



Network Function Virtualisation (NFV) Release 2; Testing; API Conformance Testing Specification

ITeH STANBARD PREVIEW
(standards.iteh.ai)
Full standard(s) list/sist/c87/56-d2c1-
https://standards.iteh.ai/catalog/standards/sist/c87-56-d2c1-
48c8-baba-48163ea12999/etsi-gs-nfv-010-v2.4.1-2020-03

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

ReferenceDGS/NFV-TST010

KeywordsAPI, conformance, NFV, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights	25
Foreword.....	25
Modal verbs terminology.....	25
Introduction	25
1 Scope	26
2 References	26
2.1 Normative references	26
2.2 Informative references.....	27
3 Definition of terms, symbols and abbreviations.....	27
3.1 Terms.....	27
3.2 Symbols.....	27
3.3 Abbreviations	27
4 Methodology	28
4.1 General	28
4.2 System under test	28
4.3 Test configurations	28
4.3.1 General.....	28
4.3.2 Config_prod_VE.....	29
4.3.3 Config_prod_VNFM	29
4.3.4 Config_prod_NFVO	30
4.3.5 Config_prod_VNFM_GRANT.....	30
4.4 Void.....	31
4.5 Generic Test Description.....	31
4.5.1 General.....	31
4.5.2 Test Description format	31
4.5.3 Scope of the tests	33
4.5.3.1 General	33
4.5.3.2 General characteristics of the reference points.....	33
4.5.3.3 Basic behaviours of the API producer/consumer and verification steps	34
4.5.3.3.0 Introduction	34
4.5.3.3.1 Producer sends event triggered notifications based on consumer subscriptions.....	34
4.5.3.3.2 Producer sends periodic notifications based on the consumer subscriptions.....	36
4.5.3.3.3 Producer executes the requested API operation	37
4.5.3.3.4 Consumer fetches the files/package info	38
4.5.3.4 Workflow test considerations.....	40
4.5.4 Verification	40
4.5.4.0 Introduction.....	40
4.5.4.1 Common verification aspects	40
4.5.4.2 Verification aspects for individual API.....	41
4.5.4.3 Verification aspects not considered.....	42
5 Os-Ma-Nfvo Reference Point.....	42
5.1 General	42
5.2 Test configuration.....	42
5.3 Test Descriptions.....	42
5.3.1 NSD Management interface.....	42
5.3.1.1 Network Service Descriptors Resource Endpoint	42
5.3.1.1.1 GET all Network Service Descriptors Information	42
5.3.1.1.2 GET Network Service Descriptors Information with attribute-based filter.....	43
5.3.1.1.3 GET Network Service Descriptors Information with invalid attribute-based filter.....	43
5.3.1.1.4 Get all Network Service Descriptors Information with malformed authorization token.....	43
5.3.1.1.5 Get all Network Service Descriptors Information without authorization token	44
5.3.1.1.6 GET all Network Service Descriptors Information with expired or revoked authorization token	44

5.3.1.1.7	GET all Network Service Descriptors Information with "all_fields" attribute selector.....	44
5.3.1.1.8	GET all Network Service Descriptors Information with "exclude_default" attribute selector	45
5.3.1.1.9	GET all Network Service Descriptors Information with "fields" attribute selector.....	45
5.3.1.1.10	GET all Network Service Descriptors Information with "exclude_fields" attribute selector	45
5.3.1.1.11	Create new Network Service Descriptor Resource.....	46
5.3.1.1.12	PUT all Network Service Descriptors - Method not implemented.....	46
5.3.1.1.13	PATCH all Network Service Descriptors - Method not implemented	47
5.3.1.1.14	DELETE all Network Service Descriptors - Method not implemented	47
5.3.1.2	Individual NS Descriptor Resource Endpoint	47
5.3.1.2.1	GET Individual Network Service Descriptor Information	47
5.3.1.2.2	GET Individual Network Service Descriptor Information with invalid resource identifier	48
5.3.1.2.3	Disable Individual Network Service Descriptor	48
5.3.1.2.4	Enable Individual Network Service Descriptor	48
5.3.1.2.5	Enable Individual Network Service Descriptor with conflict due to operational state ENABLED	49
5.3.1.2.6	Enable Individual Network Service Descriptor with conflict due to onboarding state.....	49
5.3.1.2.7	Enable Individual Network Service Descriptor with HTTP Etag precondition failure	50
5.3.1.2.8	DELETE Individual Network Service Descriptor	50
5.3.1.2.9	DELETE Individual Network Service Descriptor in operational state ENABLED	50
5.3.1.2.10	POST Individual Network Service Descriptor - Method not implemented.....	51
5.3.1.2.11	PUT Individual Network Service Descriptor - Method not implemented	51
5.3.1.3	NSD Content Resource Endpoint.....	51
5.3.1.3.1	Get single file NSD Content in Plain Format	51
5.3.1.3.2	Get NSD Content in Zip Format.....	52
5.3.1.3.3	Get single file NSD Content in Plain or Zip Format	52
5.3.1.3.4	Get multi file NSD Content in Plain or Zip Format	52
5.3.1.3.5	Get multi file NSD Content in Plain Format	53
5.3.1.3.6	Get NSD Content with invalid resource identifier.....	53
5.3.1.3.7	Get NSD Content with conflict due to onboarding state	53
5.3.1.3.8	GET NSD Content with Range Request and NFVO supporting Range Requests.....	54
5.3.1.3.9	GET NSD Content with Range Request and NFVO not supporting Range Requests.....	54
5.3.1.3.10	GET NSD Content with invalid Range Request.....	54
5.3.1.3.11	Upload NSD Content as Zip file in asynchronous mode	55
5.3.1.3.12	Upload NSD Content as plain text file in asynchronous mode.....	55
5.3.1.3.13	Upload NSD Content as Zip file in synchronous mode.....	55
5.3.1.3.14	Upload NSD Content as plain text file in synchronous mode	56
5.3.1.3.15	Upload NSD Content with conflict due to onboarding state	56
5.3.1.3.16	POST NSD Content - Method not implemented	56
5.3.1.3.17	PATCH NSD Content - Method not implemented	57
5.3.1.3.18	DELETE NSD Content - Method not implemented.....	57
5.3.1.4	PNF Descriptors Resource Endpoint.....	57
5.3.1.4.1	GET all PNF Descriptors Information.....	57
5.3.1.4.2	GET PNF Descriptors Information with attribute-based filter	58
5.3.1.4.3	GET PNF Descriptors Information with invalid attribute-based filter	58
5.3.1.4.4	GET all PNF Descriptors Information with "all_fields" attribute selector	58
5.3.1.4.5	GET all PNF Descriptors Information with "exclude_default" attribute selector	59
5.3.1.4.6	GET all PNF Descriptors Information with "fields" attribute selector.....	59
5.3.1.4.7	GET all PNF Descriptors Information with "exclude_fields" attribute selector.....	60
5.3.1.4.8	Create new PNF Descriptor Resource	60
5.3.1.4.9	PUT all PNF Descriptors - Method not implemented	60
5.3.1.4.10	PATCH all PNF Descriptors - Method not implemented.....	61
5.3.1.4.11	DELETE all PNF Descriptors - Method not implemented.....	61
5.3.1.5	Individual PNF Descriptor Resource Endpoint.....	61
5.3.1.5.1	GET Individual PNF Descriptor Information.....	61
5.3.1.5.2	GET Individual PNF Descriptor Information with invalid resource identifier	62
5.3.1.5.3	Update Individual PNF Descriptor	62
5.3.1.5.4	Update Individual PNF Descriptor with HTTP Etag precondition failure	62
5.3.1.5.5	POST Individual PNF Descriptor - Method not implemented	63
5.3.1.5.6	PUT Individual PNF Descriptor - Method not implemented.....	63
5.3.1.5.7	DELETE Individual PNF Descriptor.....	63
5.3.1.6	PNFD Content Resource Endpoint	64
5.3.1.6.1	Get PNFD Content	64

5.3.1.6.2	Get PNFD Content with invalid resource identifier	64
5.3.1.6.3	Get PNFD Content with conflict due to onboarding state	64
5.3.1.6.4	Upload PNFD Content as plain text file	65
5.3.1.6.5	Upload PNFD Content with conflict due to onboarding state	65
5.3.1.6.6	POST PNFD Content - Method not implemented	65
5.3.1.6.7	PATCH PNFD Content - Method not implemented.....	66
5.3.1.6.8	DELETE PNFD Content - Method not implemented.....	66
5.3.1.7	Subscriptions Resource Endpoint.....	66
5.3.1.7.1	Get All NSD Management Subscriptions.....	66
5.3.1.7.2	Get NSD Management Subscriptions with attribute-based filter.....	67
5.3.1.7.3	Get NSD Management Subscriptions with invalid attribute-based filter.....	67
5.3.1.7.4	GET NSD Management Subscription with invalid resource endpoint.....	67
5.3.1.7.5	Create new NSD Management subscription.....	68
5.3.1.7.6	Create duplicated NSD Management subscription with NFVO not creating duplicated subscriptions	68
5.3.1.7.7	Create duplicated NSD Management subscription with NFVO creating duplicated subscriptions	69
5.3.1.7.8	PUT NSD Management Subscriptions - Method not implemented.....	69
5.3.1.7.9	PATCH NSD Management Subscriptions - Method not implemented	69
5.3.1.7.10	DELETE NSD Management Subscriptions - Method not implemented	70
5.3.1.8	Individual Subscription Resource Endpoint.....	70
5.3.1.8.1	GET Individual NSD Management Subscription	70
5.3.1.8.2	GET Individual NSD Management Subscription with invalid resource identifier	70
5.3.1.8.3	DELETE Individual NSD Management Subscription with invalid resource identifier.....	71
5.3.1.8.4	POST Individual NSD Management Subscription - Method not implemented.....	71
5.3.1.8.5	PUT Individual NSD Management Subscription - Method not implemented	71
5.3.1.8.6	PATCH Individual NSD Management Subscription - Method not implemented	72
5.3.1.8.7	DELETE Individual NSD Management Subscription.....	72
5.3.1.9	Notifications Endpoint	72
5.3.1.9.1	NSD Onboarding Notification.....	72
5.3.1.9.2	NSD Onboarding Failure Notification.....	73
5.3.1.9.3	NSD Operational State Change Notification.....	73
5.3.1.9.4	NSD Deletion Notification.....	74
5.3.1.9.5	PNFD Onboarding Notification.....	74
5.3.1.9.6	PNFD Onboarding Failure Notification.....	75
5.3.1.9.7	PNFD Deletion Notification.....	75
5.3.2	NS Lifecycle Management interface.....	76
5.3.2.1	NsInstances	76
5.3.2.1.1	POST Create a new NsInstance.....	76
5.3.2.1.2	GET information about multiple NS instances.....	76
5.3.2.1.3	GET information about multiple NS instances Bad Request Invalid attribute-based filtering parameters	77
5.3.2.1.4	GET information about multiple NS instances Bad Request Invalid attribute selector.....	77
5.3.2.1.5	GET NSInstances with "all_fields" attribute selector.....	77
5.3.2.1.6	GET NSInstances with "exclude_default" attribute selector	78
5.3.2.1.7	GET NSInstances with "fields" attribute selector.....	78
5.3.2.1.8	GET NSInstances with "exclude_fields" attribute selector	78
5.3.2.1.9	PUT NSInstances - Method not implemented	79
5.3.2.1.10	PATCH NSInstances - Method not implemented.....	79
5.3.2.1.11	DELETE NSInstances - Method not implemented.....	79
5.3.2.2	Individual NSInstance.....	80
5.3.2.2.1	POST Individual NSInstance - Method not implemented	80
5.3.2.2.2	GET Information about an individual NS Instance	80
5.3.2.2.3	PUT Individual NSInstance - Method not implemented	80
5.3.2.2.4	PATCH Individual NSInstance - Method not implemented.....	81
5.3.2.2.5	DELETE Individual NSInstance	81
5.3.2.2.6	DELETE Individual NSInstance Conflict	81
5.3.2.3	Instantiate NS Task	82
5.3.2.3.1	POST Instantiate a nsInstance	82
5.3.2.3.2	POST Instantiate a nsInstance Conflict	82
5.3.2.3.3	GET Instantiate NSInstance - Method not implemented	82
5.3.2.3.4	PUT Instantiate NSInstance - Method not implemented	83
5.3.2.3.5	PATCH Instantiate NSInstance - Method not implemented.....	83

5.3.2.3.6	DELETE Instantiate NSInstance - Method not implemented.....	83
5.3.2.4	Scale a nsInstance	84
5.3.2.4.1	POST Scale a nsInstance	84
5.3.2.4.2	POST Instantiate a nsInstance Conflict	84
5.3.2.4.3	GET Scale NSInstance- Method not implemented.....	84
5.3.2.4.4	PUT Scale NSInstance - Method not implemented	85
5.3.2.4.5	PATCH Scale NSInstance - Method not implemented.....	85
5.3.2.4.6	DELETE Scale NSInstance - Method not implemented.....	85
5.3.2.5	Update a NSInstance	86
5.3.2.5.1	POST Update a NSInstance.....	86
5.3.2.5.2	POST Update a NSInstance Conflict.....	86
5.3.2.5.3	GET Update NSInstance- Method not implemented	86
5.3.2.5.4	PUT Update NSInstance - Method not implemented	87
5.3.2.5.5	PATCH Update NSInstance - Method not implemented.....	87
5.3.2.5.6	DELETE Update NSInstance - Method not implemented.....	87
5.3.2.6	Heal a NSInstance	88
5.3.2.6.1	POST Heal a NSInstance.....	88
5.3.2.6.2	POST Heal a NSInstance Conflict.....	88
5.3.2.6.3	GET Heal NSInstance- Method not implemented	88
5.3.2.6.4	PUT Heal NSInstance - Method not implemented	89
5.3.2.6.5	PATCH Heal NSInstance - Method not implemented.....	89
5.3.2.6.6	DELETE Heal NSInstance - Method not implemented.....	89
5.3.2.7	Terminate a NSInstance	90
5.3.2.7.1	POST Terminate a NSInstance.....	90
5.3.2.7.2	POST Terminate a NSInstance Conflict.....	90
5.3.2.7.3	GET Terminate NSInstance- Method not implemented	90
5.3.2.7.4	PUT Terminate NSInstance - Method not implemented	91
5.3.2.7.5	PATCH Terminate NSInstance - Method not implemented.....	91
5.3.2.7.6	DELETE Terminate NSInstance - Method not implemented.....	91
5.3.2.8	NS LCM occurrences	92
5.3.2.8.1	POST NS LCM occurrences - Method not implemented	92
5.3.2.8.2	GET status information about multiple NS LCM occurrences.....	92
5.3.2.8.3	GET status information about multiple NS LCM occurrences Bad Request Invalid attribute-based filtering parameters.....	92
5.3.2.8.4	GET status information about multiple NS LCM occurrences Bad Request Invalid attribute selector.....	93
5.3.2.8.5	PUT status information about multiple NS LCM occurrences - Method not implemented.....	93
5.3.2.8.6	PATCH status information about multiple NS LCM occurrences - Method not implemented.....	93
5.3.2.8.7	DELETE status information about multiple NS LCM occurrences - Method not implemented	94
5.3.2.9	Individual NS LCM Occurrences.....	94
5.3.2.9.1	Post Individual NS LCM Occurrences - Method not implemented.....	94
5.3.2.9.2	Get status information about Individual NS LCM occurrences.....	94
5.3.2.9.3	PUT status information about Individual NS LCM occurrences - Method not implemented	95
5.3.2.9.4	PATCH status information about Individual NS LCM occurrences - Method not implemented.....	95
5.3.2.9.5	DELETE status information about Individual NS LCM occurrences - Method not implemented	95
5.3.2.10	Retry operation task	96
5.3.2.10.1	Post Retry operation task.....	96
5.3.2.10.2	Post Retry operation task Not Found.....	96
5.3.2.10.3	Post Retry operation task Conflict.....	96
5.3.2.10.4	GET Retry operation task - Method not implemented.....	97
5.3.2.10.5	PUT Retry operation task - Method not implemented.....	97
5.3.2.10.6	PATCH Retry operation task - Method not implemented	97
5.3.2.10.7	DELETE Retry operation task - Method not implemented	98
5.3.2.11	Rollback operation task.....	98
5.3.2.11.1	POST Rollback operation task	98
5.3.2.11.2	POST Rollback operation task Not Found	98
5.3.2.11.3	POST Rollback operation task Conflict	99
5.3.2.11.4	GET Rollback operation task - Method not implemented	99
5.3.2.11.5	PUT Rollback operation task - Method not implemented	99
5.3.2.11.6	PATCH Rollback operation task - Method not implemented.....	100
5.3.2.11.7	DELETE Rollback operation task - Method not implemented.....	100

5.3.2.12	Continue operation task.....	100
5.3.2.12.1	POST Continue operation task	100
5.3.2.12.2	POST Continue operation task Not Found	101
5.3.2.12.3	POST Continue operation task Conflict	101
5.3.2.12.4	GET Continue operation task - Method not implemented.....	101
5.3.2.12.5	PUT Continue operation task - Method not implemented.....	102
5.3.2.12.6	PATCH Continue operation task - Method not implemented.....	102
5.3.2.12.7	DELETE Continue operation task - Method not implemented.....	102
5.3.2.13	Fail operation task.....	103
5.3.2.13.1	Post Fail operation task	103
5.3.2.13.2	Post Fail operation task Not Found	103
5.3.2.13.3	Post Fail operation task Conflict	103
5.3.2.13.4	GET Fail operation task - Method not implemented	104
5.3.2.13.5	PUT Fail operation task - Method not implemented	104
5.3.2.13.6	PATCH Fail operation task - Method not implemented.....	104
5.3.2.13.7	DELETE Fail operation task - Method not implemented.....	105
5.3.2.14	Cancel operation task	105
5.3.2.14.1	POST Cancel operation task.....	105
5.3.2.14.2	POST Cancel operation task Not Found.....	105
5.3.2.14.3	POST Cancel operation task Conflict.....	106
5.3.2.14.4	GET Cancel operation task - Method not implemented	106
5.3.2.14.5	PUT Cancel operation task - Method not implemented.....	106
5.3.2.14.6	PATCH Cancel operation task - Method not implemented	107
5.3.2.14.7	DELETE Cancel operation task - Method not implemented	107
5.3.2.15	Subscriptions.....	107
5.3.2.15.1	POST Create a new subscription	107
5.3.2.15.2	POST Create a new Subscription - DUPLICATION	108
5.3.2.15.3	POST Create a new Subscription - NO-DUPLICATION	108
5.3.2.15.4	GET Subscriptions.....	108
5.3.2.15.5	GET Subscription - Filter	109
5.3.2.15.6	GET subscriptions - Bad Request Invalid attribute-based filtering parameters.....	109
5.3.2.15.7	GET subscriptions with "all_fields" attribute selector.....	109
5.3.2.15.8	GET subscriptions with "exclude_default" attribute selector	110
5.3.2.15.9	GET subscriptions with "fields" attribute selector.....	110
5.3.2.15.10	GET subscriptions with "exclude_fields" attribute selector	110
5.3.2.15.11	PUT subscriptions - Method not implemented	111
5.3.2.15.12	PATCH subscriptions - Method not implemented	111
5.3.2.15.13	DELETE subscriptions - Method not implemented	111
5.3.2.16	Individual Subscription.....	112
5.3.2.16.1	POST Individual Subscription - Method not implemented	112
5.3.2.16.2	GET Information about an individual subscription	112
5.3.2.16.3	PUT an individual subscription - Method not implemented.....	112
5.3.2.16.4	PATCH an individual subscription - Method not implemented	113
5.3.2.16.5	DELETE an individual subscription.....	113
5.3.2.17	Notification Endpoint.....	113
5.3.2.17.1	NS LCM Operation Occurrence Start Notification	113
5.3.2.17.2	NS LCM Operation Occurrence Result Notification.....	114
5.3.2.17.3	NS Identifier Creation Notification	114
5.3.2.17.4	NS Identifier Deletion Notification	115
5.3.2.18	Other NSLCM operations	115
5.3.2.18.1	NS Instance Creation.....	115
5.3.2.18.2	NS Instance Deletion	116
5.3.2.18.3	Heal Flow of NS lifecycle management operations	116
5.3.2.18.4	Instantiate Flow of NS lifecycle management operations	116
5.3.2.18.5	Scale Flow of NS lifecycle management operations	117
5.3.2.18.6	Terminate Flow of NS lifecycle management operations.....	117
5.3.2.18.7	Update Flow of NS lifecycle management operations	118
5.3.3	NS Fault Management interface	118
5.3.3.1	Alarms.....	118
5.3.3.1.1	POST Alarms - Method not implemented	118
5.3.3.1.2	GET information about multiple alarms.....	119
5.3.3.1.3	GET information about multiple alarms with filters.....	119

5.3.3.1.4	GET information about multiple alarms Bad Request Invalid attribute-based filtering parameters	119
5.3.3.1.5	PUT Alarms - Method not implemented	120
5.3.3.1.6	PATCH Alarms - Method not implemented.....	120
5.3.3.1.7	DELETE Alarms - Method not implemented.....	120
5.3.3.2	Individual Alarm	121
5.3.3.2.1	POST Individual Alarm - Method not implemented	121
5.3.3.2.2	GET information about Individual Alarm	121
5.3.3.2.3	GET information about Invalid Individual Alarm	121
5.3.3.2.4	PUT Individual Alarm - Method not implemented	122
5.3.3.2.5	PATCH Alarm.....	122
5.3.3.2.6	PATCH Alarm - Conflict	122
5.3.3.2.7	PATCH Alarm - Precondition failed	123
5.3.3.2.8	DELETE Individual Alarm - Method not implemented	123
5.3.3.3	Subscriptions.....	123
5.3.3.3.1	Create a new alarm subscription.....	123
5.3.3.3.2	Create a new alarm subscription - DUPLICATION.....	124
5.3.3.3.3	Create a new alarm subscription - NO DUPLICATION	124
5.3.3.3.4	Retrieve a list of alarm subscriptions.....	124
5.3.3.3.5	Retrieve a list of alarm subscriptions - Filter.....	125
5.3.3.3.6	GET subscriptions - Bad Request Invalid attribute-based filtering parameters.....	125
5.3.3.3.7	PUT subscriptions - Method not implemented	125
5.3.3.3.8	PATCH subscriptions - Method not implemented	126
5.3.3.3.9	DELETE subscriptions - Method not implemented	126
5.3.3.4	Individual Subscription	126
5.3.3.4.1	Post Individual Subscription - Method not implemented	126
5.3.3.4.2	Get Information about an individual subscription	127
5.3.3.4.3	PUT an individual subscription - Method not implemented.....	127
5.3.3.4.4	PATCH an individual subscription - Method not implemented	127
5.3.3.4.5	DELETE an individual subscription.....	128
5.3.3.5	Notification Endpoint.....	128
5.3.3.5.1	NS Fault Alarm Notification	128
5.3.3.5.2	NS Fault Alarm Cleared Notification	128
5.3.3.5.3	NS Fault Alarm List Rebuilt Notification	129
5.3.4	NS Performance Management interface	129
5.3.4.1	PM Jobs Resource Endpoint	129
5.3.4.1.1	GET all NS Performance Monitoring Jobs.....	129
5.3.4.1.2	GET NS Performance Monitoring Jobs with attribute-based filter	130
5.3.4.1.3	GET all NS Performance Monitoring Jobs with "all_fields" attribute selector	130
5.3.4.1.4	GET all NS Performance Monitoring Jobs with "exclude_default" attribute selector.....	130
5.3.4.1.5	GET all NS Performance Monitoring Jobs with "include" attribute selector	131
5.3.4.1.6	GET all NS Performance Monitoring Jobs with "exclude" attribute selector.....	131
5.3.4.1.7	GET NS Performance Monitoring Jobs with invalid attribute-based filter	132
5.3.4.1.8	GET NS Performance Monitoring Jobs with invalid resource endpoint	132
5.3.4.1.9	Create new NS Performance Monitoring Job.....	132
5.3.4.1.10	PUT all NS Performance Monitoring Jobs - Method not implemented.....	133
5.3.4.1.11	PATCH all NS Performance Monitoring Jobs - Method not implemented	133
5.3.4.1.12	DELETE all NS Performance Monitoring Jobs - Method not implemented	133
5.3.4.2	Individual PM Job Resource Endpoint.....	134
5.3.4.2.1	GET individual NS Performance Job	134
5.3.4.2.2	GET individual NS Performance Job with invalid resource identifier	134
5.3.4.2.3	DELETE Individual NS Performance Job with invalid resource identifier.....	134
5.3.4.2.4	POST Individual NS Performance Job - Method not implemented.....	135
5.3.4.2.5	PUT Individual NS Performance Job - Method not implemented.....	135
5.3.4.2.6	PATCH Individual NS Performance Job - Method not implemented	135
5.3.4.2.7	DELETE Individual NS Performance Job.....	136
5.3.4.3	Individual Performance Report Resource Endpoint.....	136
5.3.4.3.1	Get Individual Performance Report.....	136
5.3.4.3.2	Get Individual Performance Report with invalid resource endpoint	136
5.3.4.3.3	POST Individual Performance Report - Method not implemented	137
5.3.4.3.4	PUT Individual Performance Report - Method not implemented.....	137
5.3.4.3.5	PATCH Individual Performance Report - Method not implemented	137

5.3.4.3.6	DELETE Individual Performance Report - Method not implemented	138
5.3.4.4	Thresholds Resource Endpoint.....	138
5.3.4.4.1	GET All Performance Thresholds	138
5.3.4.4.2	GET Performance Thresholds with attribute-based filter	138
5.3.4.4.3	GET Performance Thresholds with invalid attribute-based filter	139
5.3.4.4.4	GET Performance Thresholds with invalid resource endpoint	139
5.3.4.4.5	Create new Performance Threshold.....	139
5.3.4.4.6	PUT Performance Thresholds - Method not implemented	140
5.3.4.4.7	PATCH Performance Thresholds - Method not implemented.....	140
5.3.4.4.8	DELETE Performance Thresholds - Method not implemented.....	140
5.3.4.5	Individual Threshold Resource Endpoint	141
5.3.4.5.1	GET Individual Threshold.....	141
5.3.4.5.2	GET Individual Threshold with invalid resource identifier	141
5.3.4.5.3	DELETE Individual Threshold with invalid resource identifier	141
5.3.4.5.4	POST Individual Threshold - Method not implemented	142
5.3.4.5.5	PUT Individual Threshold - Method not implemented.....	142
5.3.4.5.6	PATCH Individual Threshold - Method not implemented	142
5.3.4.5.7	DELETE Individual Threshold	143
5.3.4.6	Subscriptions Resource Endpoint.....	143
5.3.4.6.1	Get All NS Performance Subscriptions	143
5.3.4.6.2	Get NS Performance Subscriptions with attribute-based filter	143
5.3.4.6.3	Get NS Performance Subscriptions with invalid attribute-based filter	144
5.3.4.6.4	GET NS Performance Subscription with invalid resource endpoint	144
5.3.4.6.5	Create new NS Performance subscription	144
5.3.4.6.6	Create duplicated NS Performance subscription with NFVO not creating duplicated subscriptions	145
5.3.4.6.7	Create duplicated NS Performance subscription with NFVO creating duplicated subscriptions ..	145
5.3.4.6.8	PUT NS Performance Subscriptions - Method not implemented	145
5.3.4.6.9	PATCH NS Performance Subscriptions - Method not implemented	146
5.3.4.6.10	DELETE NS Performance Subscriptions - Method not implemented.....	146
5.3.4.7	Individual Subscription Resource Endpoint.....	146
5.3.4.7.1	GET Individual NS Performance Subscription	146
5.3.4.7.2	GET Individual NS Performance Subscription with invalid resource identifier	147
5.3.4.7.3	DELETE Individual NS Performance Subscription with invalid resource identifier	147
5.3.4.7.4	POST Individual NS Performance Subscription - Method not implemented	147
5.3.4.7.5	PUT Individual NS Performance Subscription - Method not implemented	148
5.3.4.7.6	PATCH Individual NS Performance Subscription - Method not implemented.....	148
5.3.4.7.7	DELETE Individual NS Performance Subscription	148
5.3.4.8	Notifications Endpoint.....	149
5.3.4.8.1	NS Performance Information Availability Notification	149
5.3.4.8.2	NS Threshold Crossed Notification.....	149
5.3.5	VNF Package Management interface	150
5.3.5.1	VNF Packages Resource Endpoint	150
5.3.5.1.1	GET all VNF Packages	150
5.3.5.1.2	GET VNF Packages with attribute-based filter	150
5.3.5.1.3	GET VNF Packages with invalid attribute-based filter	151
5.3.5.1.4	Get all VNF Packages with malformed authorization token	151
5.3.5.1.5	Get all VNF Packages without authorization token.....	151
5.3.5.1.6	GET VNF Packages with "all_fields" attribute selector	152
5.3.5.1.7	GET VNF Packages with "exclude_default" attribute selector	152
5.3.5.1.8	GET VNF Packages with "fields" attribute selector.....	153
5.3.5.1.9	GET VNF Packages with "exclude_fields" attribute selector	153
5.3.5.1.10	GET all VNF Packages with invalid resource endpoint	153
5.3.5.1.11	Create new VNF Package Resource	154
5.3.5.1.12	PUT all VNF Packages - Method not implemented	154
5.3.5.1.13	PATCH all VNF Packages - Method not implemented.....	154
5.3.5.1.14	DELETE all VNF Packages - Method not implemented.....	155
5.3.5.2	Individual VNF Package Resource Endpoint.....	155
5.3.5.2.1	GET Individual VNF Package.....	155
5.3.5.2.2	GET Individual VNF Package with invalid resource identifier.....	155
5.3.5.2.3	POST Individual VNF Package - Method not implemented	156
5.3.5.2.4	PUT Individual VNF Package - Method not implemented	156

5.3.5.2.5	Disable Individual VNF Package	156
5.3.5.2.6	Disable Individual VNF Package with conflict due to operational state DISABLED.....	157
5.3.5.2.7	Enable Individual VNF Package	157
5.3.5.2.8	Enable Individual VNF Package with conflict due to operational state ENABLED.....	157
5.3.5.2.9	DELETE Individual VNF Package	158
5.3.5.2.10	DELETE Individual VNF Package in operational state ENABLED.....	158
5.3.5.2.11	DELETE Individual VNF Package used for instantiated VNF instances.....	158
5.3.5.3	VNFD In Individual VNF Package Resource Endpoint.....	159
5.3.5.3.1	Get single file VNFD in Individual VNF Package in Plain Format	159
5.3.5.3.2	Get VNFD in Individual VNF Package in Zip Format.....	159
5.3.5.3.3	Get single file VNFD in Individual VNF Package in Plain or Zip Format.....	159
5.3.5.3.4	Get multi file VNFD in Individual VNF Package in Plain or Zip Format.....	160
5.3.5.3.5	Get multi file VNFD in Individual VNF Package in Plain Format	160
5.3.5.3.6	Get VNFD in Individual VNF Package with invalid resource identifier.....	160
5.3.5.3.7	Get VNFD in Individual VNF Package with conflict due to onboarding state	161
5.3.5.3.8	POST VNFD in Individual VNF Package - Method not implemented	161
5.3.5.3.9	PUT VNFD in Individual VNF Package - Method not implemented.....	161
5.3.5.3.10	PATCH VNFD in Individual VNF Package - Method not implemented	162
5.3.5.3.11	DELETE VNFD in Individual VNF Package - Method not implemented	162
5.3.5.4	VNF Package Content Resource Endpoint	162
5.3.5.4.1	GET Individual VNF Package Content	162
5.3.5.4.2	GET Individual VNF Package Content with Range Request and NFVO supporting Range Requests.....	163
5.3.5.4.3	GET Individual VNF Package Content with Range Request and NFVO not supporting Range Requests.....	163
5.3.5.4.4	GET Individual VNF Package Content with invalid Range Request	164
5.3.5.4.5	GET Individual VNF Package Content with invalid resource identifier	164
5.3.5.4.6	GET Individual VNF Package Content with conflict due to onboarding state	164
5.3.5.4.7	POST Individual VNF Package Content - Method not implemented.....	165
5.3.5.4.8	Upload VNF Package Content	165
5.3.5.4.9	Upload VNF Package Content with conflict due to onboarding state	165
5.3.5.4.10	PATCH Individual VNF Package Content - Method not implemented	166
5.3.5.4.11	DELETE Individual VNF Package Content - Method not implemented	166
5.3.5.5	Upload VNF Package From URI Resource Endpoint.....	166
5.3.5.5.1	Upload VNF Package Content from URI	166
5.3.5.5.2	Upload VNF Package Content from URI with conflict due to onboarding state.....	167
5.3.5.5.3	GET Individual VNF Package Content from URI - Method not implemented	167
5.3.5.5.4	PUT Individual VNF Package Content from URI - Method not implemented	167
5.3.5.5.5	PATCH Individual VNF Package Content from URI - Method not implemented.....	168
5.3.5.5.6	DELETE Individual VNF Package Content from URI - Method not implemented.....	168
5.3.5.6	Individual VNF Package Artifact Resource Endpoint	168
5.3.5.6.1	GET Individual VNF Package Artifact	168
5.3.5.6.2	GET Individual VNF Package Artifact in octet stream format	169
5.3.5.6.3	GET Individual VNF Package Artifact with Range Request and NFVO supporting Range Requests.....	169
5.3.5.6.4	GET Individual VNF Package Artifact with Range Request and NFVO not supporting Range Requests.....	169
5.3.5.6.5	GET Individual VNF Package Artifact with invalid Range Request	170
5.3.5.6.6	GET Individual VNF Package Artifact with invalid resource identifier	170
5.3.5.6.7	GET Individual VNF Package Artifact with conflict due to onboarding state	170
5.3.5.6.8	POST Individual VNF Package Artifact - Method not implemented.....	171
5.3.5.6.9	PUT Individual VNF Package Artifact - Method not implemented	171
5.3.5.6.10	PATCH Individual VNF Package Artifact - Method not implemented.....	171
5.3.5.6.11	DELETE Individual VNF Package Artifact - Method not implemented.....	172
5.3.5.7	Subscriptions Resource Endpoint.....	172
5.3.5.7.1	Get All VNF Package Subscriptions	172
5.3.5.7.2	Get VNF Package Subscriptions with attribute-based filter	172
5.3.5.7.3	Get VNF Package Subscriptions with invalid attribute-based filter	173
5.3.5.7.4	GET VNF Package Subscription with invalid resource endpoint.....	173
5.3.5.7.5	Create new VNF Package subscription	173
5.3.5.7.6	Create duplicated VNF Package subscription with NFVO not creating duplicated subscriptions.....	174

5.3.5.7.7	Create duplicated VNF Package subscription with NFVO creating duplicated subscriptions	174
5.3.5.7.8	PUT VNF Package Subscriptions - Method not implemented	174
5.3.5.7.9	PATCH VNF Package Subscriptions - Method not implemented.....	175
5.3.5.7.10	DELETE VNF Package Subscriptions - Method not implemented.....	175
5.3.5.8	Individual Subscription Resource Endpoint.....	175
5.3.5.8.1	GET Individual VNF Package Subscription.....	175
5.3.5.8.2	GET Individual VNF Package Subscription with invalid resource identifier.....	176
5.3.5.8.3	DELETE Individual VNF Package Subscription with invalid resource identifier	176
5.3.5.8.4	POST Individual VNF Package Subscription - Method not implemented	176
5.3.5.8.5	PUT Individual VNF Package Subscription - Method not implemented	177
5.3.5.8.6	PATCH Individual VNF Package Subscription - Method not implemented	177
5.3.5.8.7	DELETE Individual VNF Package Subscription	177
5.3.5.9	Notifications Endpoint	178
5.3.5.9.1	VNF Package Onboarding Notification.....	178
5.3.5.9.2	VNF Package Operational State Change Notification.....	178
5.3.5.9.3	VNF Package Deletion Notification.....	179
6	Ve-Vnfm Reference Point	179
6.1	General	179
6.2	Test configuration.....	179
6.3	Test Descriptions.....	180
6.3.1	VNF Configuration Interface.....	180
6.3.1.1	Configuration Resource Endpoint.....	180
6.3.1.1.1	Set a new VNF Configuration	180
6.3.1.1.2	Get information about a VNF configuration.....	180
6.3.1.1.3	Get information about a VNF configuration with usage of HTTP Etag	180
6.3.1.1.4	Set a new VNF Configuration - HTTP Etag precondition unsuccessful	181
6.3.1.1.5	POST VNF Configuration - Method not implemented.....	181
6.3.1.1.6	PUT VNF Configuration - Method not implemented.....	181
6.3.1.1.7	DELETE VNF Configuration - Method not implemented	182
6.3.2	VNF Indicator Interface.....	182
6.3.2.1	VNF Indicators Resource Endpoint	182
6.3.2.1.1	Get all VNF Indicators	182
6.3.2.1.2	Get VNF Indicators with attribute-based filter	182
6.3.2.1.3	Get VNF Indicators with invalid attribute-based filter	183
6.3.2.1.4	Get all VNF Indicators with malformed authorization token	183
6.3.2.1.5	Get all VNF Indicators without authorization token.....	183
6.3.2.1.6	Get all VNF Indicators with expired or revoked authorization token.....	184
6.3.2.1.7	Get all VNF Indicators with invalid resource endpoint	184
6.3.2.1.8	POST all VNF Indicators - Method not implemented	184
6.3.2.1.9	PUT all VNF Indicators - Method not implemented	185
6.3.2.1.10	PATCH all VNF Indicators - Method not implemented.....	185
6.3.2.1.11	DELETE all VNF Indicators - Method not implemented.....	185
6.3.2.2	VNF Indicators related to a VNF Instance Resource Endpoint.....	186
6.3.2.2.1	Get Indicators for VNF Instance.....	186
6.3.2.2.2	GET Indicators for VNF Instance with attribute-based filter	186
6.3.2.2.3	Get Indicators for VNF Instance with invalid attribute-based filter	187
6.3.2.2.4	Get Indicators for VNF Instance with invalid resource identifier	187
6.3.2.2.5	POST Indicators for VNF instance - Method not implemented	187
6.3.2.2.6	PUT Indicators for VNF instance - Method not implemented.....	188
6.3.2.2.7	PATCH Indicators for VNF instance - Method not implemented.....	188
6.3.2.2.8	DELETE Indicators for VNF instance - Method not implemented	188
6.3.2.3	Individual VNF Indicator Resource Endpoint.....	189
6.3.2.3.1	Get Individual Indicator for VNF Instance.....	189
6.3.2.3.2	Get Individual Indicator for VNF Instance with invalid indicator identifier	189
6.3.2.3.3	POST Individual VNF Indicator - Method not implemented	189
6.3.2.3.4	PUT Individual VNF Indicator - Method not implemented	190
6.3.2.3.5	PATCH Individual VNF Indicator - Method not implemented.....	190
6.3.2.3.6	DELETE Individual VNF Indicator - Method not implemented.....	190
6.3.2.4	Subscriptions Resource Endpoint.....	191
6.3.2.4.1	GET VNF Indicators Subscriptions.....	191
6.3.2.4.2	GET VNF Indicators Subscriptions with attribute-based filter	191

6.3.2.4.3	GET VNF Indicators Subscriptions with invalid attribute-based filter	192
6.3.2.4.4	GET VNF Indicators Subscriptions with invalid resource endpoint	192
6.3.2.4.5	Create new VNF indicator subscription	192
6.3.2.4.6	PUT VNF Indicator Subscriptions - Method not implemented	193
6.3.2.4.7	PATCH VNF Indicator Subscriptions - Method not implemented.....	193
6.3.2.4.8	DELETE VNF Indicator Subscriptions - Method not implemented.....	193
6.3.2.5	Individual Subscription Resource Endpoint.....	194
6.3.2.5.1	GET Individual VNF Indicator Subscription.....	194
6.3.2.5.2	GET Individual VNF Indicator Subscription with invalid resource identifier.....	194
6.3.2.5.3	DELETE Individual VNF Indicator Subscription	194
6.3.2.5.4	DELETE Individual VNF Indicator Subscription with invalid resource identifier	195
6.3.2.5.5	PUT Individual VNF Indicator Subscription - Method not implemented	195
6.3.2.5.6	PATCH Individual VNF Indicator Subscription - Method not implemented.....	195
6.3.2.5.7	POST Individual VNF Indicator Subscription - Method not implemented	196
6.3.2.6	Notification Endpoint.....	196
6.3.2.6.1	VNF Indicator Value Change Notification.....	196
6.3.3	VNF Performance Management Interface	197
6.3.3.1	PM Jobs Resource Endpoint	197
6.3.3.1.1	GET all VNF Performance Monitoring Jobs.....	197
6.3.3.1.2	GET VNF Performance Monitoring Jobs with attribute-based filter.....	197
6.3.3.1.3	GET all VNF Performance Monitoring Jobs with all_fields attribute selector.....	198
6.3.3.1.4	GET all VNF Performance Monitoring Jobs with exclude_default attribute selector	198
6.3.3.1.5	GET all VNF Performance Monitoring Jobs with fields attribute selector	198
6.3.3.1.6	GET all VNF Performance Monitoring Jobs with exclude_fields attribute selector	199
6.3.3.1.7	GET VNF Performance Monitoring Jobs with invalid attribute-based filter	199
6.3.3.1.8	GET VNF Performance Monitoring Jobs with invalid resource endpoint	199
6.3.3.1.9	Create new VNF Performance Monitoring Job	200
6.3.3.1.10	PUT all VNF Performance Monitoring Jobs - Method not implemented.....	200
6.3.3.1.11	PATCH all VNF Performance Monitoring Jobs - (Method not implemented).....	200
6.3.3.1.12	DELETE all VNF Performance Monitoring Jobs - Method not implemented	201
6.3.3.2	Individual PM Job Resource Endpoint.....	201
6.3.3.2.1	GET individual VNF Performance Job	201
6.3.3.2.2	GET individual VNF Performance Job with invalid resource identifier	201
6.3.3.2.3	DELETE Individual VNF Performance Job.....	202
6.3.3.2.4	DELETE Individual VNF Performance Job with invalid resource identifier	202
6.3.3.2.5	POST Individual VNF Performance Job - Method not implemented.....	202
6.3.3.2.6	PUT Individual VNF Performance Job - Method not implemented	203
6.3.3.2.7	PATCH Individual VNF Performance Job - Method not implemented	203
6.3.3.3	Individual Performance Report Resource Endpoint.....	203
6.3.3.3.1	Get Individual Performance Report.....	203
6.3.3.3.2	Get Individual Performance Report with invalid resource endpoint	204
6.3.3.3.3	POST Individual Performance Report - Method not implemented	204
6.3.3.3.4	PUT Individual Performance Report - Method not implemented.....	204
6.3.3.3.5	PATCH Individual Performance Report - Method not implemented	205
6.3.3.3.6	DELETE Individual Performance Report - Method not implemented	205
6.3.3.4	Thresholds Resource Endpoint.....	206
6.3.3.4.1	GET All Performance Thresholds	206
6.3.3.4.2	GET Performance Thresholds with attribute-based filter	206
6.3.3.4.3	GET Performance Thresholds with invalid attribute-based filter	206
6.3.3.4.4	GET Performance Thresholds with invalid resource endpoint.....	207
6.3.3.4.5	Create new Performance Threshold.....	207
6.3.3.4.6	PUT Performance Thresholds - Method not implemented	207
6.3.3.4.7	PATCH Performance Thresholds - Method not implemented.....	208
6.3.3.4.8	DELETE Performance Thresholds – Method not implemented.....	208
6.3.3.5	Individual Threshold Resource Endpoint.....	208
6.3.3.5.1	GET Individual Threshold.....	208
6.3.3.5.2	GET Individual Threshold with invalid resource identifier.....	209
6.3.3.5.3	DELETE Individual Threshold	209
6.3.3.5.4	DELETE Individual Threshold with invalid resource identifier	209
6.3.3.5.5	POST Individual Threshold - Method not implemented	210
6.3.3.5.6	PUT Individual Threshold - Method not implemented.....	210
6.3.3.5.7	PATCH Individual Threshold - Method not implemented.....	210

6.3.3.6	Subscriptions Resource Endpoint.....	211
6.3.3.6.1	GET all VNF Performance Subscriptions	211
6.3.3.6.2	GET VNF Performance Subscriptions with attribute-based filter	211
6.3.3.6.3	GET VNF Performance Management Subscriptions with invalid attribute-based filter	212
6.3.3.6.4	GET VNF Performance Subscriptions with invalid resource endpoint	212
6.3.3.6.5	Create new VNF Performance subscription	212
6.3.3.6.6	Create duplicated VNF Performance subscription with VNFM not creating duplicated subscriptions	213
6.3.3.6.7	Create duplicated VNF Performance subscription with VNFM creating duplicated subscriptions	213
6.3.3.6.8	PUT VNF Performance Subscriptions - Method not implemented	214
6.3.3.6.9	PATCH VNF Performance Subscriptions - Method not implemented.....	214
6.3.3.6.10	DELETE VNF Performance Subscriptions - Method not implemented.....	214
6.3.3.7	Individual Subscription Resource Endpoint.....	215
6.3.3.7.1	GET Individual VNF Performance Subscription.....	215
6.3.3.7.2	GET Individual VNF Performance Subscription with invalid resource identifier.....	215
6.3.3.7.3	DELETE Individual VNF Performance Subscription	215
6.3.3.7.4	DELETE Individual VNF Performance Subscription with invalid resource identifier	216
6.3.3.7.5	POST Individual VNF Performance Subscription - Method not implemented	216
6.3.3.7.6	PUT Individual VNF Performance Subscription - Method not implemented	216
6.3.3.7.7	PATCH Individual VNF Performance Subscription - Method not implemented	217
6.3.3.8	Notification Endpoint.....	217
6.3.3.8.1	VNF Performance Information Availability Notification.....	217
6.3.3.8.2	VNF Threshold Crossed Notification	218
6.3.4	VNF Fault Management interface	218
6.3.4.1	Alarms.....	218
6.3.4.1.1	POST Alarms - Method not implemented	218
6.3.4.1.2	GET information about multiple alarms.....	218
6.3.4.1.3	GET information about multiple alarms with filters.....	219
6.3.4.1.4	GET information about multiple alarms Bad Request Invalid attribute-based filtering parameters	219
6.3.4.1.5	PUT Alarms - Method not implemented	219
6.3.4.1.6	PATCH Alarms - Method not implemented.....	220
6.3.4.1.7	DELETE Alarms - Method not implemented.....	220
6.3.4.2	Individual Alarm	220
6.3.4.2.1	POST Alarm - Method not implemented.....	220
6.3.4.2.2	GET information about an individual alarm.....	221
6.3.4.2.3	PUT Alarm - Method not implemented.....	221
6.3.4.2.4	PATCH Alarm.....	221
6.3.4.2.5	PATCH Alarm - Precondition failed	222
6.3.4.2.6	PATCH Alarm - Conflict	222
6.3.4.2.7	DELETE Alarm - Method not implemented	222
6.3.4.3	Escalate the perceived severity	223
6.3.4.3.1	Escalate the perceived severity.....	223
6.3.4.3.2	GET Escalate the perceived severity - Method not implemented.....	223
6.3.4.3.3	PUT Escalate the perceived severity - Method not implemented.....	223
6.3.4.3.4	PATCH Escalate the perceived severity - Method not implemented	224
6.3.4.4	Subscriptions.....	224
6.3.4.4.1	Create a new subscription.....	224
6.3.4.4.2	Create a new Subscription - DUPLICATION	225
6.3.4.4.3	Create a new Subscription - NO-DUPLICATION	225
6.3.4.4.4	GET Subscriptions.....	225
6.3.4.4.5	GET Subscription - Filter	226
6.3.4.4.6	GET subscriptions - Bad Request Invalid attribute-based filtering parameters.....	226
6.3.4.4.7	PUT subscriptions - Method not implemented	226
6.3.4.4.8	PATCH subscriptions - Method not implemented	227
6.3.4.4.9	DELETE subscriptions - Method not implemented	227
6.3.4.5	Individual Subscription	227
6.3.4.5.1	POST Individual Subscription - Method not implemented	227
6.3.4.5.2	GET Information about an individual subscription	228
6.3.4.5.3	PUT an individual subscription - Method not implemented.....	228
6.3.4.5.4	PATCH an individual subscription - Method not implemented	228