other parameter could be used

Error Scenarios

TC_ServiceTest_E1 – Unknown Service Test identifier

- Send a GET message to /{apiRoot}/serviceTest/{ID3}, where {ID3} does not match any of the identifiers previously creted in the server

- Wait for a response from the server with the following characteristics

    - Response Code 404-Not Found

TC_ServiceTest_E2 – Invalid Request – Missing mandatory parameter

- Send a POST message to {apiRoot}/serviceTest with the following contents in the BODY.

```
{

  "name": "test",
  "relatedService":{
    "id":"spec",
    "href":"/spec"
  }
}
```

  Notice that this request is missing mandatory parameter “testSpecification” , but any other mandatory parameter could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

TC_ServiceTest_E3 – Invalid Request – Missing parameter mandatory in specParameter

- Send a POST message to {apiRoot}/serviceTest/ with the following contents in the BODY.

```
{

  "name": "test",
  "relatedService":{
```

```
  "href":"/spec"
},
"testSpecification":{
  "id":"spec",
    "href":"/spec"
}
```

} Notice that this request is missing mandatory parameters “id” when information element “relatedService” is included in the request, but any other parameter that becomes mandatory based on the context could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

Note: Not all the test cases is included in this document. There are more test cases in CTK file which is coming soon.

ServiceTestSpecification Resource TEST CASES

Nominal Scenarios

TC_ServiceTestSpecification_N1 – Create new service Test Specification with minimum required information

- Send a POST message to {apiRoot}/serviceTestSpecification/ with the following contents in the BODY

```
{
  "name": "tspec1",
  "relatedServiceSpecification":{
    "id":"rs1",
    "href":"/rs1"
  }
}
```

- Wait for a response from the server with the following characteristics

  - Response Code 201-Created

  - Include a location header in the body set to /{apiRoot}/ serviceTestSpecification/{ID1} where {ID1} indicates the identifier assigned by the server to the new service Level Specification resource

  - The response message includes all mandatory parameters

  - The body of the response matches the values set in the original request

- Send a GET message to /{apiRoot}/serviceTestSpecification/

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one service Test Specification resource with ID set to {ID1}, the same identifier as assigned by the server to the new resource created

  - The response message includes all mandatory parameters

  - The body of the response for the resource with identifier {ID1} matches the values set in the original request

- Send a GET message to /{apiRoot}/serviceTestSpecification/{ID1}

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The response message includes all mandatory parameters

  - The body of the response includes a service Test Specification resource structure that matches the values in the original request

TC_ServiceTestSpecification_N2 – Create new service Level Specification with minimum set of parameters supported by server

- Send a POST message to {apiRoot}/serviceTestSpecification/ with the following contents in the BODY

```
{
   "id":" tspec2",
   "name": "tspec2",
   "relatedServiceSpecification":{
      "id":"rs2",
        "href":"/rs2"
      }
    }
```

- Wait for a response from the server with the following characteristics
  - Response Code 201-Created
  - Include a location header in the body set to /{apiRoot}/serviceTestSpecification/{ID2} where {ID2} indicates the identifier assigned by the server to the new service Test Specification resource
  - The response message includes all mandatory parameters
  - The body of the response matches the values set in the original request
- Send a GET message to /{apiRoot}/serviceTestSpecification /
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one service Test Specification resource with ID set to {ID2}, the same identifier as assigned by the server to the new resource created
  - The response message includes all mandatory parameters
  - The body of the response for the resource with identifier {ID2} matches the values set in the original request
- Send a GET message to /{apiRoot}/serviceTestSpecification/{ID2}
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The response message includes all mandatory parameters

  - The body of the response includes a service Test Specification resource structure that matches the values in the original request

TC_ServiceTestSpecification_N3 – Search for ServiceTestSpecifications with specific characteristics

- Send a GET message to /{apiRoot}/serviceTestSpecification

- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

  - The body of the response includes at least two service Level Specification resources referring to {ID1} and {ID2}

  - 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}/serviceTestSpecification?name=tspec2

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one service Test Specification resource referring to {ID1} and there is no reference to service Level Specification resource {ID2}

  - The response message includes all mandatory parameters

  - The body of the response for the resource with identifier {ID1} matches the values in the original request

- Send a GET message to /{apiRoot}/serviceTestSpecification?relatedServiceSpecification.id=rs2

- Wait for a response from the server with the following characteristics

  - Response Code 200-OK

  - The body of the response includes one service Test Specification resource referring to {ID2} and there is no reference to service Test Specification resource {ID1}

- The response message includes all mandatory parameters
- The body of the response for the resource with identifier {ID2} matches the values in the original request

TC_ServiceTestSpecification_N4 – Filtered retrieval of a service Test Specification

- Send a GET message to /{apiRoot}/serviceTestSpecification/{ID1}?fields=name
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one service Test Specification resource referring to {ID1} and including only attributes name , matching the values in the original request
- Send a GET message to /{apiRoot}/serviceTestSpecification/{ID2}?fields=relatedServiceSpecification,name
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK
  - The body of the response includes one service Test Specification resource referring to {ID2} and including only attributes objective and name, matching the values in the original request

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

TC_ServiceTestSpecification_N5 – Filtered Search and Filtered data response

- Send a GET message to /{apiRoot}/serviceTestSpecification?name=tspec2&relatedServiceSpecification.id=rs2
- Wait for a response from the server with the following characteristics
  - Response Code 200-OK

- The body of the response includes one ServiceTestSpecification resource referring to {ID2} and there is no reference to service Level Specification resource {ID1}

- The body of the response for the resource with each identifier includes only attribute description, matching the values in the corresponding original request

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

Error Scenarios

TC_ServiceTestSpecification_E1 – Unknown service Test Specification identifier

- Send a GET message to /{apiRoot}/serviceTestSpecification/{ID3}, where {ID3} does not match any of the identifiers previously creted in the server

- Wait for a response from the server with the following characteristics

  - Response Code 404-Not Found

TC_ServiceTestSpecification_E2 – Invalid Request – Missing mandatory parameter

- Send a POST message to {apiRoot}/serviceTestSpecification with the following contents in the BODY.

```
{

  "relatedServiceSpecification":{
    "id":"",
    "href":""
  }

}
```

  Notice that this request is missing mandatory parameter "name", but any other mandatory parameter could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

TC_ServiceTestSpecification_E3 – Invalid Request – Missing parameter mandatory in objective

- Send a POST message to {apiRoot}/serviceTestSpecification/ with the following contents in the BODY.

  ```
  {
     "name": "tspec1",
     "relatedServiceSpecification":{

       "href":""
     }

  }
  ```

  Notice that this request is missing mandatory parameters “id” when information element “relatedServiceSpecification” is included in the request, but any other parameter that becomes mandatory based on the context could be used

- Wait for an error response from the server indicating the mandatory parameter is missing in the request

Note: Not all the test cases are included in this document. There are more test cases in CTK file which is coming soon.

ACKNOWLEDGEMENTS

VERSION HISTORY

Version Number	Date	Modified by	Description
Version 1.0.0	8/05/2017	Ma Xu Maxu@huawei.com Hongxia Hao haohongxia@huawei.com	initial version
Version 1.01	14-Jun-2017		Updated version
Version 1.02	9-Sep-2017	Hongxia Hao haohongxia@huawei.com Ma Xu Maxu@huawei.com	Align with the newest specification document
Version 1.0.3	23-Jan-2018	Adrienne Walcott	Formatting/style edits prior to publishing
Version 2.0.0	12-Jun-2018	Hongxia Hao haohongxia@huawei.com	Modify some typos. Change the descriptions of @attributes related with DG3.0 Add test cases
Version 2.0.1	27-Jun-2018	Adrienne Walcott	Formatting/style edits prior to R18 publishing.

RELEASE HISTORY

Release Number	Date	Release led by:	Description
17.5.0	January 2018	Hongxia Hao, Ma Xu	Initial Release
18.0.0	June 2018	Hongxia Hao,	Second Release