SOL003 - VNF Lifecycle Operation Granting interface

Overview

SOL003 - VNF Lifecycle Operation Granting interface

IMPORTANT	Please note that this file might be not aligned to the current version of the ETSI Group Specification it refers to. In case of discrepancies the published ETSI Group Specification takes precedence.

In clause 4.3.2 of ETSI GS NFV-SOL 003 v2.4.1, an attribute-based filtering mechanism is defined. This mechanism is currently not included in the corresponding OpenAPI design for this GS version. Changes to the attribute-based filtering mechanism are being considered in v2.5.1 of this GS for inclusion in the corresponding future ETSI NFV OpenAPI design. Please report bugs to https://forge.etsi.org/bugzilla/buglist.cgi?component=Nfv-Openapis&list_id=61&product=NFV&resolution=

Version information

Version : 1.1.0

License information

License : ETSI Forge copyright notice
License URL : https://forge.etsi.org/etsi-forge-copyright-notice.txt
Terms of service : null

URI scheme

BasePath : /grant/v1
Schemes : HTTPS

Consumes

- `application/json`

Produces

- `application/json`

External Docs

Description : ETSI GS NFV-SOL 003 V2.4.1
URL : http://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/003/02.04.01_60/gs_NFV-SOL003v020401p.pdf

Paths

POST /grants

Description

Grant Lifecycle Operation

The POST method requests a grant for a particular VNF lifecycle operation.

Parameters

Type	Name	Description	Schema
Header	**Accept** *required*	Content-Types that are acceptable for the response. Reference: IETF RFC 7231	string
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string
Header	**Content-Type** *required*	The MIME type of the body of the request. Reference: IETF RFC 7231	string
Body	**GrantRequest** *required*		GrantRequest

GrantRequest

Name	Description	Schema
_links *required*	Links to resources related to this request.	_links
addResources *optional*	List of resource definitions in the VNFD for resources to be added by the LCM operation which is related to this grant request, with one entry per resource. If the granting request is for InstantiateVNF, either instantiationLevel or addResources shall be present.	< addResources > array
additionalParams *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

Name	Description	Schema
flavourId *optional*	An identifier with the intention of being globally unique.	string
instantiationLevelId *optional*	An identifier with the intention of being globally unique.	string
isAutomaticInvocation *required*	Set to true if this VNF LCM operation occurrence has been triggered by an automated procedure inside the VNFM (i.e. ScaleVnf / ScaleVnfToLevel triggered by auto-scale, or HealVnf triggered by auto-heal). Set to false otherwise.	boolean
operation *required*	Value | Description ——|——— INSTANTIATE | Represents the "Instantiate VNF" LCM operation. SCALE | Represents the "Scale VNF" LCM operation. SCALE_TO_LEVEL | Represents the "Scale VNF to Level" LCM operation. CHANGE_FLAVOUR | Represents the "Change VNF Flavour" LCM operation. TERMINATE | Represents the "Terminate VNF" LCM operation. HEAL | Represents the "Heal VNF" LCM operation. OPERATE | Represents the "Operate VNF" LCM operation. CHANGE_EXT_CONN | Represents the "Change external VNF connectivity" LCM operation. MODIFY_INFO | Represents the "Modify VNF Information" LCM operation.	enum (INSTANTIATE, SCALE, SCALE_TO_LEVEL, CHANGE_FLAVOUR, TERMINATE, HEAL, OPERATE, CHANGE_EXT_CONN, MODIFY_INFO)
placementConstraints *optional*	Placement constraints that the VNFM may send to the NFVO in order to influence the resource placement decision. If sent, the NFVO shall take the constraints into consideration when making resource placement decisions, and shall reject the grant if they cannot be honoured. The affinity/anti-affinity rules defined in the VNFD , and the placement constraints in the GrantVnfLifecycleOperation as defined in this clause should be conflict-free. In case of conflicts, the placement constraints in the GrantVnfLifecycleOperation shall take precedence. Passing constraints allows the VNFM or the lifecycle management scripts to influence resource placement decisions by the NFVO to ensure VNF properties such as performance or fault tolerance.	< placementConstraints > array
removeResources *optional*	Provides the definitions of resources to be removed by the LCM operation which is related to this grant request, with one entry per resource.	< removeResources > array

Name	Description	Schema
tempResources *optional*	List of resource definitions in the VNFD for resources to be temporarily instantiated during the runtime of the LCM operation which is related to this grant request, with one entry per resource. The NFVO will assume that the VNFM will be responsible to both allocate and release the temporary resource during the runtime of the LCM operation. This means, the resource can be allocated and consumed after the "start" notification for the LCM operation is sent by the VNFM, and the resource will be released before the "result" notification of the VNF LCM operation is sent by the VNFM.	< tempResources > array
updateResources *optional*	Provides the definitions of resources to be modified by the LCM operation which is related to this grant request, with one entry per resource.	< updateResources > array
vimConstraints *optional*	Used by the VNFM to require that multiple resources are managed through the same VIM connection. If sent, the NFVO shall take the constraints into consideration when making VIM selection decisions, and shall reject the grant if they cannot be honoured. This attribute shall be supported if VNF-related Resource Management in direct mode is applicable. The applicability and further details of this attribute for indirect mode are left for future specification.	< vimConstraints > array
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfLcmOpOccId *required*	An identifier with the intention of being globally unique.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string

_links

Name	Description	Schema
vnfInstance *required*	This type represents a link to a resource.	vnfInstance
vnfLcmOpOcc *required*	This type represents a link to a resource.	vnfLcmOpOcc

vnfInstance

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

vnfLcmOpOcc

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

addResources

Name	Description	Schema
id *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM.	resource
resourceTemplateId *optional*	An identifier that is unique within a VNF descriptor.	string
type *required*	Type of the resource definition referenced. Permitted values: * COMPUTE * VL * STORAGE * LINKPORT	enum (COMPUTE, VL, STORAGE, LINKPORT)
vduId *optional*	An identifier that is unique within a VNF descriptor.	string

resource

Name	Description	Schema
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *required*	An identifier with the intention of being globally unique.	string
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider.	string

placementConstraints

Name	Description	Schema
affinityOrAntiAffinity *required*	The type of the constraint. Permitted values: * AFFINITY * ANTI_AFFINITY	enum (AFFINITY, ANTI_AFFINITY)
resource *required*	References to resources in the constraint rule.	< resource > array
scope *required*	The scope of the placement constraint indicating the category of the "place" where the constraint applies. Permitted values: * NFVI_POP * ZONE * ZONE_GROUP * NFVI_NODE	enum (NFVI_POP, ZONE, ZONE_GROUP, NFVI_NODE)

resource

Name	Description	Schema
idType *required*	The type of the identifier. Permitted values: * RES_MGMT: Resource-management-level identifier; this identifier is managed by the VIM in the direct mode of VNF-related resource management, and is managed by the NFVO in the indirect mode) * GRANT: Reference to the identifier of a "ResourceDefinition" structure in the "GrantRequest" structure.	enum (RES_MGMT, GRANT)
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string

removeResources

Name	Description	Schema
id *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM.	resource
resourceTemplateId *optional*	An identifier that is unique within a VNF descriptor.	string
type *required*	Type of the resource definition referenced. Permitted values: * COMPUTE * VL * STORAGE * LINKPORT	enum (COMPUTE, VL, STORAGE, LINKPORT)
vduId *optional*	An identifier that is unique within a VNF descriptor.	string

resource

Name	Description	Schema
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectionId *required*	An identifier with the intention of being globally unique.	string
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider.	string

tempResources

Name	Description	Schema
id *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM.	resource
resourceTemplateId *optional*	An identifier that is unique within a VNF descriptor.	string
type *required*	Type of the resource definition referenced. Permitted values: * COMPUTE * VL * STORAGE * LINKPORT	enum (COMPUTE, VL, STORAGE, LINKPORT)
vduId *optional*	An identifier that is unique within a VNF descriptor.	string

resource

Name	Description	Schema
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectio nId *required*	An identifier with the intention of being globally unique.	string
vimLevelReso urceType *optional*	Type of the resource in the scope of the VIM or the resource provider.	string

updateResources

Name	Description	Schema
id *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM.	resource
resourceTemp lateId *optional*	An identifier that is unique within a VNF descriptor.	string
type *required*	Type of the resource definition referenced. Permitted values: * COMPUTE * VL * STORAGE * LINKPORT	enum (COMPUTE, VL, STORAGE, LINKPORT)
vduId *optional*	An identifier that is unique within a VNF descriptor.	string

resource

Name	Description	Schema
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectionId *required*	An identifier with the intention of being globally unique.	string
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider.	string

vimConstraints

Name	Description	Schema
resource *required*	References to resources in the constraint rule. The NFVO shall ensure that all resources in this list are managed through the same VIM connection. If "sameResourceGroup" is set to true, the NFVO shall further ensure that all resources in this list are part of the same infrastructure resource group in that VIM connection.	< resource > array
sameResourceGroup *optional*	If present and set to true, this signals that the constraint applies not only to the same VIM connection, but also to the same infrastructure resource group.	boolean

resource

Name	Description	Schema
idType *required*	The type of the identifier. Permitted values: * RES_MGMT: Resource-management-level identifier; this identifier is managed by the VIM in the direct mode of VNF-related resource management, and is managed by the NFVO in the indirect mode) * GRANT: Reference to the identifier of a "ResourceDefinition" structure in the "GrantRequest" structure.	enum (RES_MGMT, GRANT)
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string

Responses

HTTP Code	Description	Schema
201	Created The grant was created successfully (synchronous mode). A representation of the created "Individual grant" resource shall be returned in the response body. The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource just created. **Headers** : Content-Type (string) : The MIME type of the body of the response. Location (string (url)) : The resource URI of the created VNF instance. WWW-Authenticate (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 201
202	Accepted The request was accepted for processing, but the processing has not been completed. It is expected to take some time to create the grant (asynchronous mode). The response body shall be empty. The HTTP response shall include a "Location" HTTP header that indicates the URI of the "Individual grant" resource that will be created once the granting decision has been made. **Headers** : Location (string (url)) : The resource URI of the created VNF instance. WWW-Authenticate (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	No Content

HTTP Code	Description	Schema
400	Bad Request If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or a syntactically incorrect payload body), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. — If the request contains a malformed access token, the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. — If there is an application error related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond with this response code.The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : Content-Type (string) : The MIME type of the body of the response. WWW-Authenticate (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 400
401	Unauthorized If the request contains no access token even though one is required, or if the request contains an authorization token that is invalid (e.g. expired or revoked), the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. **Headers** : Content-Type (string) : The MIME type of the body of the response. WWW-Authenticate (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 401

HTTP Code	Description	Schema
403	Forbidden The grant was rejected. A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 403
404	Not Found If the API producer did not find a current representation for the resource addressed by the URI passed in the request, or is not willing to disclose that one exists, it shall respond with this response code. The "ProblemDetails" structure may be provided, including in the "detail" attribute information about the source of the problem, e.g. a wrong resource URI variable. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 404
405	Method Not Allowed If a particular HTTP method is not supported for a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 405
406	Not Acceptable If the "Accept" HTTP header does not contain at least one name of a content type that is acceptable to the API producer, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 406
416	Requested Range Not Satisfiable This code is returned if the requested byte range in the Range HTTP header is not present in the requested resource. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 416

HTTP Code	Description	Schema
422	Unprocessable Entity If the payload body of a request contains syntactically correct data (e.g. well-formed JSON) but the data cannot be processed (e.g. because it fails validation against a schema), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. NOTE 2: This error response code is only applicable for methods that have a request body. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 422
500	Internal Server Error If there is an application error not related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond withthis response code. The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 500
503	Service Unavailable If the API producer encounters an internal overload situation of itself or of a system it relies on, it should respond with this response code, following the provisions in IETF RFC 7231 [13] for the use of the "Retry-After" HTTP header and for the alternative to refuse the connection. The "ProblemDetails" structure may be omitted. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 503

Response 201

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
addResources *optional*	List of resources that are approved to be added, with one entry per resource.	< addResources > array
additionalParams *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

Name	Description	Schema
computeReservationId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
extManagedVirtualLinks *optional*	Information about internal VLs that are managed by other entities than the VNFM. The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM. External and/or externally-managed internal VLs can be passed in VNF lifecycle management operation requests such as InstantiateVnf or ChangeVnfFlavor, and/or in the grant response. The NFVO may choose to override in the grant response external and/or externally-managed VL instances that have been passed previously in the associated VNF lifecycle management request, if the lifecycle management request has originated from the NFVO itself.	< extManagedVirtualLinks > array
extVirtualLinks *optional*	Information about external VLs to connect the VNF to. External and/or externally-managed internal VLs can be passed in VNF lifecycle management operation requests such as InstantiateVnf or ChangeVnfFlavor, and/or in the grant response. The NFVO may choose to override in the grant response external and/or externally-managed VL instances that have been passed previously in the associated VNF lifecycle management request, if the lifecycle management request has originated from the NFVO itself.	< extVirtualLinks > array
id *required*	An identifier with the intention of being globally unique.	string
networkReservationId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
removeResources *optional*	List of resources that are approved to be removed, with one entry per resource.	< removeResources > array

Name	Description	Schema
storageReservationId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
tempResources *optional*	List of resources that are approved to be temporarily instantiated during the runtime of the lifecycle operation, with one entry per resource.	< tempResources > array
updateResources *optional*	List of resources that are approved to be modified, with one entry per resource.	< updateResources > array
vimAssets *optional*	Information about assets for the VNF that are managed by the NFVO in the VIM, such as software images and virtualised compute resource flavours. This attribute is not intended for the modification of vimAssets entries passed earlier. Modification of VIM assets during the lifetime of a VNF instance is not necessary, since it is expected that all applicable assets have been on boarded into the VIM before the VNF is instantiated.	vimAssets
vimConnections *optional*	Provides information regarding VIM connections that are approved to be used by the VNFM to allocate resources, and provides parameters of these VIM connections. The VNFM shall update the " vimConnectionInfo" attribute of the "VnfInstance" structure by adding unknown entries received in this attribute. This attribute is not intended for the modification of VimConnection entries passed earlier; for that, the VnfInfoModificationRequest structure shall be used. This attribute shall only be supported when VNF-related Resource Management in direct mode is applicable. In direct mode, this parameter shall be absent if the VIM information was configured to the VNFM in another way, present otherwise. This interface allows to signal the use of multiple VIMs per VNF. However, due to the partial support of this feature in the present release, it is recommended in the present document that the number of entries in the "vims" attribute in the Grant is not greater than 1.	< vimConnections > array
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vnfLcmOpOcc Id *required*	An identifier with the intention of being globally unique.	string
zoneGroups *optional*	Information about groups of resource zones that are related and that the NFVO has chosen to fulfil a zoneGroup constraint in the GrantVnfLifecycleOperation request. This information confirms that the NFVO has honoured the zoneGroup constraints that were passed as part of "placementConstraints" in the GrantRequest.	< zoneGroups > array
zones *optional*	Identifies resource zones where the resources are approved to be allocated by the VNFM.	< zones > array

_links

Name	Description	Schema
self *required*	This type represents a link to a resource.	self
vnfInstance *required*	This type represents a link to a resource.	vnfInstance
vnfLcmOpOcc *required*	This type represents a link to a resource.	vnfLcmOpOcc

self

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

vnfInstance

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

vnfLcmOpOcc

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

addResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefin itionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGrou pId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

extManagedVirtualLinks

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string
virtualLinkDescId *required*	An identifier that is unique within a VNF descriptor.	string

extVirtualLinks

Name	Description	Schema
extCps *required*	External CPs of the VNF to be connected to this external VL.	< extCps > array
extLinkPorts *optional*	Externally provided link ports to be used to connect external connection points to this external VL. If this attribute is not present, the VNFM shall create the link ports on the external VL.	< extLinkPorts > array
id *required*	An identifier with the intention of being globally unique.	string
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string

extCps

Name	Description	Schema
cpConfig *optional*	List of instance data that need to be configured on the CP instances created from the respective CPD.	< cpConfig > array
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

cpConfig

Name	Description	Schema
cpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpProtocolDat a *optional*	Parameters for configuring the network protocols on the link port that connects the CP to a VL. The following conditions apply to the attributes "linkPortId" and "cpProtocolData": * The "linkPortId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by cpInstanceId. * At least one of these attributes shall be present for a to-be-created external CP instance or an existing external CP instance. * If the "linkPortId" attribute is absent, the VNFM shall create a link port. * If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port, and the VNFM can use means outside the scope of the present document to obtain the pre-configured address information for the connection point from the resource representing the link port. * If both "cpProtocolData" and "linkportId" are provided, the API consumer shall ensure that the cpProtocolData can be used with the pre-created link port referenced by "linkPortId".	< cpProtocolData > array
linkPortId *optional*	An identifier with the intention of being globally unique.	string

cpProtocolData

Name	Description	Schema
ipOverEthern et *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet

Name	Description	Schema
layerProtocol *required*	Identifier of layer(s) and protocol(s). This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
ipAddresses *optional*	List of IP addresses to assign to the CP instance. Each entry represents IP address data for fixed or dynamic IP address assignment per subnet. If this attribute is not present, no IP address shall be assigned.	< ipAddresses > array
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range to be used, e.g. in case of egress connections. In case this attribute is present, IP addresses from the range will be used.	addressRange
fixedAddresses *optional*	Fixed addresses to assign (from the subnet defined by "subnetId" if provided). Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.	< string (IP) > array
numDynamic Addresses *optional*	Number of dynamic addresses to assign (from the subnet defined by "subnetId" if provided). Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.	integer
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
type *required*	The type of the IP addresses. Permitted values: IPV4, IPV6.	enum (IPV4, IPV6)

addressRange

Name	Description	Schema
maxAddress *required*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
minAddress *required*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)

extLinkPorts

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM.	resourceHandle

resourceHandle

Name	Description	Schema
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *required*	An identifier with the intention of being globally unique.	string
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider.	string

removeResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefin itionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGrou pId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

tempResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefin itionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGrou pId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

updateResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefin itionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGrou pId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

vimAssets

Name	Description	Schema
computeResourceFlavours *optional*	Mappings between virtual compute descriptors defined in the VNFD and compute resource flavours managed in the VIM.	< computeResourceFlavours > array
softwareImages *optional*	Mappings between software images defined in the VNFD and software images managed in the VIM.	< softwareImages > array

computeResourceFlavours

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string
vimFlavourId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
vnfdVirtualComputeDescId *required*	An identifier that is unique within a VNF descriptor.	string

softwareImages

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string
vimSoftwareImageId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string

Name	Description	Schema
vnfdSoftwareI mageId *required*	An identifier that is unique within a VNF descriptor.	string

vimConnections

Name	Description	Schema
accessInfo *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object
extra *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object
id *required*	An identifier with the intention of being globally unique.	string
interfaceInfo *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object
vimId *optional*	An identifier with the intention of being globally unique.	string
vimType *required*	Discriminator for the different types of the VIM information. The value of this attribute determines the structure of the "interfaceInfo" and "accessInfo" attributes, based on the type of the VIM. The set of permitted values is expected to change over time as new types or versions of VIMs become available. The ETSI NFV registry of VIM-related information provides access to information about VimConnectionInfo definitions for various VIM types. The structure of the registry is defined in Annex C of SOL003.	string

zoneGroups

Name	Description	Schema
zoneId *required*	References of identifiers of "ZoneInfo" structures, each of which provides information about a resource zone that belongs to this group.	< string > array

zones

Name	Description	Schema
id *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string
zoneId *required*	An identifier with the intention of being globally unique.	string

Response 400

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 401

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 403

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 404

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 405

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 406

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 416

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 422

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 500

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 503

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

GET /grants/{grantId}

Description

Grant Lifecycle Operation

The GET method retrieves information about a specific grant by reading an individual grant resource.

Parameters

Type	Name	Description	Schema
Header	**Accept** *required*	Content-Types that are acceptable for the response. Reference: IETF RFC 7231	string
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string
Path	**grantId** *required*	Identifier of the grant. This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request granting a new VNF lifecycle operation. It can also be retrieved from the "id" attribute in the payload body of that response.	string

Responses

HTTP Code	Description	Schema
200	OK The grant was read successfully. A representation of the "individual grant" resource shall be returned in the response body. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	[Response 200](#)
202	Accepted The process of creating the grant is ongoing, no grant is available yet. The response body shall be empty. **Headers** : `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	No Content
400	Bad Request If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or a syntactically incorrect payload body), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. — If the request contains a malformed access token, the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. — If there is an application error related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond with this response code.The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	[Response 400](#)

HTTP Code	Description	Schema
401	Unauthorized If the request contains no access token even though one is required, or if the request contains an authorization token that is invalid (e.g. expired or revoked), the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 401
403	Forbidden The grant was rejected. A ProblemDetails structure shall be included in the response to provide more details about the rejection in the "details" attribute. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 403
404	Not Found If the API producer did not find a current representation for the resource addressed by the URI passed in the request, or is not willing to disclose that one exists, it shall respond with this response code. The "ProblemDetails" structure may be provided, including in the "detail" attribute information about the source of the problem, e.g. a wrong resource URI variable. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 404
405	Method Not Allowed If a particular HTTP method is not supported for a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 405

HTTP Code	Description	Schema
406	Not Acceptable If the "Accept" HTTP header does not contain at least one name of a content type that is acceptable to the API producer, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 406
416	Requested Range Not Satisfiable This code is returned if the requested byte range in the Range HTTP header is not present in the requested resource. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 416
422	Unprocessable Entity If the payload body of a request contains syntactically correct data (e.g. well-formed JSON) but the data cannot be processed (e.g. because it fails validation against a schema), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. NOTE 2: This error response code is only applicable for methods that have a request body. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 422
500	Internal Server Error If there is an application error not related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond withthis response code. The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 500
503	Service Unavailable If the API producer encounters an internal overload situation of itself or of a system it relies on, it should respond with this response code, following the provisions in IETF RFC 7231 [13] for the use of the "Retry-After" HTTP header and for the alternative to refuse the connection. The "ProblemDetails" structure may be omitted. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.	Response 503

Response 200

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
addResources *optional*	List of resources that are approved to be added, with one entry per resource.	< addResources > array
additionalParams *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object
computeReservationId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
extManagedVirtualLinks *optional*	Information about internal VLs that are managed by other entities than the VNFM. The indication of externally-managed internal VLs is needed in case networks have been pre-configured for use with certain VNFs, for instance to ensure that these networks have certain properties such as security or acceleration features, or to address particular network topologies. The present document assumes that externally-managed internal VLs are managed by the NFVO and created towards the VIM. External and/or externally-managed internal VLs can be passed in VNF lifecycle management operation requests such as InstantiateVnf or ChangeVnfFlavor, and/or in the grant response. The NFVO may choose to override in the grant response external and/or externally-managed VL instances that have been passed previously in the associated VNF lifecycle management request, if the lifecycle management request has originated from the NFVO itself.	< extManagedVirtualLinks > array
extVirtualLinks *optional*	Information about external VLs to connect the VNF to. External and/or externally-managed internal VLs can be passed in VNF lifecycle management operation requests such as InstantiateVnf or ChangeVnfFlavor, and/or in the grant response. The NFVO may choose to override in the grant response external and/or externally-managed VL instances that have been passed previously in the associated VNF lifecycle management request, if the lifecycle management request has originated from the NFVO itself.	< extVirtualLinks > array

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
networkReservationId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
removeResources *optional*	List of resources that are approved to be removed, with one entry per resource.	< removeResources > array
storageReservationId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
tempResources *optional*	List of resources that are approved to be temporarily instantiated during the runtime of the lifecycle operation, with one entry per resource.	< tempResources > array
updateResources *optional*	List of resources that are approved to be modified, with one entry per resource.	< updateResources > array
vimAssets *optional*	Information about assets for the VNF that are managed by the NFVO in the VIM, such as software images and virtualised compute resource flavours. This attribute is not intended for the modification of vimAssets entries passed earlier. Modification of VIM assets during the lifetime of a VNF instance is not necessary, since it is expected that all applicable assets have been on boarded into the VIM before the VNF is instantiated.	vimAssets

Name	Description	Schema
vimConnectio ns *optional*	Provides information regarding VIM connections that are approved to be used by the VNFM to allocate resources, and provides parameters of these VIM connections. The VNFM shall update the " vimConnectionInfo" attribute of the "VnfInstance" structure by adding unknown entries received in this attribute. This attribute is not intended for the modification of VimConnection entries passed earlier; for that, the VnfInfoModificationRequest structure shall be used. This attribute shall only be supported when VNF-related Resource Management in direct mode is applicable. In direct mode, this parameter shall be absent if the VIM information was configured to the VNFM in another way, present otherwise. This interface allows to signal the use of multiple VIMs per VNF. However, due to the partial support of this feature in the present release, it is recommended in the present document that the number of entries in the "vims" attribute in the Grant is not greater than 1.	< vimConnections > array
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfLcmOpOcc Id *required*	An identifier with the intention of being globally unique.	string
zoneGroups *optional*	Information about groups of resource zones that are related and that the NFVO has chosen to fulfil a zoneGroup constraint in the GrantVnfLifecycleOperation request. This information confirms that the NFVO has honoured the zoneGroup constraints that were passed as part of "placementConstraints" in the GrantRequest.	< zoneGroups > array
zones *optional*	Identifies resource zones where the resources are approved to be allocated by the VNFM.	< zones > array

_links

Name	Description	Schema
self *required*	This type represents a link to a resource.	self
vnfInstance *required*	This type represents a link to a resource.	vnfInstance

Name	Description	Schema
vnfLcmOpOcc *required*	This type represents a link to a resource.	[vnfLcmOpOcc](#)

self

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

vnfInstance

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

vnfLcmOpOcc

Name	Description	Schema
href *required*	URI of the referenced resource.	string (url)

addResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefinitionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGroupId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

extManagedVirtualLinks

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
virtualLinkDe scId *required*	An identifier that is unique within a VNF descriptor.	string

extVirtualLinks

Name	Description	Schema
extCps *required*	External CPs of the VNF to be connected to this external VL.	< extCps > array
extLinkPorts *optional*	Externally provided link ports to be used to connect external connection points to this external VL. If this attribute is not present, the VNFM shall create the link ports on the external VL.	< extLinkPorts > array

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string

extCps

Name	Description	Schema
cpConfig *optional*	List of instance data that need to be configured on the CP instances created from the respective CPD.	< cpConfig > array
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

cpConfig

Name	Description	Schema
cpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string

Name	Description	Schema
cpProtocolDat a *optional*	Parameters for configuring the network protocols on the link port that connects the CP to a VL. The following conditions apply to the attributes "linkPortId" and "cpProtocolData": * The "linkPortId" and "cpProtocolData" attributes shall both be absent for the deletion of an existing external CP instance addressed by cpInstanceId. * At least one of these attributes shall be present for a to-be-created external CP instance or an existing external CP instance. * If the "linkPortId" attribute is absent, the VNFM shall create a link port. * If the "cpProtocolData" attribute is absent, the "linkPortId" attribute shall be provided referencing a pre-created link port, and the VNFM can use means outside the scope of the present document to obtain the pre-configured address information for the connection point from the resource representing the link port. * If both "cpProtocolData" and "linkportId" are provided, the API consumer shall ensure that the cpProtocolData can be used with the pre-created link port referenced by "linkPortId".	< cpProtocolData > array
linkPortId *optional*	An identifier with the intention of being globally unique.	string

cpProtocolData

Name	Description	Schema
ipOverEthern et *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). This attribute allows to signal the addition of further types of layer and protocol in future versions of the present document in a backwards-compatible way. In the current version of the present document, only IP over Ethernet is supported.	enum (IP_OVER_ETHERNE T)

ipOverEthernet

Name	Description	Schema
ipAddresses *optional*	List of IP addresses to assign to the CP instance. Each entry represents IP address data for fixed or dynamic IP address assignment per subnet. If this attribute is not present, no IP address shall be assigned.	< ipAddresses > array

Name	Description	Schema
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range to be used, e.g. in case of egress connections. In case this attribute is present, IP addresses from the range will be used.	addressRange
fixedAddresses *optional*	Fixed addresses to assign (from the subnet defined by "subnetId" if provided). Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.	< string (IP) > array
numDynamic Addresses *optional*	Number of dynamic addresses to assign (from the subnet defined by "subnetId" if provided). Exactly one of "fixedAddresses", "numDynamicAddresses" or "ipAddressRange" shall be present.	integer
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
type *required*	The type of the IP addresses. Permitted values: IPV4, IPV6.	enum (IPV4, IPV6)

addressRange

Name	Description	Schema
maxAddress *required*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
minAddress *required*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)

extLinkPorts

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
resourceHand le *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance. Information about the resource is available from the VIM.	resourceHandle

resourceHandle

Name	Description	Schema
resourceId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *required*	An identifier with the intention of being globally unique.	string
vimLevelReso urceType *optional*	Type of the resource in the scope of the VIM or the resource provider.	string

removeResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefin itionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGrou pId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string

Name	Description	Schema
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

tempResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefin itionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGrou pId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

updateResources

Name	Description	Schema
reservationId *optional*	An identifier with the intention of being globally unique.	string
resourceDefinitionId *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string
resourceGroupId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimConnectionId *optional*	An identifier with the intention of being globally unique.	string
zoneId *optional*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

vimAssets

Name	Description	Schema
computeResourceFlavours *optional*	Mappings between virtual compute descriptors defined in the VNFD and compute resource flavours managed in the VIM.	< computeResourceFlavours > array
softwareImages *optional*	Mappings between software images defined in the VNFD and software images managed in the VIM.	< softwareImages > array

computeResourceFlavours

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
vimFlavourId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
vnfdVirtualCo mputeDescId *required*	An identifier that is unique within a VNF descriptor.	string

softwareImages

Name	Description	Schema
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
vimSoftwareI mageId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance.	string
vnfdSoftwareI mageId *required*	An identifier that is unique within a VNF descriptor.	string

vimConnections

Name	Description	Schema
accessInfo *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

Name	Description	Schema
extra *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object
id *required*	An identifier with the intention of being globally unique.	string
interfaceInfo *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object
vimId *optional*	An identifier with the intention of being globally unique.	string
vimType *required*	Discriminator for the different types of the VIM information. The value of this attribute determines the structure of the "interfaceInfo" and "accessInfo" attributes, based on the type of the VIM. The set of permitted values is expected to change over time as new types or versions of VIMs become available. The ETSI NFV registry of VIM-related information provides access to information about VimConnectionInfo definitions for various VIM types. The structure of the registry is defined in Annex C of SOL003.	string

zoneGroups

Name	Description	Schema
zoneId *required*	References of identifiers of "ZoneInfo" structures, each of which provides information about a resource zone that belongs to this group.	< string > array

zones

Name	Description	Schema
id *required*	An identifier that is unique within a limited local scope other than above listed identifiers, such as within a complex data structure or within a request-response pair. Representation: string of variable length.	string

Name	Description	Schema
resourceProvi derId *optional*	An identifier with the intention of being globally unique.	string
vimConnectio nId *optional*	An identifier with the intention of being globally unique.	string
zoneId *required*	An identifier with the intention of being globally unique.	string

Response 400

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 401

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 403

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 404

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 405

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 406

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 416

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 422

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 500

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 503

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)