SOL005 - NS Lifecycle Management Interface

Overview

SOL005 - NS Lifecycle Management Interface IMPORTANT: Please note that this file might be not aligned to the current version of the ETSI Group Specification it refers to and has not been approved by the ETSI NFV ISG. In case of discrepancies the published ETSI Group Specification takes precedence. Please report bugs to https://forge.etsi.org/bugzilla/buglist.cgi?component=Nfv-Openapis

Version information

Version : 1.2.0-impl:etsi.org:ETSI_NFV_OpenAPI:4

Contact information

Contact : NFV-SOL WG

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 : /nslcm/v1
Schemes : HTTP, HTTPS

Consumes

- `application/json`

Produces

- `application/json`

External Docs

Description : ETSI GS NFV-SOL 005 V2.6.1
URL : https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/005/02.06.01_60/gs_NFV-

Paths

POST /api_versions

Description

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response

Parameters

Type	Name	Description	Schema
Header	**Version** *optional*	Version of the API requested to use when responding to this request.	string

Responses

HTTP Code	Description	Schema
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405

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)

Name	Description	Schema
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)

Retrieve API version information

```
GET /api_versions
```

Description

The GET method reads API version information. This method shall follow the provisions specified in table 4.6.3.3.3.2-1 for request and response data structures, and response codes. URI query parameters are not supported.

Parameters

Type	Name	Description	Schema
Header	**Version** *optional*	Version of the API requested to use when responding to this request.	string

Responses

HTTP Code	Description	Schema
200	API version information was read successfully. The response body shall contain 4.4 API version information, as defined in clause 4.4.1.13. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Version` (string) : The used API version.	Response 200
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
413	413 PAYLOAD TOO LARGE If the payload body of a request is larger than the amount of data the API producer is willing or able to process, it shall respond with this response code, following the provisions in IETF RFC 7231 for the use of the "Retry-After" HTTP header and for closing the connection. The "ProblemDetails" structure may be omitted. **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. `Version` (string) : Version of the API used in the response.	Response 413
414	414 URI TOO LONG If the request URI of a request is longer than the API producer is willing or able to process, it shall respond with this response code. This condition can e.g. be caused by passing long queries in the request URI of a GET request. The "ProblemDetails" structure may be omitted. **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. `Version` (string) : Version of the API used in the response.	Response 414
416	416 RANGE NOT SATISFIABLE **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. `Version` (string) : Version of the API used in the response.	Response 416

HTTP Code	Description	Schema
422	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. 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. `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. `Version` (string) : Version of the API used in the response.	Response 422
429	429 TOO MANY REQUESTS If the API consumer has sent too many requests in a defined period of time and the API producer is able to detect that condition ("rate limiting"), the API producer shall respond with this response code, following the provisions in IETF RFC 6585 [17] for the use of the "Retry-After" HTTP header. The "ProblemDetails" structure shall be provided and shall include in the "detail" attribute more information about the source of the problem. The period of time and allowed number of requests are configured within the API producer by means outside the scope of the present document. **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. `Version` (string) : Version of the API used in the response.	Response 429

HTTP Code	Description	Schema
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
apiVersions *required*	Version(s) supported for the API signalled by the uriPrefix attribute.	< apiVersions > array
uriPrefix *required*	Specifies the URI prefix for the API, in the following form {apiRoot}/{apiName}/{apiMajorVersion}/.	string

apiVersions

Name	Description	Schema
isDeprecated *optional*	The Boolean is a data type having two values (TRUE and FALSE).	boolean
version *required*	Identifies a supported version. The value of the version attribute shall be a version identifier as specified in clause 4.6.1.	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

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 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

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 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

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 413

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 414

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 429

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 500

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 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

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 504

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)

PUT /api_versions

Description

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response

Parameters

Type	Name	Description	Schema
Header	**Version** *optional*	Version of the API requested to use when responding to this request.	string

Responses

HTTP Code	Description	Schema
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405

Response 405

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)

DELETE /api_versions

Description

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response

Parameters

Type	Name	Description	Schema
Header	**Version** *optional*	Version of the API requested to use when responding to this request.	string

Responses

HTTP Code	Description	Schema
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405

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)

PATCH /api_versions

Description

This method is not supported. When this method is requested on this resource, the API producer shall return a "405 Method Not Allowed" response

Parameters

Type	Name	Description	Schema
Header	**Version** *optional*	Version of the API requested to use when responding to this request.	string

Responses

HTTP Code	Description	Schema
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405

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)

Create a NS instance resource.

```
POST /ns_instances
```

Description

The POST method creates a new NS instance 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
Header	**Content-Type** *required*	The MIME type of the body of the request. Reference: IETF RFC 7231	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Body	**body** *required*	The NS creation parameters, as defined in clause 6.5.2.7.	body

body

Name	Description	Schema
nsDescription *required*	Human-readable description of the NS instance to be created.	string
nsName *required*	Human-readable name of the NS instance to be created.	string
nsdId *required*	An identifier with the intention of being globally unique.	string

Responses

HTTP Code	Description	Schema
201	201 Created A NS Instance identifier has been created successfully. The response body shall contain a representation of the created NS instance, as defined in clause 6.5.2.8. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NS instance. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response.	Response 201

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
416	416 RANGE NOT SATISFIABLE **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. `Version` (string) : Version of the API used in the response.	Response 416
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500

HTTP Code	Description	Schema
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 201

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
additionalAffinityOrAntiAffinityRule *optional*	Information on the additional affinity or anti-affinity rule from NS instantiation operation. Shall not conflict with rules already specified in the NSD.	< additionalAffinityOrAntiAffinityRule > array
flavourId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
id *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
monitoringParameter *optional*	Performance metrics tracked by the NFVO (e.g. for auto-scaling purposes) as identified by the NS designer in the NSD.	< monitoringParameter > array
nestedNsInstanceId *optional*	Identifier of the nested NS(s) of the NS instance.	< string > array
nsInstanceDescription *required*	Human readable description of the NS instance.	string
nsInstanceName *required*	Human readable name of the NS instance.	string
nsScaleStatus *optional*	Status of each NS scaling aspect declared in the applicable DF, how "big" the NS instance has been scaled w.r.t. that aspect. This attribute shall be present if the nsState attribute value is INSTANTIATED.	< nsScaleStatus > array
nsState *required*	The state of the NS instance. Permitted values: NOT_INSTANTIATED: The NS instance is terminated or not instantiated. INSTANTIATED: The NS instance is instantiated.	enum (NOT_INSTANTIATED, INSTANTIATED)
nsdId *required*	An identifier with the intention of being globally unique.	string
nsdInfoId *required*	An identifier with the intention of being globally unique.	string
pnfInfo *optional*	Information on the PNF(s) that are part of the NS instance.	< pnfInfo > array
sapInfo *optional*	Information on the SAP(s) of the NS instance.	< sapInfo > array
virtualLinkInfo *optional*	Information on the VL(s) of the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED and if the NS instance has specified connectivity.	< virtualLinkInfo > array

Name	Description	Schema
vnfInstance *optional*	Information on constituent VNF(s) of the NS instance.	< vnfInstance > array
vnffgInfo *optional*	Information on the VNFFG(s) of the NS instance.	< vnffgInfo > array

_links

Name	Description	Schema
heal *optional*	This type represents a link to a resource.	heal
instantiate *optional*	This type represents a link to a resource.	instantiate
nestedNsInstances *optional*	Links to resources related to this notification.	< nestedNsInstances > array
scale *optional*	This type represents a link to a resource.	scale
self *required*	This type represents a link to a resource.	self
terminate *optional*	This type represents a link to a resource.	terminate
update *optional*	This type represents a link to a resource.	update

heal

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

instantiate

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

nestedNsInstances

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

scale

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

terminate

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

update

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

additionalAffinityOrAntiAffinityRule

Name	Description	Schema
affinityOrAntiAffiinty *required*	The type of the constraint. Permitted values: AFFINITY ANTI_AFFINITY.	enum (AFFINITY, ANTI_AFFINITY)
scope *required*	Specifies the scope of the rule where the placement constraint applies. Permitted values: NFVI_POP ZONE ZONE_GROUP NFVI_NODE.	enum (NFVI_POP, ZONE, ZONE_GROUP, NFVI_NODE)
vnfInstanceId *optional*	Reference to the existing VNF instance as the subject of the affinity or anti-affinity rule. The existing VNF instance is not necessary as a part of the NS to be instantiated.	< string > array
vnfProfileId *optional*	Reference to a vnfProfile defined in the NSD. At least one VnfProfile which is used to instantiate VNF for the NS to be instantiated as the subject of the affinity or anti-affinity rule shall be present. When the VnfProfile which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VnfProfile presents is not necessary as a part of the NS to be instantiated.	< string > array
vnfdId *optional*	Reference to a VNFD. When the VNFD which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VNFD presents is not necessary as a part of the NS to be instantiated.	< string > array

monitoringParameter

Name	Description	Schema
id *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

Name	Description	Schema
name *optional*	Human readable name of the monitoring parameter, as defined in the NSD.	string
performanceMetric *required*	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027.	string

nsScaleStatus

Name	Description	Schema
nsScaleLevelId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsScalingAspectId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

pnfInfo

Name	Description	Schema
cpInfo *optional*	This type represents the information about the external CP of the PNF. It shall comply with the provisions defined in Table 6.5.3.17-1.	cpInfo
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *optional*	Name of the PNF.	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier with the intention of being globally unique.	string
pnfdInfoId *required*	An identifier with the intention of being globally unique.	string

cpInfo

Name	Description	Schema
cpInstanceId *required*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
cpProtocolData *optional*	Parameters for configuring the network protocols on the CP.	< cpProtocolData > array
cpdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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

Name	Description	Schema
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
numDynamicAddresses *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. Representation: string of variable length.	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)

sapInfo

Name	Description	Schema
description *optional*	Human readable description for the SAP instance.	string
id *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string

Name	Description	Schema
sapName *required*	Human readable name for the SAP instance.	string
sapProtocolInfo *required*	Network protocol information for this SAP.	< sapProtocolInfo > array
sapdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

sapProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean

Name	Description	Schema
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

virtualLinkInfo

Name	Description	Schema
id *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string

Name	Description	Schema
linkPort *optional*	Link ports of the VL instance. Cardinality of zero indicates that no port has yet been created for the VL instance.	< linkPort > array
nsVirtualLinkDescId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsVirtualLinkProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
resourceHandle *optional*	Identifier(s) of the virtualised network resource(s) realizing the VL instance. See note.	< resourceHandle > array

linkPort

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
nsCpHandle *optional*	This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.	nsCpHandle
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

nsCpHandle

Name	Description	Schema
nsInstanceId *optional*	An identifier with the intention of being globally unique.	string
nsSapInstanceId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string

Name	Description	Schema
pnfExtCpInstanceId *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
pnfInfoId *optional*	An identifier with the intention of being globally unique.	string
vnfExtCpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfInstanceId *optional*	An identifier with the intention of being globally unique.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfInstance

Name	Description	Schema
extensions *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
instantiatedVnfInfo *optional*	Information specific to an instantiated VNF instance. This attribute shall be present if the instantiateState attribute value is INSTANTIATED.	instantiatedVnfInfo
instantiationState *required*	The instantiation state of the VNF.	enum (NOT_INSTANTIATED, INSTANTIATED)
metadata *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

Name	Description	Schema
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	Human-readable description of the VNF instance. This attribute can be modified with the PATCH method.	string
vnfInstanceName *optional*	Name of the VNF instance. This attribute can be modified with the PATCH method.	string
vnfPkgId *required*	An identifier with the intention of being globally unique.	string
vnfProductName *required*	Name to identify the VNF Product. The value is copied from the VNFD.	string
vnfProvider *required*	Provider of the VNF and the VNFD. The value is copied from the VNFD.	string
vnfSoftwareVersion *required*	A Version. Representation: string of variable length.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string
vnfdVersion *required*	A Version. Representation: string of variable length.	string

instantiatedVnfInfo

Name	Description	Schema
extCpInfo *optional*	Information about the external CPs exposed by the VNF instance.	< extCpInfo > array
extManagedVirtualLinkInfo *optional*	External virtual links the VNF instance is connected to.	< extManagedVirtualLinkInfo > array

Name	Description	Schema
extVirtualLinkInfo *optional*	Information about the external VLs the VNF instance is connected to.	< extVirtualLinkInfo > array
flavourId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
localizationLanguage *optional*	Information about localization language of the VNF (includes e.g. strings in the VNFD). The localization languages supported by a VNF can be declared in the VNFD, and localization language selection can take place at instantiation time. The value shall comply with the format defined in IETF RFC 5646.	string
monitoringParameters *optional*	Performance metrics tracked by the VNFM (e.g. for auto-scaling purposes) as identified by the VNF provider in the VNFD.	< monitoringParameters > array
scaleStatus *optional*	Scale status of the VNF, one entry per aspect. Represents for every scaling aspect how "big" the VNF has been scaled w.r.t. that aspect.	< scaleStatus > array
virtualLinkResourceInfo *optional*	Information about the virtualised network resources used by the VLs of the VNF instance.	< virtualLinkResourceInfo > array
virtualStorageResourceInfo *optional*	Information on the virtualised storage resource(s) used as storage for the VNF instance.	< virtualStorageResourceInfo > array
vnfState *required*		enum (STARTED, STOPPED)
vnfcResourceInfo *optional*	Information about the virtualised compute and storage resources used by the VNFCs of the VNF instance.	< vnfcResourceInfo > array

extCpInfo

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

Name	Description	Schema
associatedVnfcCpId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpProtocolInfo *optional*	Network protocol information for this CP.	< cpProtocolInfo > array
cpdId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
extLinkPortId *optional*	This type describes the protocol layer(s) that a CP or SAP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in Table 6.5.3.58-1.	extLinkPortId
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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

cpProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange

Name	Description	Schema
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

extLinkPortId

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean

Name	Description	Schema
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

extManagedVirtualLinkInfo

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
networkResource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	networkResource
vnfLinkPorts *optional*	Link ports of this VL.	< vnfLinkPorts > array
vnfVirtualLinkDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

networkResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfLinkPorts

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
cpInstanceType *optional*	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: * VNFC_CP: The link port is connected to a VNFC CP * EXT_CP: The link port is associated to an external CP.	enum (VNFC_CP, EXT_CP)
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

extVirtualLinkInfo

Name	Description	Schema
extLinkPorts *optional*	Link ports of this VL.	< extLinkPorts > array
id *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

extLinkPorts

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
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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

monitoringParameters

Name	Description	Schema
id *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
name *optional*	Human readable name of the monitoring parameter, as defined in the VNFD.	string
performanceMetric *required*	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027.	string

scaleStatus

Name	Description	Schema
aspectId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
scaleLevel *required*	Indicates the scale level. The minimum value shall be 0 and the maximum value shall be <= maxScaleLevel as described in the VNFD.	integer

virtualLinkResourceInfo

Name	Description	Schema
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
networkResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	networkResource
reservationId *optional*	An identifier with the intention of being globally unique.	string
vnfLinkPorts *optional*	Links ports of this VL. Shall be present when the linkPort is used for external connectivity by the VNF (refer to VnfLinkPortInfo). May be present otherwise.	< vnfLinkPorts > array
vnfVirtualLinkDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

networkResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfLinkPorts

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
cpInstanceType *optional*	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: * VNFC_CP: The link port is connected to a VNFC CP * EXT_CP: The link port is associated to an external CP.	enum (VNFC_CP, EXT_CP)
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

virtualStorageResourceInfo

Name	Description	Schema
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
reservationId *optional*	An identifier with the intention of being globally unique.	string
storageResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	storageResource
virtualStorageDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

storageResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfcResourceInfo

Name	Description	Schema
computeResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	computeResource
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
reservationId *optional*	An identifier with the intention of being globally unique.	string
storageResourceIds *optional*	References to the VirtualStorage resources. The value refers to a VirtualStorageResourceInfo item in the VnfInstance.	< string > array
vduId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
vnfcCpInfo *optional*	CPs of the VNFC instance. Shall be present when that particular CP of the VNFC instance is associated to an external CP of the VNF instance. May be present otherwise.	< vnfcCpInfo > array

computeResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfcCpInfo

Name	Description	Schema
cpProtocolInfo *optional*	Network protocol information for this CP.	< cpProtocolInfo > array
cpdId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
vnfExtCpId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfLinkPortId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string

cpProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

vnffgInfo

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
nsCpHandle *optional*	This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.	nsCpHandle
nsVirtualLinkInfoId *optional*	Identifier(s) of the constituent VL instance(s) of this VNFFG instance.	< string > array
pnfdInfoId *optional*	Identifier(s) of the constituent PNF instance(s) of this VNFFG instance.	< string > array
vnfInstanceId *required*	Identifier(s) of the constituent VNF instance(s) of this VNFFG instance.	< string > array
vnffgdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

nsCpHandle

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

Name	Description	Schema
nsSapInstanceId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
pnfExtCpInstanceId *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
pnfInfoId *optional*	An identifier with the intention of being globally unique.	string
vnfExtCpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfInstanceId *optional*	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

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 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

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 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

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 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

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 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
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 409

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 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

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 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
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 504

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)

Query multiple NS instances.

```
GET /ns_instances
```

Description

Query NS Instances. The GET method queries information about multiple NS instances. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 6.4.2.3.2-1 and 6.4.2.3.2-2.

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

Type	Name	Description	Schema
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Query	**all_fields** *optional*	Include all complex attributes in the response. See clause 5.3 of ETSI GS NFV-SOL 013 for details. The NFVO shall support this parameter.	string
Query	**exclude_default** *optional*	"Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV-SOL 013 for details. The NFVO shall support this parameter. The following attributes shall be excluded from the NsInstance structure in the response body if this parameter is provided, or none of the parameters "all_fields," "fields", "exclude_fields", "exclude_default" are provided: - vnfInstances - pnfInfo - virtualLinkInfo - vnffgInfo - sapInfo - nsScaleStatus - additionalAffinityOrAntiAffinityRules"	string
Query	**exclude_fields** *optional*	"Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV-SOL 013 for details. The NFVO should support this parameter."	string
Query	**fields** *optional*	"Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV-SOL 013 for details. The NFVO should support this parameter."	string
Query	**filter** *optional*	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV-SOL 013. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the NsInstance and in data types referenced from it shall be supported by the NFVO in the filter expression.	string
Query	**nextpage_opaque_marker** *optional*	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV-SOL 013 for this resource.	string

Responses

HTTP Code	Description	Schema
200	200 OK Information about zero or more NS instances has been queried successfully. The response body shall contain in an array the representations of zero or more NS instances, as defined in clause 6.5.2.8. If the NFVO supports alternative 2 (paging) according to clause 4.7.2.1 for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV-SOL 013. **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. `Version` (string) : Version of the API used in the response. `Link` (string) : Reference to other resources. Used for paging in the present document, see clause 4.7.2.1.	< Response 200 > array

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
416	416 RANGE NOT SATISFIABLE **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. `Version` (string) : Version of the API used in the response.	Response 416
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500

HTTP Code	Description	Schema
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
NsInstance *optional*	This type represents a response for Query NS operation. It shall comply with the provisions defined in Table 6.5.2.10-1.	NsInstance

NsInstance

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
additionalAffinityOrAntiAffinityRule *optional*	Information on the additional affinity or anti-affinity rule from NS instantiation operation. Shall not conflict with rules already specified in the NSD.	< additionalAffinityOrAntiAffinityRule > array

Name	Description	Schema
flavourId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
id *required*	An identifier with the intention of being globally unique.	string
monitoringParameter *optional*	Performance metrics tracked by the NFVO (e.g. for auto-scaling purposes) as identified by the NS designer in the NSD.	< monitoringParameter > array
nestedNsInstanceId *optional*	Identifier of the nested NS(s) of the NS instance.	< string > array
nsInstanceDescription *required*	Human readable description of the NS instance.	string
nsInstanceName *required*	Human readable name of the NS instance.	string
nsScaleStatus *optional*	Status of each NS scaling aspect declared in the applicable DF, how "big" the NS instance has been scaled w.r.t. that aspect. This attribute shall be present if the nsState attribute value is INSTANTIATED.	< nsScaleStatus > array
nsState *required*	The state of the NS instance. Permitted values: NOT_INSTANTIATED: The NS instance is terminated or not instantiated. INSTANTIATED: The NS instance is instantiated.	enum (NOT_INSTANTIATED, INSTANTIATED)
nsdId *required*	An identifier with the intention of being globally unique.	string
nsdInfoId *required*	An identifier with the intention of being globally unique.	string
pnfInfo *optional*	Information on the PNF(s) that are part of the NS instance.	< pnfInfo > array
sapInfo *optional*	Information on the SAP(s) of the NS instance.	< sapInfo > array

Name	Description	Schema
virtualLinkInfo *optional*	Information on the VL(s) of the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED and if the NS instance has specified connectivity.	< virtualLinkInfo > array
vnfInstance *optional*	Information on constituent VNF(s) of the NS instance.	< vnfInstance > array
vnffgInfo *optional*	Information on the VNFFG(s) of the NS instance.	< vnffgInfo > array

_links

Name	Description	Schema
heal *optional*	This type represents a link to a resource.	heal
instantiate *optional*	This type represents a link to a resource.	instantiate
nestedNsInstances *optional*	Links to resources related to this notification.	< nestedNsInstances > array
scale *optional*	This type represents a link to a resource.	scale
self *required*	This type represents a link to a resource.	self
terminate *optional*	This type represents a link to a resource.	terminate
update *optional*	This type represents a link to a resource.	update

heal

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

instantiate

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

nestedNsInstances

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

scale

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

terminate

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

update

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

additionalAffinityOrAntiAffinityRule

Name	Description	Schema
affinityOrAntiAffiinty *required*	The type of the constraint. Permitted values: AFFINITY ANTI_AFFINITY.	enum (AFFINITY, ANTI_AFFINITY)
scope *required*	Specifies the scope of the rule where the placement constraint applies. Permitted values: NFVI_POP ZONE ZONE_GROUP NFVI_NODE.	enum (NFVI_POP, ZONE, ZONE_GROUP, NFVI_NODE)
vnfInstanceId *optional*	Reference to the existing VNF instance as the subject of the affinity or anti-affinity rule. The existing VNF instance is not necessary as a part of the NS to be instantiated.	< string > array
vnfProfileId *optional*	Reference to a vnfProfile defined in the NSD. At least one VnfProfile which is used to instantiate VNF for the NS to be instantiated as the subject of the affinity or anti-affinity rule shall be present. When the VnfProfile which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VnfProfile presents is not necessary as a part of the NS to be instantiated.	< string > array

Name	Description	Schema
vnfdId *optional*	Reference to a VNFD. When the VNFD which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VNFD presents is not necessary as a part of the NS to be instantiated.	< string > array

monitoringParameter

Name	Description	Schema
id *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
name *optional*	Human readable name of the monitoring parameter, as defined in the NSD.	string
performanceMetric *required*	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027.	string

nsScaleStatus

Name	Description	Schema
nsScaleLevelId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsScalingAspectId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

pnfInfo

Name	Description	Schema
cpInfo *optional*	This type represents the information about the external CP of the PNF. It shall comply with the provisions defined in Table 6.5.3.17-1.	cpInfo
pnfId *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
pnfName *optional*	Name of the PNF.	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier with the intention of being globally unique.	string
pnfdInfoId *required*	An identifier with the intention of being globally unique.	string

cpInfo

Name	Description	Schema
cpInstanceId *required*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
cpProtocolData *optional*	Parameters for configuring the network protocols on the CP.	< cpProtocolData > array
cpdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	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)

Name	Description	Schema
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)

sapInfo

Name	Description	Schema
description *optional*	Human readable description for the SAP instance.	string
id *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
sapName *required*	Human readable name for the SAP instance.	string
sapProtocolInfo *required*	Network protocol information for this SAP.	< sapProtocolInfo > array
sapdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

sapProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange

Name	Description	Schema
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

virtualLinkInfo

Name	Description	Schema
id *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
linkPort *optional*	Link ports of the VL instance. Cardinality of zero indicates that no port has yet been created for the VL instance.	< linkPort > array
nsVirtualLinkDescId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsVirtualLinkProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
resourceHandle *optional*	Identifier(s) of the virtualised network resource(s) realizing the VL instance. See note.	< resourceHandle > array

linkPort

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
nsCpHandle *optional*	This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.	nsCpHandle

Name	Description	Schema
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

nsCpHandle

Name	Description	Schema
nsInstanceId *optional*	An identifier with the intention of being globally unique.	string
nsSapInstanceId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
pnfExtCpInstanceId *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
pnfInfoId *optional*	An identifier with the intention of being globally unique.	string
vnfExtCpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfInstanceId *optional*	An identifier with the intention of being globally unique.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfInstance

Name	Description	Schema
extensions *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
instantiatedVnfInfo *optional*	Information specific to an instantiated VNF instance. This attribute shall be present if the instantiateState attribute value is INSTANTIATED.	instantiatedVnfInfo

Name	Description	Schema
instantiationState *required*	The instantiation state of the VNF.	enum (NOT_INSTANTIATED, INSTANTIATED)
metadata *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
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	Human-readable description of the VNF instance. This attribute can be modified with the PATCH method.	string
vnfInstanceName *optional*	Name of the VNF instance. This attribute can be modified with the PATCH method.	string
vnfPkgId *required*	An identifier with the intention of being globally unique.	string
vnfProductName *required*	Name to identify the VNF Product. The value is copied from the VNFD.	string
vnfProvider *required*	Provider of the VNF and the VNFD. The value is copied from the VNFD.	string
vnfSoftwareVersion *required*	A Version. Representation: string of variable length.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vnfdVersion *required*	A Version. Representation: string of variable length.	string

instantiatedVnfInfo

Name	Description	Schema
extCpInfo *optional*	Information about the external CPs exposed by the VNF instance.	< extCpInfo > array
extManagedVirtualLinkInfo *optional*	External virtual links the VNF instance is connected to.	< extManagedVirtualLinkInfo > array
extVirtualLinkInfo *optional*	Information about the external VLs the VNF instance is connected to.	< extVirtualLinkInfo > array
flavourId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
localizationLanguage *optional*	Information about localization language of the VNF (includes e.g. strings in the VNFD). The localization languages supported by a VNF can be declared in the VNFD, and localization language selection can take place at instantiation time. The value shall comply with the format defined in IETF RFC 5646.	string
monitoringParameters *optional*	Performance metrics tracked by the VNFM (e.g. for auto-scaling purposes) as identified by the VNF provider in the VNFD.	< monitoringParameters > array
scaleStatus *optional*	Scale status of the VNF, one entry per aspect. Represents for every scaling aspect how "big" the VNF has been scaled w.r.t. that aspect.	< scaleStatus > array
virtualLinkResourceInfo *optional*	Information about the virtualised network resources used by the VLs of the VNF instance.	< virtualLinkResourceInfo > array
virtualStorageResourceInfo *optional*	Information on the virtualised storage resource(s) used as storage for the VNF instance.	< virtualStorageResourceInfo > array

Name	Description	Schema
vnfState *required*		enum (STARTED, STOPPED)
vnfcResourceInfo *optional*	Information about the virtualised compute and storage resources used by the VNFCs of the VNF instance.	< vnfcResourceInfo > array

extCpInfo

Name	Description	Schema
associatedVnfVirtualLinkId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
associatedVnfcCpId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpProtocolInfo *optional*	Network protocol information for this CP.	< cpProtocolInfo > array
cpdId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
extLinkPortId *optional*	This type describes the protocol layer(s) that a CP or SAP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in Table 6.5.3.58-1.	extLinkPortId
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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

cpProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)

Name	Description	Schema
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string

Name	Description	Schema
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)

extLinkPortId

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array

Name	Description	Schema
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

extManagedVirtualLinkInfo

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
networkResource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	networkResource
vnfLinkPorts *optional*	Link ports of this VL.	< vnfLinkPorts > array
vnfVirtualLinkDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

networkResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfLinkPorts

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
cpInstanceType *optional*	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: * VNFC_CP: The link port is connected to a VNFC CP * EXT_CP: The link port is associated to an external CP.	enum (VNFC_CP, EXT_CP)
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

extVirtualLinkInfo

Name	Description	Schema
extLinkPorts *optional*	Link ports of this VL.	< extLinkPorts > array
id *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

extLinkPorts

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
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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

monitoringParameters

Name	Description	Schema
id *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
name *optional*	Human readable name of the monitoring parameter, as defined in the VNFD.	string
performanceMetric *required*	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027.	string

scaleStatus

Name	Description	Schema
aspectId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
scaleLevel *required*	Indicates the scale level. The minimum value shall be 0 and the maximum value shall be <= maxScaleLevel as described in the VNFD.	integer

virtualLinkResourceInfo

Name	Description	Schema
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
networkResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	networkResource
reservationId *optional*	An identifier with the intention of being globally unique.	string
vnfLinkPorts *optional*	Links ports of this VL. Shall be present when the linkPort is used for external connectivity by the VNF (refer to VnfLinkPortInfo). May be present otherwise.	< vnfLinkPorts > array
vnfVirtualLinkDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

networkResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfLinkPorts

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
cpInstanceType *optional*	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: * VNFC_CP: The link port is connected to a VNFC CP * EXT_CP: The link port is associated to an external CP.	enum (VNFC_CP, EXT_CP)
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

virtualStorageResourceInfo

Name	Description	Schema
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
reservationId *optional*	An identifier with the intention of being globally unique.	string
storageResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	storageResource
virtualStorageDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

storageResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfcResourceInfo

Name	Description	Schema
computeResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	computeResource
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
reservationId *optional*	An identifier with the intention of being globally unique.	string
storageResourceIds *optional*	References to the VirtualStorage resources. The value refers to a VirtualStorageResourceInfo item in the VnfInstance.	< string > array
vduId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
vnfcCpInfo *optional*	CPs of the VNFC instance. Shall be present when that particular CP of the VNFC instance is associated to an external CP of the VNF instance. May be present otherwise.	< vnfcCpInfo > array

computeResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfcCpInfo

Name	Description	Schema
cpProtocolInfo *optional*	Network protocol information for this CP.	< cpProtocolInfo > array
cpdId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
vnfExtCpId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfLinkPortId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string

cpProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

vnffgInfo

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
nsCpHandle *optional*	This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.	nsCpHandle
nsVirtualLinkInfoId *optional*	Identifier(s) of the constituent VL instance(s) of this VNFFG instance.	< string > array
pnfdInfoId *optional*	Identifier(s) of the constituent PNF instance(s) of this VNFFG instance.	< string > array
vnfInstanceId *required*	Identifier(s) of the constituent VNF instance(s) of this VNFFG instance.	< string > array
vnffgdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

nsCpHandle

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

Name	Description	Schema
nsSapInstanceId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
pnfExtCpInstanceId *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
pnfInfoId *optional*	An identifier with the intention of being globally unique.	string
vnfExtCpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfInstanceId *optional*	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

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 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

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 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

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 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

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 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
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 409

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 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

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 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
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 504

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)

Read an individual NS instance resource.

```
GET /ns_instances/{nsInstanceId}
```

Description

The GET method retrieves information about a NS instance by reading an individual NS instance 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

Type	Name	Description	Schema
Header	**Content-Type** *required*	The MIME type of the body of the request. Reference: IETF RFC 7231	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	Identifier of the NS instance.	string

Responses

HTTP Code	Description	Schema
200	200 OK Information about an individual NS instance has been queried successfully. The response body shall contain a representation of the NS instance. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response.	Response 200

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
416	416 RANGE NOT SATISFIABLE **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. `Version` (string) : Version of the API used in the response.	Response 416
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
additionalAffinityOrAntiAffinityRule *optional*	Information on the additional affinity or anti-affinity rule from NS instantiation operation. Shall not conflict with rules already specified in the NSD.	< additionalAffinityOrAntiAffinityRule > array
flavourId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
id *required*	An identifier with the intention of being globally unique.	string
monitoringParameter *optional*	Performance metrics tracked by the NFVO (e.g. for auto-scaling purposes) as identified by the NS designer in the NSD.	< monitoringParameter > array
nestedNsInstanceId *optional*	Identifier of the nested NS(s) of the NS instance.	< string > array
nsInstanceDescription *required*	Human readable description of the NS instance.	string

Name	Description	Schema
nsInstanceName *required*	Human readable name of the NS instance.	string
nsScaleStatus *optional*	Status of each NS scaling aspect declared in the applicable DF, how "big" the NS instance has been scaled w.r.t. that aspect. This attribute shall be present if the nsState attribute value is INSTANTIATED.	< nsScaleStatus > array
nsState *required*	The state of the NS instance. Permitted values: NOT_INSTANTIATED: The NS instance is terminated or not instantiated. INSTANTIATED: The NS instance is instantiated.	enum (NOT_INSTANTIATED, INSTANTIATED)
nsdId *required*	An identifier with the intention of being globally unique.	string
nsdInfoId *required*	An identifier with the intention of being globally unique.	string
pnfInfo *optional*	Information on the PNF(s) that are part of the NS instance.	< pnfInfo > array
sapInfo *optional*	Information on the SAP(s) of the NS instance.	< sapInfo > array
virtualLinkInfo *optional*	Information on the VL(s) of the NS instance. This attribute shall be present if the nsState attribute value is INSTANTIATED and if the NS instance has specified connectivity.	< virtualLinkInfo > array
vnfInstance *optional*	Information on constituent VNF(s) of the NS instance.	< vnfInstance > array
vnffgInfo *optional*	Information on the VNFFG(s) of the NS instance.	< vnffgInfo > array

_links

Name	Description	Schema
heal *optional*	This type represents a link to a resource.	heal

Name	Description	Schema
instantiate *optional*	This type represents a link to a resource.	instantiate
nestedNsInstances *optional*	Links to resources related to this notification.	< nestedNsInstances > array
scale *optional*	This type represents a link to a resource.	scale
self *required*	This type represents a link to a resource.	self
terminate *optional*	This type represents a link to a resource.	terminate
update *optional*	This type represents a link to a resource.	update

heal

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

instantiate

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

nestedNsInstances

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

scale

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

terminate

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

update

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

additionalAffinityOrAntiAffinityRule

Name	Description	Schema
affinityOrAntiAffiinty *required*	The type of the constraint. Permitted values: AFFINITY ANTI_AFFINITY.	enum (AFFINITY, ANTI_AFFINITY)
scope *required*	Specifies the scope of the rule where the placement constraint applies. Permitted values: NFVI_POP ZONE ZONE_GROUP NFVI_NODE.	enum (NFVI_POP, ZONE, ZONE_GROUP, NFVI_NODE)
vnfInstanceId *optional*	Reference to the existing VNF instance as the subject of the affinity or anti-affinity rule. The existing VNF instance is not necessary as a part of the NS to be instantiated.	< string > array
vnfProfileId *optional*	Reference to a vnfProfile defined in the NSD. At least one VnfProfile which is used to instantiate VNF for the NS to be instantiated as the subject of the affinity or anti-affinity rule shall be present. When the VnfProfile which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VnfProfile presents is not necessary as a part of the NS to be instantiated.	< string > array
vnfdId *optional*	Reference to a VNFD. When the VNFD which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VNFD presents is not necessary as a part of the NS to be instantiated.	< string > array

monitoringParameter

Name	Description	Schema
id *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
name *optional*	Human readable name of the monitoring parameter, as defined in the NSD.	string
performanceMetric *required*	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027.	string

nsScaleStatus

Name	Description	Schema
nsScaleLevelId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsScalingAspectId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

pnfInfo

Name	Description	Schema
cpInfo *optional*	This type represents the information about the external CP of the PNF. It shall comply with the provisions defined in Table 6.5.3.17-1.	cpInfo
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *optional*	Name of the PNF.	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier with the intention of being globally unique.	string
pnfdInfoId *required*	An identifier with the intention of being globally unique.	string

cpInfo

Name	Description	Schema
cpInstanceId *required*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
cpProtocolData *optional*	Parameters for configuring the network protocols on the CP.	< cpProtocolData > array
cpdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	string

Name	Description	Schema
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)

sapInfo

Name	Description	Schema
description *optional*	Human readable description for the SAP instance.	string
id *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
sapName *required*	Human readable name for the SAP instance.	string
sapProtocolInfo *required*	Network protocol information for this SAP.	< sapProtocolInfo > array
sapdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

sapProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)

Name	Description	Schema
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string

Name	Description	Schema
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)

virtualLinkInfo

Name	Description	Schema
id *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
linkPort *optional*	Link ports of the VL instance. Cardinality of zero indicates that no port has yet been created for the VL instance.	< linkPort > array
nsVirtualLinkDescId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsVirtualLinkProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
resourceHandle *optional*	Identifier(s) of the virtualised network resource(s) realizing the VL instance. See note.	< resourceHandle > array

linkPort

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
nsCpHandle *optional*	This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.	nsCpHandle
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

nsCpHandle

Name	Description	Schema
nsInstanceId *optional*	An identifier with the intention of being globally unique.	string
nsSapInstanceId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
pnfExtCpInstanceId *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
pnfInfoId *optional*	An identifier with the intention of being globally unique.	string
vnfExtCpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfInstanceId *optional*	An identifier with the intention of being globally unique.	string

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. Representation: string of variable length.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfInstance

Name	Description	Schema
extensions *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
id *required*	An identifier with the intention of being globally unique.	string
instantiatedVnfInfo *optional*	Information specific to an instantiated VNF instance. This attribute shall be present if the instantiateState attribute value is INSTANTIATED.	instantiatedVnfInfo
instantiationState *required*	The instantiation state of the VNF.	enum (NOT_INSTANTIATED, INSTANTIATED)
metadata *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
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	Human-readable description of the VNF instance. This attribute can be modified with the PATCH method.	string
vnfInstanceName *optional*	Name of the VNF instance. This attribute can be modified with the PATCH method.	string
vnfPkgId *required*	An identifier with the intention of being globally unique.	string
vnfProductName *required*	Name to identify the VNF Product. The value is copied from the VNFD.	string
vnfProvider *required*	Provider of the VNF and the VNFD. The value is copied from the VNFD.	string

Name	Description	Schema
vnfSoftwareVersion *required*	A Version. Representation: string of variable length.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string
vnfdVersion *required*	A Version. Representation: string of variable length.	string

instantiatedVnfInfo

Name	Description	Schema
extCpInfo *optional*	Information about the external CPs exposed by the VNF instance.	< extCpInfo > array
extManagedVirtualLinkInfo *optional*	External virtual links the VNF instance is connected to.	< extManagedVirtualLinkInfo > array
extVirtualLinkInfo *optional*	Information about the external VLs the VNF instance is connected to.	< extVirtualLinkInfo > array
flavourId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
localizationLanguage *optional*	Information about localization language of the VNF (includes e.g. strings in the VNFD). The localization languages supported by a VNF can be declared in the VNFD, and localization language selection can take place at instantiation time. The value shall comply with the format defined in IETF RFC 5646.	string
monitoringParameters *optional*	Performance metrics tracked by the VNFM (e.g. for auto-scaling purposes) as identified by the VNF provider in the VNFD.	< monitoringParameters > array
scaleStatus *optional*	Scale status of the VNF, one entry per aspect. Represents for every scaling aspect how "big" the VNF has been scaled w.r.t. that aspect.	< scaleStatus > array

Name	Description	Schema
virtualLinkResourceInfo *optional*	Information about the virtualised network resources used by the VLs of the VNF instance.	< virtualLinkResourceInfo > array
virtualStorageResourceInfo *optional*	Information on the virtualised storage resource(s) used as storage for the VNF instance.	< virtualStorageResourceInfo > array
vnfState *required*		enum (STARTED, STOPPED)
vnfcResourceInfo *optional*	Information about the virtualised compute and storage resources used by the VNFCs of the VNF instance.	< vnfcResourceInfo > array

extCpInfo

Name	Description	Schema
associatedVnfVirtualLinkId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
associatedVnfcCpId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpProtocolInfo *optional*	Network protocol information for this CP.	< cpProtocolInfo > array
cpdId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
extLinkPortId *optional*	This type describes the protocol layer(s) that a CP or SAP uses together with protocol-related information, like addresses. It shall comply with the provisions defined in Table 6.5.3.58-1.	extLinkPortId
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string

Name	Description	Schema
metadata *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

cpProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

Name	Description	Schema
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange

Name	Description	Schema
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

extLinkPortId

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

extManagedVirtualLinkInfo

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
networkResource *optional*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	networkResource
vnfLinkPorts *optional*	Link ports of this VL.	< vnfLinkPorts > array
vnfVirtualLinkDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

networkResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfLinkPorts

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
cpInstanceType *optional*	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: * VNFC_CP: The link port is connected to a VNFC CP * EXT_CP: The link port is associated to an external CP.	enum (VNFC_CP, EXT_CP)
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vimLevelResourceType *optional*	Type of the resource in the scope of the VIM or the resource provider. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

extVirtualLinkInfo

Name	Description	Schema
extLinkPorts *optional*	Link ports of this VL.	< extLinkPorts > array
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 or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

extLinkPorts

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
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 or by an NS 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. Representation: string of variable length.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

monitoringParameters

Name	Description	Schema
id *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
name *optional*	Human readable name of the monitoring parameter, as defined in the VNFD.	string

Name	Description	Schema
performanceMetric *required*	Performance metric that is monitored. This attribute shall contain the related "Measurement Name" value as defined in clause 7.2 of ETSI GS NFV-IFA 027.	string

scaleStatus

Name	Description	Schema
aspectId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
scaleLevel *required*	Indicates the scale level. The minimum value shall be 0 and the maximum value shall be <= maxScaleLevel as described in the VNFD.	integer

virtualLinkResourceInfo

Name	Description	Schema
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
networkResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	networkResource
reservationId *optional*	An identifier with the intention of being globally unique.	string
vnfLinkPorts *optional*	Links ports of this VL. Shall be present when the linkPort is used for external connectivity by the VNF (refer to VnfLinkPortInfo). May be present otherwise.	< vnfLinkPorts > array
vnfVirtualLinkDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

networkResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfLinkPorts

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
cpInstanceType *optional*	Type of the CP instance that is identified by cpInstanceId. Shall be present if "cpInstanceId" is present, and shall be absent otherwise. Permitted values: * VNFC_CP: The link port is connected to a VNFC CP * EXT_CP: The link port is associated to an external CP.	enum (VNFC_CP, EXT_CP)
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
resourceHandle *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

virtualStorageResourceInfo

Name	Description	Schema
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
reservationId *optional*	An identifier with the intention of being globally unique.	string
storageResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	storageResource
virtualStorageDescId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

storageResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfcResourceInfo

Name	Description	Schema
computeResource *required*	This type represents the information that allows addressing a virtualised resource that is used by a VNF instance or by an NS instance. Information about the resource is available from the VIM.	computeResource
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
reservationId *optional*	An identifier with the intention of being globally unique.	string
storageResourceIds *optional*	References to the VirtualStorage resources. The value refers to a VirtualStorageResourceInfo item in the VnfInstance.	< string > array
vduId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

Name	Description	Schema
vnfcCpInfo *optional*	CPs of the VNFC instance. Shall be present when that particular CP of the VNFC instance is associated to an external CP of the VNF instance. May be present otherwise.	< vnfcCpInfo > array

computeResource

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

vnfcCpInfo

Name	Description	Schema
cpProtocolInfo *optional*	Network protocol information for this CP.	< cpProtocolInfo > array
cpdId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
id *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
metadata *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
vnfExtCpId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfLinkPortId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string

cpProtocolInfo

Name	Description	Schema
ipOverEthernet *required*	This type represents information about a network address that has been assigned. It shall comply with the provisions defined in Table 6.5.3.18-1.	ipOverEthernet
layerProtocol *required*	The identifier of layer(s) and protocol(s) associated to the network address information. Permitted values: IP_OVER_ETHERNET See note.	enum (IP_OVER_ETHERNET)

ipOverEthernet

Name	Description	Schema
addressRange *required*	An IP address range used, e.g. in case of egress connections. See note.	addressRange
addresses *required*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided). See note.	< string (IP) > array
ipAddresses *required*	Addresses assigned to the CP instance. Each entry represents IP addresses assigned by fixed or dynamic IP address assignment per subnet.	< ipAddresses > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
macAddress *required*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

Name	Description	Schema
maxAddress *optional*	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 *optional*	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)
subnetId *required*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	string
type *optional*	The type of the IP addresses	enum (PV4, PV6)

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)

ipAddresses

Name	Description	Schema
addressRange *optional*	An IP address range used, e.g., in case of egress connections. Exactly one of "addresses" or "addressRange" shall be present.	addressRange

Name	Description	Schema
addresses *optional*	Fixed addresses assigned (from the subnet defined by "subnetId" if provided).	< string (IP) > array
isDynamic *optional*	Indicates whether this set of addresses was assigned dynamically (true) or based on address information provided as input from the API consumer (false). Shall be present if "addresses" is present and shall be absent otherwise.	boolean
subnetId *optional*	An identifier maintained by the VIM or other resource provider. It is expected to be unique within the VIM instance. Representation: string of variable length.	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)

vnffgInfo

Name	Description	Schema
id *required*	An identifier with the intention of being globally unique.	string
nsCpHandle *optional*	This type represents an identifier of the CP or SAP instance. It shall comply with the provisions defined in Table 6.5.3.56-1.	nsCpHandle

Name	Description	Schema
nsVirtualLinkInfoId *optional*	Identifier(s) of the constituent VL instance(s) of this VNFFG instance.	< string > array
pnfdInfoId *optional*	Identifier(s) of the constituent PNF instance(s) of this VNFFG instance.	< string > array
vnfInstanceId *required*	Identifier(s) of the constituent VNF instance(s) of this VNFFG instance.	< string > array
vnffgdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

nsCpHandle

Name	Description	Schema
nsInstanceId *optional*	An identifier with the intention of being globally unique.	string
nsSapInstanceId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
pnfExtCpInstanceId *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
pnfInfoId *optional*	An identifier with the intention of being globally unique.	string
vnfExtCpInstanceId *optional*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfInstanceId *optional*	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

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 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

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 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

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 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
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

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 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

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 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

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)

Response 504

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)

Delete NS instance resource.

```
DELETE /ns_instances/{nsInstanceId}
```

Description

Delete NS Identifier This method deletes an individual NS instance resource.

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235.	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	Identifier of the NS instance.	string

Responses

HTTP Code	Description	Schema
204	204 No Content The NS instance resource and the associated NS identifier were deleted successfully. 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
412	412 PRECONDITION FAILED Error: A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error. **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. `Version` (string) : Version of the API used in the response.	Response 412
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500

HTTP Code	Description	Schema
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 412

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)

Response 504

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)

Heal a NS instance.

```
POST /ns_instances/{nsInstanceId}/heal
```

Description

The POST method requests to heal a NS instance resource. This method shall follow the provisions specified in the Tables 6.4.7.3.1-1 and 6.4.7.3.1-2 for URI query parameters, request and response data structures, and response codes.

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
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	Identifier of the NS instance to be healed.	string
Body	**body** *required*	Parameters for the heal NS operation, as defined in clause 6.5.2.12.	body

body

Name	Description	Schema
healNsData *optional*	This type represents the information used to heal a NS. It shall comply with the provisions defined in Table 6.5.3.43-1.	healNsData
healVnfData *optional*	Additional parameters passed by the NFVO as input to the healing process, specific to the VNF being healed, as declared in the VNFD as part of "HealVnfOpConfig".	< healVnfData > array

healNsData

Name	Description	Schema
actionsHealing *optional*	Used to specify dedicated healing actions in a particular order (e.g. as a script). The actionsHealing attribute can be used to provide a specific script whose content and actions might only be possible to be derived during runtime.	< string > array
additionalParamsforNs *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
degreeHealing *required*	Indicates the degree of healing. Possible values include: - HEAL_RESTORE: Complete the healing of the NS restoring the state of the NS before the failure occurred - HEAL_QOS: Complete the healing of the NS based on the newest QoS values - HEAL_RESET: Complete the healing of the NS resetting to the original instantiation state of the NS - PARTIAL_HEALING	enum (HEAL_RESTORE, HEAL_QOS, HEAL_RESET, PARTIAL_HEALING)
healScript *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

healVnfData

Name	Description	Schema
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
cause *optional*	Indicates the reason why a healing procedure is required.	string
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Instantiate a NS.

```
POST /ns_instances/{nsInstanceId}/instantiate
```

Description

The POST method requests to instantiate a NS instance 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

Type	Name	Description	Schema
Header	**Content-Type** *required*	The MIME type of the body of the request. Reference: IETF RFC 7231	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	Identifier of the NS instance to be instantiated.	string
Body	**body** *required*	Parameters for the instantiate NS operation, as defined in clause 6.5.2.10.	body

body

Name	Description	Schema
additionalAffinityOrAntiAffinityRule *optional*	Specifies additional affinity or anti-affinity constraint for the VNF instances to be instantiated as part of the NS instantiation. Shall not conflict with rules already specified in the NSD.	< additionalAffinityOrAntiAffinityRule > array
additionalParamForNestedNs *optional*	Allows the OSS/BSS to provide additional parameter(s) per nested NS instance (as opposed to the composite NS level, which is covered in additionalParamForNs, and as opposed to the VNF level, which is covered in additionalParamForVnf). This is for nested NS instances that are to be created by the NFVO as part of the NS instantiation and not for existing nested NS instances that are referenced for reuse.	< additionalParamForNestedNs > array
additionalParamsForNs *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
additionalParamsForVnf *optional*	Allows the OSS/BSS to provide additional parameter(s) per VNF instance (as opposed to the composite NS level, which is covered in additionalParamsForNs and as opposed to the nested NS level, which is covered in additionalParamForNestedNs). This is for VNFs that are to be created by the NFVO as part of the NS instantiation and not for existing VNF that are referenced for reuse.	< additionalParamsForVnf > array

Name	Description	Schema
addpnfData *optional*	Information on the PNF(s) that are part of this NS.	< addpnfData > array
locationConstraints *optional*	Defines the location constraints for the VNF to be instantiated as part of the NS instantiation. An example can be a constraint for the VNF to be in a specific geographic location..	< locationConstraints > array
nestedNsInstanceData *optional*	Specify an existing NS instance to be used as a nested NS within the NS. If needed, the NS Profile to be used for this nested NS instance is also provided. NOTE 2: The NS DF of each nested NS shall be one of the allowed flavours in the associated NSD (as referenced in the nestedNsd attribute of the NSD of the NS to be instantiated). NOTE 3: The NSD of each referenced NSs (i.e. each nestedInstanceId) shall match the one of the nested NSD in the composite NSD.	< nestedNsInstanceData > array
nsFlavourId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsInstantiationLevelId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
sapData *optional*	Create data concerning the SAPs of this NS.	< sapData > array
vnfInstanceData *optional*	Specify an existing VNF instance to be used in the NS. If needed, the VNF Profile to be used for this VNF instance is also provided. The DF of the VNF instance shall match the VNF DF present in the associated VNF Profile.	< vnfInstanceData > array

additionalAffinityOrAntiAffinityRule

Name	Description	Schema
affinityOrAntiAffiinty *required*	The type of the constraint. Permitted values: AFFINITY ANTI_AFFINITY.	enum (AFFINITY, ANTI_AFFINITY)
scope *required*	Specifies the scope of the rule where the placement constraint applies. Permitted values: NFVI_POP ZONE ZONE_GROUP NFVI_NODE.	enum (NFVI_POP, ZONE, ZONE_GROUP, NFVI_NODE)

Name	Description	Schema
vnfInstanceId *optional*	Reference to the existing VNF instance as the subject of the affinity or anti-affinity rule. The existing VNF instance is not necessary as a part of the NS to be instantiated.	< string > array
vnfProfileId *optional*	Reference to a vnfProfile defined in the NSD. At least one VnfProfile which is used to instantiate VNF for the NS to be instantiated as the subject of the affinity or anti-affinity rule shall be present. When the VnfProfile which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VnfProfile presents is not necessary as a part of the NS to be instantiated.	< string > array
vnfdId *optional*	Reference to a VNFD. When the VNFD which is not used to instantiate VNF, it presents all VNF instances of this type as the subjects of the affinity or anti-affinity rule. The VNF instance which the VNFD presents is not necessary as a part of the NS to be instantiated.	< string > array

additionalParamForNestedNs

Name	Description	Schema
additionalParam *optional*	Additional parameters that are to be applied on a per nested NS instance.	< object > array
nsProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

additionalParamsForVnf

Name	Description	Schema
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
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

addpnfData

Name	Description	Schema
cpData *optional*	Address assigned for the PNF external CP(s).	< cpData > array
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *required*	Name of the PNF	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier with the intention of being globally unique.	string

cpData

Name	Description	Schema
cpInstanceI16 *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
cpProtocolData *required*	Address assigned for this CP.	< cpProtocolData > array
cpdId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	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)

Name	Description	Schema
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)

locationConstraints

Name	Description	Schema
locationConstraints *optional*	This type represents location constraints for a VNF to be instantiated. The location constraints shall be presented as a country code, optionally followed by a civic address based on the format defined by IETF RFC 4776 [13].	locationConstraints
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

locationConstraints

Name	Description	Schema
civicAddressElement *optional*	Zero or more elements comprising the civic address.	< civicAddressElement > array
countryCode *required*	The two-letter ISO 3166 [29] country code in capital letters.	string

civicAddressElement

Name	Description	Schema
caType *required*	Describe the content type of caValue. The value of caType shall comply with Section 3.4 of IETF RFC 4776 [13].	integer
caValue *required*	Content of civic address element corresponding to the caType. The format caValue shall comply with Section 3.4 of IETF RFC 4776 [13].	string

nestedNsInstanceData

Name	Description	Schema
nestedNsInstanceId *required*	An identifier with the intention of being globally unique.	string
nsProfileId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string

sapData

Name	Description	Schema
description *required*	Human readable description for the SAP.	string
sapName *required*	Human readable name for the SAP.	string
sapProtocolData *optional*	Parameters for configuring the network protocols on the SAP.	< sapProtocolData > array
sapdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

sapProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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

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
numDynamicAddresses *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. Representation: string of variable length.	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)

vnfInstanceData

Name	Description	Schema
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
416	416 RANGE NOT SATISFIABLE **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. `Version` (string) : Version of the API used in the response.	Response 416
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500

HTTP Code	Description	Schema
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 416

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)

Response 504

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)

Scale a NS instance.

```
POST /ns_instances/{nsInstanceId}/scale
```

Description

The POST method requests to scale a NS instance 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
Header	**Content-Type** *required*	The MIME type of the body of the request. Reference: IETF RFC 7231	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	Identifier of the NS instance to be scaled.	string
Body	**body** *required*	Parameters for the scale NS operation, as defined in clause 6.5.2.13.	body

body

Name	Description	Schema
scaleNsData *optional*	This type represents the information to scale a NS.	scaleNsData
scaleType *required*	Indicates the type of scaling to be performed. Possible values: - SCALE_NS - SCALE_VNF	enum (SCALE_NS, SCALE_VNF)
scaleVnfData *optional*	The necessary information to scale the referenced NS instance. It shall be present when scaleType = SCALE_VNF.	< scaleVnfData > array

scaleNsData

Name	Description	Schema
additionalParamsForNs *optional*	This type defines the additional parameters for the VNF instance to be created associated with an NS instance. It shall comply with the provisions defined in Table 6.5.3.22-1.	additionalParamsForNs

Name	Description	Schema
additionalParamsForVnf *optional*	Allows the OSS/BSS to provide additional parameter(s) per VNF instance (as opposed to the NS level, which is covered in additionalParamforNs). This is for VNFs that are to be created by the NFVO as part of the NS scaling and not for existing VNF that are covered by the scaleVnfData.	< additionalParamsForVnf > array
locationConstraints *optional*	The location constraints for the VNF to be instantiated as part of the NS scaling. An example can be a constraint for the VNF to be in a specific geographic location.	< locationConstraints > array
scaleNsByStepsData *optional*	This type represents the information used to scale an NS instance by one or more scaling steps, with respect to a particular NS scaling aspect. Performing a scaling step means increasing/decreasing the capacity of an NS instance in a discrete manner, i.e. moving from one NS scale level to another. The NS scaling aspects and their corresponding NS scale levels applicable to the NS instance are declared in the NSD.	scaleNsByStepsData
scaleNsToLevelData *optional*	This type represents the information used to scale an NS instance to a target size. The target size is either expressed as an NS instantiation level or as a list of NS scale levels, one per NS scaling aspect, of the current DF. The NS instantiation levels, the NS scaling aspects and their corresponding NS scale levels applicable to the NS instance are declared in the NSD.	scaleNsToLevelData
vnfInstanceToBeAdded *optional*	An existing VNF instance to be added to the NS instance as part of the scaling operation. If needed, the VNF Profile to be used for this VNF instance may also be provided.	< vnfInstanceToBeAdded > array
vnfInstanceToBeRemoved *optional*	The VNF instance to be removed from the NS instance as part of the scaling operation.	< string > array

additionalParamsForNs

Name	Description	Schema
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
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

additionalParamsForVnf

Name	Description	Schema
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
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

locationConstraints

Name	Description	Schema
locationConstraints *optional*	This type represents location constraints for a VNF to be instantiated. The location constraints shall be presented as a country code, optionally followed by a civic address based on the format defined by IETF RFC 4776 [13].	locationConstraints
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

locationConstraints

Name	Description	Schema
civicAddressElement *optional*	Zero or more elements comprising the civic address.	< civicAddressElement > array
countryCode *required*	The two-letter ISO 3166 [29] country code in capital letters.	string

civicAddressElement

Name	Description	Schema
caType *required*	Describe the content type of caValue. The value of caType shall comply with Section 3.4 of IETF RFC 4776 [13].	integer

Name	Description	Schema
caValue *required*	Content of civic address element corresponding to the caType. The format caValue shall comply with Section 3.4 of IETF RFC 4776 [13].	string

scaleNsByStepsData

Name	Description	Schema
aspectId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
numberOfSteps *optional*	The number of scaling steps to be performed. Defaults to 1.	integer
scalingDirection *required*	The scaling direction. Possible values are: - SCALE_IN - SCALE_OUT.	enum (SCALE_IN, SCALE_OUT)

scaleNsToLevelData

Name	Description	Schema
nsInstantiationLevel *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsScaleInfo *optional*	For each NS scaling aspect of the current DF, defines the target NS scale level to which the NS instance is to be scaled.	< nsScaleInfo > array

nsScaleInfo

Name	Description	Schema
nsScaleLevelId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsScalingAspectId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

vnfInstanceToBeAdded

Name	Description	Schema
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

scaleVnfData

Name	Description	Schema
scaleByStepData *optional*	This type describes the information to scale a VNF instance by steps. The NFVO shall then invoke the Scale VNF operation towards the appropriate VNFM.	scaleByStepData
scaleToLevelData *optional*	This type describes the information used to scale a VNF instance to a target size. The target size is either expressed as an instantiation level of that DF as defined in the VNFD, or given as a list of scale levels, one per scaling aspect of that DF. Instantiation levels and scaling aspects are declared in the VNFD. The NFVO shall then invoke the ScaleVnfToLevel operation towards the appropriate VNFM..	scaleToLevelData
scaleVnfType *required*	Type of the scale VNF operation requested. Allowed values are: - SCALE_OUT - SCALE_IN - SCALE_TO_INSTANTIATION_LEVEL - SCALE_TO_SCALE_LEVEL(S) The set of types actually supported depends on the capabilities of the VNF being managed.	enum (SCALE_OUT, SCALE_IN, SCALE_TO_INSTANTIATION_LEVEL, SCALE_TO_SCALE_LEVEL(S))
vnfInstanceid *required*	An identifier with the intention of being globally unique.	string

scaleByStepData

Name	Description	Schema
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
aspectId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

Name	Description	Schema
numberOfSteps *optional*	Number of scaling steps. It shall be a positive number. Defaults to 1. The VNF provider defines in the VNFD whether or not a particular VNF supports performing more than one step at a time. Such a property in the VNFD applies for all instances of a particular VNF.	integer

scaleToLevelData

Name	Description	Schema
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
vnfInstantiationLevelId *optional*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
vnfScaleInfo *optional*	For each scaling aspect of the current deployment flavor, indicates the target scale level to which the VNF is to be scaled.	< vnfScaleInfo > array

vnfScaleInfo

Name	Description	Schema
aspectId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
scaleLevel *required*	Indicates the scale level. The minimum value shall be 0 and the maximum value shall be <= maxScaleLevel as described in the VNFD.	integer

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Terminate a NS instance.

```
POST /ns_instances/{nsInstanceId}/terminate
```

Description

Terminate NS task. The POST method terminates a NS instance. This method can only be used with a NS instance in the INSTANTIATED state. Terminating a NS instance does not delete the NS instance identifier, but rather transitions the NS into the NOT_INSTANTIATED state. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 6.4.8.3.1-1 and 6.8.8.3.1-2.

Parameters

Type	Name	Description	Schema
Header	**Accept** *required*	Content-Types that are acceptable for the response. Reference: IETF RFC 7231	string

Type	Name	Description	Schema
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
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	The identifier of the NS instance to be terminated.	string
Body	**body** *required*	The terminate NS request parameters, as defined in clause 6.5.2.14.	object

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Updates a NS instance.

```
POST /ns_instances/{nsInstanceId}/update
```

Description

Scale NS instance. The POST method requests to scale a NS instance 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

Type	Name	Description	Schema
Header	**Content-Type** *required*	The MIME type of the body of the request. Reference: IETF RFC 7231	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsInstanceId** *required*	Identifier of the NS instance to be updated.	string
Body	**body** *required*	Parameters for the update NS operation, as defined in clause 6.5.2.11.	body

body

Name	Description	Schema
addNestedNsData *optional*	The identifier of an existing nested NS instance to be added to (nested within) the NS instance. It shall be present only if updateType = "ADD_NESTED_NS".	< addNestedNsData > array
addPnfData *optional*	specifies the PNF to be added into the NS instance. It shall be present only if updateType = "ADD_PNF".	< addPnfData > array
addSap *optional*	Identifies a new SAP to be added to the NS instance. It shall be present only if updateType = "ADD_SAP."	< addSap > array
addVnfIstance *optional*	Identifies an existing VNF instance to be added to the NS instance. It shall be present only if updateType = "ADD_VNF".	< addVnfIstance > array
addVnffg *optional*	Specify the new VNFFG to be created to the NS Instance. It shall be present only if updateType = "ADD_VNFFG".	< addVnffg > array

Name	Description	Schema
assocNewNsdVersionData *optional*	This type specifies a new NSD version that is associated to the NS instance. After issuing the Update NS operation with updateType = "AssocNewNsdVersion", the NFVO shall use the referred NSD as a basis for the given NS instance. Different versions of the same NSD have same nsdInvariantId, but different nsdId attributes, therefore if the nsdInvariantId of the NSD version that is to be associated to this NS instance is different from the one used before, the NFVO shall reject the request. Only new versions of the same NSD can be associated to an existing NS instance. This data type shall comply with the provisions defined in Table 6.5.3.34-1.	assocNewNsdVersionData
changeExtVnfConnectivityData *optional*	Specifies the new external connectivity data of the VNF instance to be changed. It shall be present only if updateType = "CHANGE_EXTERNAL_VNF_CONNECTIVITY".	< changeExtVnfConnectivityData > array
changeNsFlavourData *optional*	This type specifies an existing NS instance for which the DF needs to be changed. This specifies the new DF, the instantiationLevel of the new DF that may be used and the additional parameters as input for the flavour change. It shall comply with the provisions defined in Table 6.5.3.39-1.	changeNsFlavourData
changeVnfFlavourData *optional*	Identifies the new DF of the VNF instance to be changed to. It shall be present only if updateType = "CHANGE_VNF_DF".	< changeVnfFlavourData > array
instantiateVnfData *optional*	Identifies the new VNF to be instantiated. It can be used e.g. for the bottom-up NS creation. It shall be present only if updateType = "INSTANTIATE_VNF".	< instantiateVnfData > array
modifyPnfData *optional*	Specifies the PNF to be modified in the NS instance. It shall be present only if updateType = "MODIFY_PNF".	< modifyPnfData > array
modifyVnfInfoData *optional*	Identifies the VNF information parameters and/or the configurable properties of VNF instance to be modified. It shall be present only if updateType = "MODIFY_VNF_INFORMATION".	< modifyVnfInfoData > array
moveVnfInstanceData *optional*	Specify existing VNF instance to be moved from one NS instance to another NS instance. It shall be present only if updateType = MOVE_VNF".	< moveVnfInstanceData > array

Name	Description	Schema
operateVnfData *optional*	Identifies the state of the VNF instance to be changed. It shall be present only if updateType = "OPERATE_VNF".	< operateVnfData > array
removeNestedNsId *optional*	The identifier of an existing nested NS instance to be removed from the NS instance. It shall be present only if updateType = "REMOVE_NESTED_NS".	< string > array
removePnfId *optional*	Identifier of the PNF to be deleted from the NS instance. It shall be present only if updateType = "REMOVE_PNF".	< string > array
removeSapId *optional*	The identifier an existing SAP to be removed from the NS instance. It shall be present only if updateType = "REMOVE_SAP."	< string > array
removeVnfInstanceId *optional*	Identifies an existing VNF instance to be removed from the NS instance. It contains the identifier(s) of the VNF instances to be removed. It shall be present only if updateType = "REMOVE_VNF." Note: If a VNF instance is removed from a NS and this NS was the last one for which this VNF instance was a part, the VNF instance is terminated by the NFVO.	< string > array
removeVnffgId *optional*	Identifier of an existing VNFFG to be removed from the NS Instance. It shall be present only if updateType = "REMOVE_VNFFG".	< string > array

Name	Description	Schema
updateType *required*	The type of update. It determines also which one of the following parameters is present in the operation. Possible values include: * ADD_VNF: Adding existing VNF instance(s) * REMOVE_VNF: Removing VNF instance(s) * INSTANTIATE_VNF: Instantiating new VNF(s) * CHANGE_VNF_DF: Changing VNF DF * OPERATE_VNF: Changing VNF state, * MODIFY_VNF_INFORMATION: Modifying VNF information and/or the configurable properties of VNF instance(s) * CHANGE_EXTERNAL_VNF_CONNECTIVITY: Changing the external connectivity of VNF instance(s)ADD_SAP: Adding SAP(s) * REMOVE_SAP: Removing SAP(s) * ADD_NESTED_NS: Adding existing NS instance(s) as nested NS(s) * REMOVE_NESTED_NS: Removing existing nested NS instance(s) * ASSOC_NEW_NSD_VERSION: Associating a new NSD version to the NS instance * MOVE_VNF: Moving VNF instance(s) from one origin NS instance to another target NS instance * ADD_VNFFG: Adding VNFFG(s) * REMOVE_VNFFG: Removing VNFFG(s) * UPDATE_VNFFG: Updating VNFFG(s) * CHANGE_NS_DF: Changing NS DF * ADD_PNF: Adding PNF * MODIFY_PNF: Modifying PNF * REMOVE_PNF: Removing PNF	enum (ADD_VNF, REMOVE_VNF, INSTANTIATE_VNF, CHANGE_VNF_DF, OPERATE_VNF, MODIFY_VNF_INFORMATION, CHANGE_EXTERNAL_VNF_CONNECTIVITY, REMOVE_SAP, ADD_NESTED_NS, REMOVE_NESTED_NS, ASSOC_NEW_NSD_VERSION, MOVE_VNF, ADD_VNFFG, REMOVE_VNFFG, UPDATE_VNFFG, CHANGE_NS_DF, ADD_PNF, MODIFY_PNF, REMOVE_PNF)
updateVnffg *optional*	Specify the new VNFFG Information data to be updated for a VNFFG of the NS Instance. It shall be present only if updateType = "UPDATE_VNFFG".	< updateVnffg > array

addNestedNsData

Name	Description	Schema
nestedNsInstanceId *required*	An identifier with the intention of being globally unique.	string
nsProfileId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string

addPnfData

Name	Description	Schema
cpData *optional*	Address assigned for the PNF external CP(s).	< cpData > array

Name	Description	Schema
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *required*	Name of the PNF	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier with the intention of being globally unique.	string

cpData

Name	Description	Schema
cpInstanceI16 *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
cpProtocolData *required*	Address assigned for this CP.	< cpProtocolData > array
cpdId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	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)

Name	Description	Schema
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)

addSap

Name	Description	Schema
description *required*	Human readable description for the SAP.	string
sapName *required*	Human readable name for the SAP.	string
sapProtocolData *optional*	Parameters for configuring the network protocols on the SAP.	< sapProtocolData > array
sapdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

sapProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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

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
numDynamicAddresses *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. Representation: string of variable length.	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)

addVnfIstance

Name	Description	Schema
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

addVnffg

Name	Description	Schema
description *required*	Human readable description for the VNFFG.	string
targetNsInstanceId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
vnffgName *required*	Human readable name for the VNFFG.	string

assocNewNsdVersionData

Name	Description	Schema
newNsdId *required*	An identifier with the intention of being globally unique.	string
sync *optional*	Specify whether the NS instance shall be automatically synchronized to the new NSD by the NFVO (in case of true value) or the NFVO shall not do any action (in case of a false value) and wait for further guidance from OSS/BSS (i.e. waiting for OSS/BSS to issue NS lifecycle management operation to explicitly add/remove VNFs and modify information of VNF instances according to the new NSD). The synchronization to the new NSD means e.g. instantiating/adding those VNFs whose VNFD is referenced by the new NSD version but not referenced by the old one, terminating/removing those VNFs whose VNFD is referenced by the old NSD version but not referenced by the new NSD version, modifying information of VNF instances to the new applicable VNFD provided in the new NSD version. A cardinality of 0 indicates that synchronization shall not be done.	boolean

changeExtVnfConnectivityData

Name	Description	Schema
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
extVirtualLink *required*	Information about external VLs to change (e.g. connect the VNF to).	< extVirtualLink > array
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string

extVirtualLink

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.	< extLinkPorts > array
extVirtualLinkId *optional*	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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *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*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	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
cpProtocolData *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
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet

Name	Description	Schema
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

changeNsFlavourData

Name	Description	Schema
instantiationLevelId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
newNsFlavourId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

changeVnfFlavourData

Name	Description	Schema
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
extManagedVirtualLinks *optional*	information about internal VLs that are managed by NFVO.	< extManagedVirtualLinks > array
extVirtualLinks *optional*	Information about external VLs to connect the VNF to.	< extVirtualLinks > array
instantiationLevelId *optional*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
newFlavourId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string

extManagedVirtualLinks

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

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	An identifier with the intention of being globally unique.	string
vmfVirtualLinkDescId *optional*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	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.	< extLinkPorts > array
extVirtualLinkId *optional*	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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *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*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	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
cpProtocolData *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
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet

Name	Description	Schema
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

instantiateVnfData

Name	Description	Schema
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
extManagedVirtualLinks *optional*	Information about internal VLs that are managed by other entities than the VNFM.	< extManagedVirtualLinks > array
extVirtualLinks *optional*	Information about external VLs to connect the VNF to.	< extVirtualLinks > array
extensions *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
localizationLanguage *optional*	Localization language of the VNF to be instantiated. The value shall comply with the format defined in IETF RFC 5646.	string
metadata *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
vnfFlavourId *required*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string
vnfInstanceDescription *optional*	Human-readable description of the VNF instance to be created.	string
vnfInstanceName *optional*	Human-readable name of the VNF instance to be created.	string
vnfInstantiationLevelId *optional*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	string

Name	Description	Schema
vnfdId *required*	An identifier with the intention of being globally unique.	string

extManagedVirtualLinks

Name	Description	Schema
extManagedVirtualLinkId *optional*	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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	An identifier with the intention of being globally unique.	string
vmfVirtualLinkDescId *optional*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	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.	< extLinkPorts > array
extVirtualLinkId *optional*	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. Representation: string of variable length.	string

Name	Description	Schema
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *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*	Identifier of the VNF Virtual Link Descriptor (VLD) in the VNFD.	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
cpProtocolData *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

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

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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

Name	Description	Schema
numDynamicAddresses *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. Representation: string of variable length.	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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

modifyPnfData

Name	Description	Schema
cpData *optional*	Address assigned for the PNF external CP(s).	< cpData > array
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *optional*	Name of the PNF.	string

cpData

Name	Description	Schema
cpInstanceI16 *optional*	An Identifier that is unique within respect to a PNF. Representation: string of variable length.	string
cpProtocolData *required*	Address assigned for this CP.	< cpProtocolData > array
cpdId *optional*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

cpProtocolData

Name	Description	Schema
ipOverEthernet *optional*	This type represents network address data for IP over Ethernet.	ipOverEthernet
layerProtocol *required*	Identifier of layer(s) and protocol(s). Permitted values: IP_OVER_ETHERNET.	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
numDynamicAddresses *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. Representation: string of variable length.	string

Name	Description	Schema
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)

modifyVnfInfoData

Name	Description	Schema
extensions *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
metadata *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
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
vnfInstanceName *optional*	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfdId *optional*	An identifier with the intention of being globally unique.	string

moveVnfInstanceData

Name	Description	Schema
targetNsInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfInstanceId *optional*	Specify the VNF instance that is moved.	< string > array

operateVnfData

Name	Description	Schema
changeStateTo *required*	STARTED - The VNF instance is up and running. STOPPED - The VNF instance has been shut down.	enum (STARTED, STOPPED)
gracefulStopTimeout *optional*	The time interval (in seconds) to wait for the VNF to be taken out of service during graceful stop, before stopping the VNF.	integer
stopType *optional*	• FORCEFUL: The VNFM will stop the VNF immediately after accepting the request. • GRACEFUL: The VNFM will first arrange to take the VNF out of service after accepting the request. Once that operation is successful or once the timer value specified in the "gracefulStopTimeout" attribute expires, the VNFM will stop the VNF.	enum (FORCEFUL, GRACEFUL)
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string

updateVnffg

Name	Description	Schema
nfp *optional*	Indicate the desired new NFP(s) for a given VNFFG after the operations of addition/removal of NS components (e.g. VNFs, VLs, etc.) have been completed, or indicate the updated or newly created NFP classification and selection rule which applied to an existing NFP.	< nfp > array
nfpInfoId *optional*	Identifier(s) of the NFP to be deleted from a given VNFFG.	< string > array
vnffgInfoId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string

nfp

Name	Description	Schema
cpGroup *optional*	Group(s) of CPs and/or SAPs which the NFP passes by. Cardinality can be 0 if only updated or newly created NFP classification and selection rule which applied to an existing NFP is provided. At least a CP or an nfpRule shall be present. When multiple identifiers are included, the position of the identifier in the cpGroup value specifies the position of the group in the path.	< cpGroup > array
description *optional*	Human readable description for the NFP. It shall be present for the new NFP, and it may be present otherwise. It shall be present for the new NFP, and it may be present otherwise.	string
nfpInfoId *optional*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
nfpName *optional*	Human readable name for the NFP. It shall be present for the new NFP, and it may be present otherwise. It shall be present for the new NFP, and it may be present otherwise.	string
nfpRule *optional*	The NfpRule data type is an expression of the conditions that shall be met in order for the NFP to be applicable to the packet. The condition acts as a flow classifier and it is met only if all the values expressed in the condition are matched by those in the packet. It shall comply with the provisions defined in Table 6.5.3.40-1.	nfpRule

cpGroup

Name	Description	Schema
cpPairInfo *optional*	One or more pair(s) of ingress and egress CPs or SAPs which the NFP passes by. All CP or SAP pairs in a group shall be instantiated from connection point descriptors or service access point descriptors referenced in the corresponding NfpPositionDesc.	< cpPairInfo > array
forwardingBehaviour *optional*	Identifies a rule to apply to forward traffic to the ingress CPs or SAPs of the group. Permitted values: * ALL = Traffic flows shall be forwarded simultaneously to all CPs or SAPs of the group. * LB = Traffic flows shall be forwarded to one CP or SAP of the group selected based on a loadbalancing algorithm.	enum (ALL, LB)
forwardingBehaviourInputParameters *optional*	This type represents provides input parameters to configure the forwarding behaviour. It shall comply with the provisions defined in Table 6.5.3.73-1.	forwardingBehaviourInputParameters

cpPairInfo

Name	Description	Schema
pnfExtCpIds *optional*	Identifier(s) of the PNF CP(s) which form the pair. The presence of a single vnfExpCpId, pnfExtCpId, or sapId occurrence indicates that the CP or SAP is used both as an ingress and egress port at a particular NFP position.	< string > array
sapIds *optional*	Identifier(s) of the SAP(s) which form the pair. The presence of a single vnfExpCpId, pnfExtCpId, or sapId occurrence indicates that the CP or SAP is used both as an ingress and egress port at a particular NFP position.	< string > array
vnfExtCpIds *optional*	Identifier(s) of the VNF CP(s) which form the pair. The presence of a single vnfExpCpId, pnfExtCpId, or sapId occurrence indicates that the CP or SAP is used both as an ingress and egress port at a particular NFP position.	< string > array

forwardingBehaviourInputParameters

Name	Description	Schema
algorithmWeights *optional*	Percentage of messages sent to a CP instance. May be included if applicable to the algorithm. If applicable to the algorithm but not provided, default values determined by the VIM or NFVI are expected to be used. Weight applies to the CP instances in the order they have been created.	< integer > array
algortihmName *optional*	May be included if forwarding behaviour is equal to LB. Shall not be included otherwise. Permitted values: * ROUND_ROBIN * LEAST_CONNECTION * LEAST_TRAFFIC * LEAST_RESPONSE_TIME * CHAINED_FAILOVER * SOURCE_IP_HASH * SOURCE_MAC_HASH	enum (ROUND_ROBIN, LEAST_CONNECTION, LEAST_TRAFFIC, LEAST_RESPONSE_TIME, CHAINED_FAILOVER, SOURCE_IP_HASH, SOURCE_MAC_HASH)

nfpRule

Name	Description	Schema
destinationIpAddressPrefix *optional*	An IPV4 or IPV6 address range in CIDR format. For IPV4 address range, refer to IETF RFC 4632 [12]. For IPV6 address range, refer to IETF RFC 4291.	string
destinationPortRange *optional*	The PortRange data type provides the lower and upper bounds of a range of Internet ports. It shall comply with the provisions defined in Table 6.5.3.42-1.	destinationPortRange
dscp *optional*	For IPv4 [7] a string of "0" and "1" digits that corresponds to the 6-bit Differentiated Services Code Point (DSCP) field of the IP header. For IPv6 [28] a string of "0" and "1" digits that corresponds to the 6 differentiated services bits of the traffic class header field	string
etherDestinationAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)
etherSourceAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

Name	Description	Schema
etherType *optional*	Human readable description for the VNFFG.	enum (IPV4, IPV6)
extendedCriteria *optional*	Indicates values of specific bits in a frame.	< extendedCriteria > array
protocol *optional*	Indicates the L4 protocol, For IPv4 [7] this corresponds to the field called "Protocol" to identify the next level protocol. For IPv6 [28] this corresponds to the field is called the "Next Header" field. Permitted values: Any keyword defined in the IANA protocol registry [1], e.g.: TCP UDP ICMP	enum (TCP, UDP, ICMP)
sourceIpAddressPrefix *optional*	An IPV4 or IPV6 address range in CIDR format. For IPV4 address range, refer to IETF RFC 4632 [12]. For IPV6 address range, refer to IETF RFC 4291.	string
sourcePortRange *optional*	The PortRange data type provides the lower and upper bounds of a range of Internet ports. It shall comply with the provisions defined in Table 6.5.3.42-1.	sourcePortRange
vlanTag *optional*	Indicates a VLAN identifier in an IEEE 802.1Q-2018 tag [6] Multiple tags can be included for QinQ stacking. See note.	< string > array

destinationPortRange

Name	Description	Schema
lowerPort *required*	Identifies the lower bound of the port range. upperPort Integer **Minimum value** : `0`	integer
upperPort *required*	Identifies the upper bound of the port range. **Minimum value** : `0`	integer

extendedCriteria

Name	Description	Schema
length *required*	Indicates the number of bits to be matched.	integer

Name	Description	Schema
startingPoint *required*	Indicates the offset between the last bit of the source mac address and the first bit of the sequence of bits to be matched.	integer
value *required*	Provide the sequence of bit values to be matched.	string

sourcePortRange

Name	Description	Schema
lowerPort *required*	Identifies the lower bound of the port range. upperPort Integer **Minimum value** : `0`	integer
upperPort *required*	Identifies the upper bound of the port range. **Minimum value** : `0`	integer

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Query multiple NS LCM operation occurrences.

```
GET /ns_lcm_op_occs
```

Description

Get Operation Status. The client can use this method to query status information about multiple NS lifecycle management operation occurrences. This method shall follow the provisions specified in the Tables 6.4.9.3.2-1 and 6.4.9.3.2-2 for URI query parameters, request and response data structures, and response codes.

Parameters

Type	Name	Description	Schema
Header	**Accept** *required*	Content-Types that are acceptable for the response. Reference: IETF RFC 7231	string

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Query	**exclude_default** *optional*	Indicates to exclude the following complex attributes from the response. See clause 5.3 of ETSI GS NFV SOL 013 for details. The NFVO shall support this parameter. The following attributes shall be excluded from the NsLcmOpOcc structure in the response body if this parameter is provided: - operationParams - changedVnfInfo - error - resourceChanges	string
Query	**exclude_fields** *optional*	Complex attributes to be excluded from the response. See clause 5.3 of ETSI GS NFV SOL 013 for details. The NFVO should support this parameter.	string
Query	**fields** *optional*	Complex attributes to be included into the response. See clause 5.3 of ETSI GS NFV SOL 013 for details. The NFVO should support this parameter.	string
Query	**filter** *optional*	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV SOL 013. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the NsLcmOpOcc and in data types referenced from it shall be supported by the NFVO in the filter expression.	string
Query	**nextpage_opaque_marker** *optional*	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV SOL 013 for this resource.	string

Responses

HTTP Code	Description	Schema
200	200 OK Status information for zero or more NS lifecycle management operation occurrences has been queried successfully. The response body shall contain in an array the representations of zero or more NS instances, as defined in clause 6.5.2.3. If the "filter" URI parameter or one of the "all_fields", "fields", "exclude_fields" or "exclude_default" URI parameters was supplied in the request and is supported, the data in the response body shall have been transformed according to the rules specified in clauses 5.2.2 and 5.3.2 of ETSI GS NFV SOL 013, respectively. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV SOL 013 for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV SOL 013. **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. `Version` (string) : Version of the API used in the response. `Link` (string) : Reference to other resources. Used for paging in the present document, see clause 4.7.2.1.	< Response 200 > array

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
NsLcmOpOcc *optional*	This type represents a request a NS lifecycle operation occurrence. It shall comply with the provisions defined in Table 6.5.2.3-1.	NsLcmOpOcc

NsLcmOpOcc

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
cancelMode *optional*	Cancellation mode. The NFVO shall not start any new VNF lifecycle management and resource management operation, and shall wait for the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, to finish execution or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state. The NFVO shall not start any new VNF lifecycle management and resource management operation, shall cancel the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, and shall wait for the cancellation to finish or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.	enum (GRACEFUL, FORCEFUL)
error *optional*	The definition of the general "ProblemDetails" data structure from IETF RFC 7807 [19] is reproduced in this structure. Compared to the general framework defined in IETF RFC 7807 [19], the "status" and "detail" attributes are mandated to be included by the present document, to ensure that the response contains additional textual information about an error. IETF RFC 7807 [19] foresees extensibility of the "ProblemDetails" type. It is possible that particular APIs in the present document, or particular implementations, define extensions to define additional attributes that provide more information about the error. The description column only provides some explanation of the meaning to Facilitate understanding of the design. For a full description, see IETF RFC 7807 [19].	error
id *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
isAutomaticInvocation *required*	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in the case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS. Set to false otherwise.	boolean
isCancelPending *required*	If the LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.	boolean
lcmOperationType *required*	The enumeration NsLcmOpType represents those lifecycle operations that trigger a NS lifecycle management operation occurrence notification. Value | Description ——|——— INSTANTIATE | Represents the "Instantiate NS" LCM operation. SCALE | Represents the "Scale NS" LCM operation. UPDATE | Represents the "Update NS" LCM operation. TERMINATE | Represents the "Terminate NS" LCM operation. HEAL | Represents the "Heal NS" LCM operation.	enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL)
nsInstanceId *required*	An identifier with the intention of being globally unique.	string
operationParams *optional*	Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. The following mapping between lcmOperationType and the data type of this attribute shall apply: - INSTANTIATE: InstantiateNsRequest - SCALE: ScaleNsRequest - UPDATE: UpdateNsRequest - HEAL: HealNsRequest - TERMINATE: TerminateNsRequest This attribute shall be present if this data type is returned in a response to reading an individual resource, and may be present according to the chosen attribute selector parameter if this data type is returned in a response to a query of a container resource.	enum (INSTANTIATE, SCALE, UPDATE, HEAL, TERMINATE)

Name	Description	Schema
operationState *required*	The enumeration NsLcmOperationStateType shall comply with the provisions defined in Table 6.5.4.4-1. Value | Description ——|———— PROCESSING | The LCM operation is currently in execution. COMPLETED | The LCM operation has been completed successfully. PARTIALLY_COMPLETED | The LCM operation has been partially completed with accepTable errors. FAILED_TEMP | The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed. FAILED | The LCM operation has failed and it cannot be retried or rolled back, as it is determined that such action won't succeed. OLLING_BACK | The LCM operation is currently being rolled back. ROLLED_BACK | The LCM operation has been successfully rolled back, i.e. The state of the VNF prior to the original operation invocation has been restored as closely as possible.	enum (PROCESSING, COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK)
resourceChanges *optional*	This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the LCM operation since its start, if applicable	resourceChanges
startTime *required*	Date-time of the start of the operation.	string (date-time)
statusEnteredTime *optional*	Date-time when the current state has been entered.	string (date-time)

_links

Name	Description	Schema
cancel *optional*	This type represents a link to a resource.	cancel
continue *optional*	This type represents a link to a resource.	continue
fail *optional*	This type represents a link to a resource.	fail
nsInstance *required*	This type represents a link to a resource.	nsInstance

Name	Description	Schema
retry *optional*	This type represents a link to a resource.	retry
rollback *optional*	This type represents a link to a resource.	rollback
self *required*	This type represents a link to a resource.	self

cancel

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

continue

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

fail

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

nsInstance

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

retry

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

rollback

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

error

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)

resourceChanges

Name	Description	Schema
affectedNss *optional*	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.	< affectedNss > array
affectedPnfs *optional*	Information about the PNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedPnfs > array
affectedSaps *optional*	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.	< affectedSaps > array
affectedVls *optional*	Information about the VL instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedVls > array

Name	Description	Schema
affectedVnffgs *optional*	Information about the VNFFG instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note	< affectedVnffgs > array
affectedVnfs *optional*	Information about the VNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedVnfs > array

affectedNss

Name	Description	Schema
changeResult *required*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED - PARTIALLY_COMPLETED	enum (COMPLETED, ROLLED_BACK, FAILED, PARTIALLY_COMPLETED)
changeType *required*	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - INSTANTIATE - SCALE - UPDATE - HEAL - TERMINATE	enum (ADD, REMOVE, INSTANTIATE, SCALE, UPDATE, HEAL, TERMINATE)
nsInstanceId *required*	An identifier with the intention of being globally unique.	string
nsdId *required*	An identifier with the intention of being globally unique.	string

affectedPnfs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of change. Permitted values: - ADD - REMOVE - MODIFY	enum (ADD, REMOVE, MODIFY)
cpInstanceId *required*	Identifier of the CP in the scope of the PNF.	< string > array

Name	Description	Schema
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *optional*	Name of the PNF.	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedSaps

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - MODIFY	enum (ADD, REMOVE, MODIFY)
sapInstanceId *required*	An identifier with the intention of being globally unique.	string
sapName *optional*	Human readable name for the SAP.	string
sapdId *required*	An identifier with the intention of being globally unique.	string

affectedVls

Name	Description	Schema
changeResult *required*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)

Name	Description	Schema
changeType *required*	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY - ADD_LINK_PORT - REMOVE_LINK_PORT	enum (ADD, DELETE, MODIFY, ADD_LINK_PORT, REMOVE_LINK_PORT)
nsVirtualLinkDescId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsVirtualLinkInstanceId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
vlProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedVnffgs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY	enum (ADD, DELETE, MODIFY)
vnffgInstanceId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
vnffgdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedVnfs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)

Name	Description	Schema
changeType *optional*	Signals the type of change Permitted values: - ADD - REMOVE - INSTANTIATE - TERMINATE - SCALE - CHANGE_FLAVOUR - HEAL - OPERATE - MODIFY_INFORMATION - CHANGE_EXTERNAL_VNF_CONNECTIVITY	enum (ADD, REMOVE, INSTANTIATE, TERMINATE, SCALE, CHANGE_FLAVOUR, HEAL, OPERATE, MODIFY_INFORMATION, CHANGE_EXTERNAL_VNF_CONNECTIVITY)
changedInfo *optional*	Information about the changed VNF instance information, including VNF configurable properties,if applicable. When the "changedInfo" attribute is present, either the "changedVnfInfo" attribute or the "changedExtConnectivity" attribute or both shall be present.	changedInfo
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfName *optional*	Name of the VNF Instance.	string
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string

changedInfo

Name	Description	Schema
changedExtConnectivity *optional*		changedExtConnectivity

Name	Description	Schema
changedVnfInfo *optional*	This type represents the information that is requested to be modified for a VNF instance. The information to be modified shall comply with the associated NSD. EXAMPLE. The vnfPkgId attribute value for a particular VNF instance can only be updated with a value that matches the identifier value of a VNF package whose vnfdId is present in the associated profile of the NSD.	changedVnfInfo

changedExtConnectivity

Name	Description	Schema
extLinkPorts *optional*	Link ports of this VL.	< extLinkPorts > array
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 or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

extLinkPorts

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
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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

changedVnfInfo

Name	Description	Schema
extensions *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
metadata *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
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfInstanceName *optional*	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfdId *optional*	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 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)

Response 504

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)

Read an individual NS LCM operation occurrence resource.

```
GET /ns_lcm_op_occs/{nsLcmOpOccId}
```

Description

The client can use this method to retrieve status information about a NS lifecycle management operation occurrence by reading an individual "NS LCM operation occurrence" resource. This method shall follow the provisions specified in the Tables 6.4.10.3.2-1 and 6.4.10.3.2-2 for URI query parameters, request and response data structures, and response codes.

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

Type	Name	Description	Schema
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsLcmOpOccId** *required*	Identifier of a NS lifecycle management operation occurrence.	string

Responses

HTTP Code	Description	Schema
200	200 OK Information about a NS LCM operation occurrence has been queried successfully. The response body shall contain status information about a NS lifecycle management operation occurrence (see clause 6.5.2.3). **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response.	Response 200

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
416	416 RANGE NOT SATISFIABLE **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. `Version` (string) : Version of the API used in the response.	Response 416
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
cancelMode *optional*	Cancellation mode. The NFVO shall not start any new VNF lifecycle management and resource management operation, and shall wait for the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, to finish execution or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state. The NFVO shall not start any new VNF lifecycle management and resource management operation, shall cancel the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, and shall wait for the cancellation to finish or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.	enum (GRACEFUL, FORCEFUL)

Name	Description	Schema
error *optional*	The definition of the general "ProblemDetails" data structure from IETF RFC 7807 [19] is reproduced in this structure. Compared to the general framework defined in IETF RFC 7807 [19], the "status" and "detail" attributes are mandated to be included by the present document, to ensure that the response contains additional textual information about an error. IETF RFC 7807 [19] foresees extensibility of the "ProblemDetails" type. It is possible that particular APIs in the present document, or particular implementations, define extensions to define additional attributes that provide more information about the error. The description column only provides some explanation of the meaning to Facilitate understanding of the design. For a full description, see IETF RFC 7807 [19].	error
id *required*	An identifier with the intention of being globally unique.	string
isAutomaticInvocation *required*	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in the case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS. Set to false otherwise.	boolean
isCancelPending *required*	If the LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.	boolean
lcmOperationType *required*	The enumeration NsLcmOpType represents those lifecycle operations that trigger a NS lifecycle management operation occurrence notification. Value | Description ——|——— INSTANTIATE | Represents the "Instantiate NS" LCM operation. SCALE | Represents the "Scale NS" LCM operation. UPDATE | Represents the "Update NS" LCM operation. TERMINATE | Represents the "Terminate NS" LCM operation. HEAL | Represents the "Heal NS" LCM operation.	enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL)
nsInstanceId *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
operationParams *optional*	Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. The following mapping between lcmOperationType and the data type of this attribute shall apply: - INSTANTIATE: InstantiateNsRequest - SCALE: ScaleNsRequest - UPDATE: UpdateNsRequest - HEAL: HealNsRequest - TERMINATE: TerminateNsRequest This attribute shall be present if this data type is returned in a response to reading an individual resource, and may be present according to the chosen attribute selector parameter if this data type is returned in a response to a query of a container resource.	enum (INSTANTIATE, SCALE, UPDATE, HEAL, TERMINATE)
operationState *required*	The enumeration NsLcmOperationStateType shall comply with the provisions defined in Table 6.5.4.4-1. Value | Description ——|———— PROCESSING | The LCM operation is currently in execution. COMPLETED | The LCM operation has been completed successfully. PARTIALLY_COMPLETED | The LCM operation has been partially completed with accepTable errors. FAILED_TEMP | The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed. FAILED | The LCM operation has failed and it cannot be retried or rolled back, as it is determined that such action won't succeed. OLLING_BACK | The LCM operation is currently being rolled back. ROLLED_BACK | The LCM operation has been successfully rolled back, i.e. The state of the VNF prior to the original operation invocation has been restored as closely as possible.	enum (PROCESSING, COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK)
resourceChanges *optional*	This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the LCM operation since its start, if applicable	resourceChanges
startTime *required*	Date-time of the start of the operation.	string (date-time)
statusEnteredTime *optional*	Date-time when the current state has been entered.	string (date-time)

_links

Name	Description	Schema
cancel *optional*	This type represents a link to a resource.	cancel
continue *optional*	This type represents a link to a resource.	continue
fail *optional*	This type represents a link to a resource.	fail
nsInstance *required*	This type represents a link to a resource.	nsInstance
retry *optional*	This type represents a link to a resource.	retry
rollback *optional*	This type represents a link to a resource.	rollback
self *required*	This type represents a link to a resource.	self

cancel

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

continue

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

fail

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

nsInstance

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

retry

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

rollback

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

error

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)

resourceChanges

Name	Description	Schema
affectedNss *optional*	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.	< affectedNss > array
affectedPnfs *optional*	Information about the PNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedPnfs > array
affectedSaps *optional*	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.	< affectedSaps > array

Name	Description	Schema
affectedVls *optional*	Information about the VL instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedVls > array
affectedVnffgs *optional*	Information about the VNFFG instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note	< affectedVnffgs > array
affectedVnfs *optional*	Information about the VNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedVnfs > array

affectedNss

Name	Description	Schema
changeResult *required*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED - PARTIALLY_COMPLETED	enum (COMPLETED, ROLLED_BACK, FAILED, PARTIALLY_COMPLETED)
changeType *required*	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - INSTANTIATE - SCALE - UPDATE - HEAL - TERMINATE	enum (ADD, REMOVE, INSTANTIATE, SCALE, UPDATE, HEAL, TERMINATE)
nsInstanceId *required*	An identifier with the intention of being globally unique.	string
nsdId *required*	An identifier with the intention of being globally unique.	string

affectedPnfs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of change. Permitted values: - ADD - REMOVE - MODIFY	enum (ADD, REMOVE, MODIFY)

Name	Description	Schema
cpInstanceId *required*	Identifier of the CP in the scope of the PNF.	< string > array
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *optional*	Name of the PNF.	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedSaps

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - MODIFY	enum (ADD, REMOVE, MODIFY)
sapInstanceId *required*	An identifier with the intention of being globally unique.	string
sapName *optional*	Human readable name for the SAP.	string
sapdId *required*	An identifier with the intention of being globally unique.	string

affectedVls

Name	Description	Schema
changeResult *required*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)

Name	Description	Schema
changeType *required*	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY - ADD_LINK_PORT - REMOVE_LINK_PORT	enum (ADD, DELETE, MODIFY, ADD_LINK_PORT, REMOVE_LINK_PORT)
nsVirtualLinkDescId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsVirtualLinkInstanceId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
vlProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedVnffgs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY	enum (ADD, DELETE, MODIFY)
vnffgInstanceId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
vnffgdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedVnfs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)

Name	Description	Schema
changeType *optional*	Signals the type of change Permitted values: - ADD - REMOVE - INSTANTIATE - TERMINATE - SCALE - CHANGE_FLAVOUR - HEAL - OPERATE - MODIFY_INFORMATION - CHANGE_EXTERNAL_VNF_CONNECTIVITY	enum (ADD, REMOVE, INSTANTIATE, TERMINATE, SCALE, CHANGE_FLAVOUR, HEAL, OPERATE, MODIFY_INFORMATION, CHANGE_EXTERNAL_VNF_CONNECTIVITY)
changedInfo *optional*	Information about the changed VNF instance information, including VNF configurable properties,if applicable. When the "changedInfo" attribute is present, either the "changedVnfInfo" attribute or the "changedExtConnectivity" attribute or both shall be present.	changedInfo
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfName *optional*	Name of the VNF Instance.	string
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string

changedInfo

Name	Description	Schema
changedExtConnectivity *optional*		changedExtConnectivity

Name	Description	Schema
changedVnfInfo *optional*	This type represents the information that is requested to be modified for a VNF instance. The information to be modified shall comply with the associated NSD. EXAMPLE. The vnfPkgId attribute value for a particular VNF instance can only be updated with a value that matches the identifier value of a VNF package whose vnfdId is present in the associated profile of the NSD.	changedVnfInfo

changedExtConnectivity

Name	Description	Schema
extLinkPorts *optional*	Link ports of this VL.	< extLinkPorts > array
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 or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

extLinkPorts

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
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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

changedVnfInfo

Name	Description	Schema
extensions *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
metadata *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
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfInstanceName *optional*	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfdId *optional*	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 500

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 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

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 504

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)

Continue a NS lifecycle management operation occurrence.

```
POST /ns_lcm_op_occs/{nsLcmOpOccId}/continue
```

Description

The POST method initiates continuing an NS lifecycle operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED_TEMP" state. This method shall follow the provisions specified in the Tables 6.4.13.3.1-1 and 6.4.13.3.1-2 for URI query parameters, request and response data structures, and response codes.

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsLcmOpOccId** *required*	Identifier of a NS lifecycle management operation occurrence to be continued.	string

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Retry a NS lifecycle management operation occurrence.

```
POST /ns_lcm_op_occs/{nsLcmOpOccId}/retry
```

Description

The POST method initiates retrying a NS lifecycle management operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED_TEMP" state. This method shall follow the provisions specified in the Tables 6.4.11.3.1-1 and 6.4.11.3.1-2 for URI query parameters, request and response data structures, and response codes.

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsLcmOpOccId** *required*	Identifier of a NS lifecycle management operation occurrence to be retried. This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering a NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.	string

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Rollback a NS lifecycle management operation occurrence.

```
POST /ns_lcm_op_occs/{nsLcmOpOccId}/rollback
```

Description

The POST method initiates rolling back a NS lifecycle operation if that operation has experienced a temporary failure, i.e. the related "NS LCM operation occurrence" is in "FAILED_TEMP" state. This method shall follow the provisions specified in the Tables 6.4.12.3.1-1 and 6.4.12.3.1-2 for URI query parameters, request and response data structures, and response codes.

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string

Type	Name	Description	Schema
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsLcmOpOccId** *required*	Identifier of a NS lifecycle management operation occurrence to be rolled back. This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering a NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.	string

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Cancel a NS lifecycle management operation occurrence.

```
POST /nslcm/v1/ns_lcm_op_occs/{nsLcmOpOccId}/cancel
```

Description

The POST method initiates canceling an ongoing NS lifecycle management operation while it is being executed or rolled back, i.e. the related "NS LCM operation occurrence" is either in "PROCESSING" or "ROLLING_BACK" state. This method shall follow the provisions specified in the Tables 6.4.15.3.1-1 and 6.4.15.3.1-2 for URI query parameters, request and response data structures, and response codes.

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
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsLcmOpOccId** *required*	Identifier of a NS lifecycle management operation occurrence to be canceled. This identifier can be retrieved from the resource referenced by the "Location" HTTP header in the response to a POST request triggering a NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.	string
Body	**body** *required*	The POST request to this resource shall include a CancelMode structure in the payload body to choose between "graceful" and "forceful" cancellation.	body

body

Name	Description	Schema
cancelMode *required*	Cancellation mode. The NFVO shall not start any new VNF lifecycle management and resource management operation, and shall wait for the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, to finish execution or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state. The NFVO shall not start any new VNF lifecycle management and resource management operation, shall cancel the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, and shall wait for the cancellation to finish or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.	enum (GRACEFUL, FORCEFUL)

Responses

HTTP Code	Description	Schema
202	202 ACCEPTED **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `Location` (string (url)) : The resource URI of the created NS 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 409

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 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

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 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
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 504

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)

Mark a NS lifecycle management operation occurrence as failed.

```
POST /nslcm/v1/ns_lcm_op_occs/{nsLcmOpOccId}/fail
```

Description

The POST method marks a NS lifecycle management operation occurrence as "finally failed" if that operation occurrence is in "FAILED_TEMP" state.

Parameters

Type	Name	Description	Schema
Header	**Accept** *required*	Content-Types that are acceptable for the response. Reference: IETF RFC 7231	string

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**nsLcmOpOccId** *required*	Identifier of a NS lifecycle management operation occurrence to be marked as "failed". This identifier can be retrieved from the resource referenced by he "Location" HTTP header in the response to a POST request triggering a NS LCM operation. It can also be retrieved from the "nsLcmOpOccId" attribute in the NsLcmOperationOccurrenceNotification.	string

Responses

HTTP Code	Description	Schema
200	200 OK The state of the NS lifecycle management operation occurrence was changed successfully. The response shall include a representation of the NS lifecycle management operation occurrence resource. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response.	Response 200

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
409	409 CONFLICT **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. `Version` (string) : Version of the API used in the response.	Response 409
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
cancelMode *optional*	Cancellation mode. The NFVO shall not start any new VNF lifecycle management and resource management operation, and shall wait for the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, to finish execution or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state. The NFVO shall not start any new VNF lifecycle management and resource management operation, shall cancel the ongoing VNF lifecycle management and resource management operations in the underlying system, typically the VNFM and VIM, and shall wait for the cancellation to finish or to time out. After that, the NFVO shall put the operation occurrence into the FAILED_TEMP state.	enum (GRACEFUL, FORCEFUL)

Name	Description	Schema
error *optional*	The definition of the general "ProblemDetails" data structure from IETF RFC 7807 [19] is reproduced in this structure. Compared to the general framework defined in IETF RFC 7807 [19], the "status" and "detail" attributes are mandated to be included by the present document, to ensure that the response contains additional textual information about an error. IETF RFC 7807 [19] foresees extensibility of the "ProblemDetails" type. It is possible that particular APIs in the present document, or particular implementations, define extensions to define additional attributes that provide more information about the error. The description column only provides some explanation of the meaning to Facilitate understanding of the design. For a full description, see IETF RFC 7807 [19].	error
id *required*	An identifier with the intention of being globally unique.	string
isAutomaticInvocation *required*	Set to true if this NS LCM operation occurrence has been automatically triggered by the NFVO. This occurs in the case of auto-scaling, auto-healing and when a nested NS is modified as a result of an operation on its composite NS. Set to false otherwise.	boolean
isCancelPending *required*	If the LCM operation occurrence is in "PROCESSING" or "ROLLING_BACK" state and the operation is being cancelled, this attribute shall be set to true. Otherwise, it shall be set to false.	boolean
lcmOperationType *required*	The enumeration NsLcmOpType represents those lifecycle operations that trigger a NS lifecycle management operation occurrence notification. Value | Description ——|——— INSTANTIATE | Represents the "Instantiate NS" LCM operation. SCALE | Represents the "Scale NS" LCM operation. UPDATE | Represents the "Update NS" LCM operation. TERMINATE | Represents the "Terminate NS" LCM operation. HEAL | Represents the "Heal NS" LCM operation.	enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL)
nsInstanceId *required*	An identifier with the intention of being globally unique.	string

Name	Description	Schema
operationParams *optional*	Input parameters of the LCM operation. This attribute shall be formatted according to the request data type of the related LCM operation. The following mapping between lcmOperationType and the data type of this attribute shall apply: - INSTANTIATE: InstantiateNsRequest - SCALE: ScaleNsRequest - UPDATE: UpdateNsRequest - HEAL: HealNsRequest - TERMINATE: TerminateNsRequest This attribute shall be present if this data type is returned in a response to reading an individual resource, and may be present according to the chosen attribute selector parameter if this data type is returned in a response to a query of a container resource.	enum (INSTANTIATE, SCALE, UPDATE, HEAL, TERMINATE)
operationState *required*	The enumeration NsLcmOperationStateType shall comply with the provisions defined in Table 6.5.4.4-1. Value | Description ——|———— PROCESSING | The LCM operation is currently in execution. COMPLETED | The LCM operation has been completed successfully. PARTIALLY_COMPLETED | The LCM operation has been partially completed with accepTable errors. FAILED_TEMP | The LCM operation has failed and execution has stopped, but the execution of the operation is not considered to be closed. FAILED | The LCM operation has failed and it cannot be retried or rolled back, as it is determined that such action won't succeed. OLLING_BACK | The LCM operation is currently being rolled back. ROLLED_BACK | The LCM operation has been successfully rolled back, i.e. The state of the VNF prior to the original operation invocation has been restored as closely as possible.	enum (PROCESSING, COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK)
resourceChanges *optional*	This attribute contains information about the cumulative changes to virtualised resources that were performed so far by the LCM operation since its start, if applicable	resourceChanges
startTime *required*	Date-time of the start of the operation.	string (date-time)
statusEnteredTime *optional*	Date-time when the current state has been entered.	string (date-time)

_links

Name	Description	Schema
cancel *optional*	This type represents a link to a resource.	cancel
continue *optional*	This type represents a link to a resource.	continue
fail *optional*	This type represents a link to a resource.	fail
nsInstance *required*	This type represents a link to a resource.	nsInstance
retry *optional*	This type represents a link to a resource.	retry
rollback *optional*	This type represents a link to a resource.	rollback
self *required*	This type represents a link to a resource.	self

cancel

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

continue

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

fail

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

nsInstance

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

retry

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

rollback

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

error

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)

resourceChanges

Name	Description	Schema
affectedNss *optional*	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.	< affectedNss > array
affectedPnfs *optional*	Information about the PNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedPnfs > array
affectedSaps *optional*	Information about the nested NS instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note.	< affectedSaps > array

Name	Description	Schema
affectedVls *optional*	Information about the VL instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedVls > array
affectedVnffgs *optional*	Information about the VNFFG instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation. See note	< affectedVnffgs > array
affectedVnfs *optional*	Information about the VNF instances that were affected during the lifecycle operation, if this notification represents the result of a lifecycle operation.	< affectedVnfs > array

affectedNss

Name	Description	Schema
changeResult *required*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED - PARTIALLY_COMPLETED	enum (COMPLETED, ROLLED_BACK, FAILED, PARTIALLY_COMPLETED)
changeType *required*	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - INSTANTIATE - SCALE - UPDATE - HEAL - TERMINATE	enum (ADD, REMOVE, INSTANTIATE, SCALE, UPDATE, HEAL, TERMINATE)
nsInstanceId *required*	An identifier with the intention of being globally unique.	string
nsdId *required*	An identifier with the intention of being globally unique.	string

affectedPnfs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of change. Permitted values: - ADD - REMOVE - MODIFY	enum (ADD, REMOVE, MODIFY)

Name	Description	Schema
cpInstanceId *required*	Identifier of the CP in the scope of the PNF.	< string > array
pnfId *required*	An identifier with the intention of being globally unique.	string
pnfName *optional*	Name of the PNF.	string
pnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
pnfdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedSaps

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of lifecycle change. Permitted values: - ADD - REMOVE - MODIFY	enum (ADD, REMOVE, MODIFY)
sapInstanceId *required*	An identifier with the intention of being globally unique.	string
sapName *optional*	Human readable name for the SAP.	string
sapdId *required*	An identifier with the intention of being globally unique.	string

affectedVls

Name	Description	Schema
changeResult *required*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)

Name	Description	Schema
changeType *required*	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY - ADD_LINK_PORT - REMOVE_LINK_PORT	enum (ADD, DELETE, MODIFY, ADD_LINK_PORT, REMOVE_LINK_PORT)
nsVirtualLinkDescId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
nsVirtualLinkInstanceId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
vlProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedVnffgs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)
changeType *optional*	Signals the type of change. Permitted values: - ADD - DELETE - MODIFY	enum (ADD, DELETE, MODIFY)
vnffgInstanceId *required*	An identifier that is unique with respect to a NS. Representation: string of variable length.	string
vnffgdId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string

affectedVnfs

Name	Description	Schema
changeResult *optional*	Signals the result of change identified by the "changeType" attribute. Permitted values: - COMPLETED - ROLLED_BACK - FAILED	enum (COMPLETED, ROLLED_BACK, FAILED)

Name	Description	Schema
changeType *optional*	Signals the type of change Permitted values: - ADD - REMOVE - INSTANTIATE - TERMINATE - SCALE - CHANGE_FLAVOUR - HEAL - OPERATE - MODIFY_INFORMATION - CHANGE_EXTERNAL_VNF_CONNECTIVITY	enum (ADD, REMOVE, INSTANTIATE, TERMINATE, SCALE, CHANGE_FLAVOUR, HEAL, OPERATE, MODIFY_INFORMATION, CHANGE_EXTERNAL_VNF_CONNECTIVITY)
changedInfo *optional*	Information about the changed VNF instance information, including VNF configurable properties,if applicable. When the "changedInfo" attribute is present, either the "changedVnfInfo" attribute or the "changedExtConnectivity" attribute or both shall be present.	changedInfo
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfName *optional*	Name of the VNF Instance.	string
vnfProfileId *required*	An identifier that is unique within a NS descriptor. Representation: string of variable length.	string
vnfdId *required*	An identifier with the intention of being globally unique.	string

changedInfo

Name	Description	Schema
changedExtConnectivity *optional*		changedExtConnectivity

Name	Description	Schema
changedVnfInfo *optional*	This type represents the information that is requested to be modified for a VNF instance. The information to be modified shall comply with the associated NSD. EXAMPLE. The vnfPkgId attribute value for a particular VNF instance can only be updated with a value that matches the identifier value of a VNF package whose vnfdId is present in the associated profile of the NSD.	changedVnfInfo

changedExtConnectivity

Name	Description	Schema
extLinkPorts *optional*	Link ports of this VL.	< extLinkPorts > array
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 or by an NS instance. Information about the resource is available from the VIM.	resourceHandle

extLinkPorts

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
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 or by an NS 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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

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. Representation: string of variable length.	string
resourceProviderId *optional*	An identifier with the intention of being globally unique.	string
vimId *optional*	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. The value set of the "vimLevelResourceType" attribute is within the scope of the VIM or the resource provider and can be used as information that complements the ResourceHandle.	string

changedVnfInfo

Name	Description	Schema
extensions *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
metadata *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
vnfConfigurableProperties *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
vnfInstanceDescription *optional*	New value of the "vnfInstanceDescription" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfInstanceId *required*	An identifier with the intention of being globally unique.	string
vnfInstanceName *optional*	New value of the "vnfInstanceName" attribute in "VnfInstance", or "null" to remove the attribute.	string
vnfdId *optional*	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 409

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 500

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 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

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 504

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)

Subscribe to NS lifecycle change notifications.

```
POST /subscriptions
```

Description

The POST method creates a new subscription. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 6.4.16.3.1-1 and 6.4.16.3.1-2. Creation of two subscription resources with the same callbackURI and the same filter can result in performance degradation and will provide duplicates of notifications to the OSS, and might make sense only in very rare use cases. Consequently, the NFVO may either allow creating a subscription resource if another subscription resource with the same filter and callbackUri already exists (in which case it shall return the "201 Created" response code), or may decide to not create a duplicate subscription resource (in which case it shall return a "303 See Other" response code referencing the existing subscription resource with the same filter and callbackUri).

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
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Body	**body** *required*	Details of the subscription to be created, as defined in clause 6.5.2.2.	body

body

Name	Description	Schema
authentication *optional*		authentication
callbackUri *required*	String formatted according to IETF RFC 3986.	string (uri)
filter *optional*	This type represents a subscription filter related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.3.8-1. At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).	filter

authentication

Name	Description	Schema
authType *required*	Defines the types of Authentication / Authorization which the API consumer is willing to accept when receiving a notification. Permitted values: - BASIC: In every HTTP request to the notification endpoint, use HTTP Basic authentication with the client credentials. - OAUTH2_CLIENT_CREDENTIALS: In every HTTP request to the notification endpoint, use an OAuth 2.0 Bearer token, obtained using the client credentials grant type. - TLS_CERT: Every HTTP request to the notification endpoint is sent over a mutually authenticated TLS session, i.e. not only the server is authenticated, but also the client is authenticated during the TLS tunnel setup.	< enum (BASIC, OAUTH2_CLIENT_CREDENTIALS, TLS_CERT) > array
paramsBasic *optional*	Parameters for authentication/authorization using BASIC. Shall be present if authType is "BASIC" and the contained information has not been provisioned out of band. Shall be absent otherwise.	paramsBasic
paramsOauth2ClientCredentials *optional*	Parameters for authentication/authorization using OAUTH2_CLIENT_CREDENTIALS. Shall be present if authType is "OAUTH2_CLIENT_CREDENTIALS" and the contained information has not been provisioned out of band. Shall be absent otherwise.	paramsOauth2ClientCredentials

paramsBasic

Name	Description	Schema
password *optional*	Password to be used in HTTP Basic authentication. Shall be present if it has not been provisioned out of band.	string
userName *optional*	Username to be used in HTTP Basic authentication. Shall be present if it has not been provisioned out of band.	string

paramsOauth2ClientCredentials

Name	Description	Schema
clientId *optional*	Client identifier to be used in the access token request of the OAuth 2.0 client credentials grant type. Shall be present if it has not been provisioned out of band. The clientId and clientPassword passed in a subscription shall not be the same as the clientId and clientPassword that are used to obtain authorization for API requests. Client credentials may differ between subscriptions. The value of clientPassword should be generated by a random process.	string
clientPassword *optional*	Client password to be used in the access token request of the OAuth 2.0 client credentials grant type. Shall be present if it has not been provisioned out of band. The clientId and clientPassword passed in a subscription shall not be the same as the clientId and clientPassword that are used to obtain authorization for API requests. Client credentials may differ between subscriptions. The value of clientPassword should be generated by a random process.	string
tokenEndpoint *optional*	String formatted according to IETF RFC 3986.	string (uri)

filter

Name	Description	Schema
lcmOpNameImpactingNsComponent *optional*	Match particular LCM operation names for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (VNF_INSTANTIATE, VNF_SCALE, VNF_SCALE_TO_LEVEL, VNF_CHANGE_FLAVOUR, VNF_TERMINATE, VNF_HEAL, VNF_OPERATE, VNF_CHANGE_EXT_CONN, VNF_MODIFY_INFO, NS_INSTANTIATE, NS_SCALE, NS_UPDATE, NS_TERMINATE, NS_HEAL) > array
lcmOpOccStatusImpactingNsComponent *optional*	Match particular LCM operation status values as reported in notifications of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (START, COMPLETED, PARTIALLY_COMPLETED, FAILED, ROLLED_BACK) > array
notificationTypes *optional*	Match particular notification types. Permitted values: - NsLcmOperationOccurenceNotification - NsIdentifierCreationNotification - NsIdentifierDeletionNotification - NsChangeNotification	< enum (NsLcmOperationOccurenceNotification, NsIdentifierCreationNotification, NsIdentifierDeletionNotification, NsChangeNotification) > array
nsComponentTypes *optional*	Match particular NS component types for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChang.	< enum (VNF, PNF, NS) > array

Name	Description	Schema
nsInstanceSubscriptionFilter *optional*	This type represents subscription filter criteria to match NS instances. NOTE 1: The attributes "nsdIds", "vnfdIds" and "pnfdIds" are alternatives to reference to NS instances that are created based on certain NSDs, or contain VNF instances that are based on certain VNFDs, or contain PNFs that are based on certain PNFDs in a filter. They should not be used together in the same filter instance, but one alternative should be chosen. NOTE 2: The attributes "nsInstanceIds" and "nsInstanceNames" are alternatives to reference to particular NS Instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.	nsInstanceSubscriptionFilter
operationStates *optional*	Match particular LCM operation state values as reported in notifications of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (PROCESSING, COMPLETED, PARTIALLY_COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK) > array
operationTypes *optional*	Match particular NS lifecycle operation types for the notification of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL) > array

nsInstanceSubscriptionFilter

Name	Description	Schema
nsInstanceIds *optional*	If present, match NS instances with an instance identifier listed in this attribute.	< string > array
nsInstanceNames *optional*	If present, match NS instances with a NS Instance Name listed in this attribute.	< string > array

Name	Description	Schema
nsdIds *optional*	If present, match NS instances that were created based on a NSD identified by one of the nsdId values listed in this attribute.	< string > array
pnfdIds *optional*	If present, match NS instances that contain PNFs that are represented by a PNFD identified by one of the pnfdId values listed in this attribute.	< string > array
vnfdIds *optional*	If present, match NS instances that contain VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute.	< string > array

Responses

HTTP Code	Description	Schema
201	201 Created The subscription has been created successfully. The response body shall contain a representation of the created subscription resource. The HTTP response shall include a "Location:" HTTP header that points to the created subscription resource. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response.	Response 201
303	303 SEE OTHER **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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 201

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
callbackUri *required*	String formatted according to IETF RFC 3986.	string (uri)
filter *optional*	This type represents a subscription filter related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.3.8-1. At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).	filter
id *required*	An identifier with the intention of being globally unique.	string

_links

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

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

filter

Name	Description	Schema
lcmOpNameImpactingNsComponent *optional*	Match particular LCM operation names for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (VNF_INSTANTIATE, VNF_SCALE, VNF_SCALE_TO_LEVEL, VNF_CHANGE_FLAVOUR, VNF_TERMINATE, VNF_HEAL, VNF_OPERATE, VNF_CHANGE_EXT_CONN, VNF_MODIFY_INFO, NS_INSTANTIATE, NS_SCALE, NS_UPDATE, NS_TERMINATE, NS_HEAL) > array
lcmOpOccStatusImpactingNsComponent *optional*	Match particular LCM operation status values as reported in notifications of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (START, COMPLETED, PARTIALLY_COMPLETED, FAILED, ROLLED_BACK) > array
notificationTypes *optional*	Match particular notification types. Permitted values: - NsLcmOperationOccurenceNotification - NsIdentifierCreationNotification - NsIdentifierDeletionNotification - NsChangeNotification	< enum (NsLcmOperationOccurenceNotification, NsIdentifierCreationNotification, NsIdentifierDeletionNotification, NsChangeNotification) > array
nsComponentTypes *optional*	Match particular NS component types for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChang.	< enum (VNF, PNF, NS) > array

Name	Description	Schema
nsInstanceSubscriptionFilter *optional*	This type represents subscription filter criteria to match NS instances. NOTE 1: The attributes "nsdIds", "vnfdIds" and "pnfdIds" are alternatives to reference to NS instances that are created based on certain NSDs, or contain VNF instances that are based on certain VNFDs, or contain PNFs that are based on certain PNFDs in a filter. They should not be used together in the same filter instance, but one alternative should be chosen. NOTE 2: The attributes "nsInstanceIds" and "nsInstanceNames" are alternatives to reference to particular NS Instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.	nsInstanceSubscriptionFilter
operationStates *optional*	Match particular LCM operation state values as reported in notifications of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (PROCESSING, COMPLETED, PARTIALLY_COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK) > array
operationTypes *optional*	Match particular NS lifecycle operation types for the notification of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL) > array

nsInstanceSubscriptionFilter

Name	Description	Schema
nsInstanceIds *optional*	If present, match NS instances with an instance identifier listed in this attribute.	< string > array
nsInstanceNames *optional*	If present, match NS instances with a NS Instance Name listed in this attribute.	< string > array

Name	Description	Schema
nsdIds *optional*	If present, match NS instances that were created based on a NSD identified by one of the nsdId values listed in this attribute.	< string > array
pnfdIds *optional*	If present, match NS instances that contain PNFs that are represented by a PNFD identified by one of the pnfdId values listed in this attribute.	< string > array
vnfdIds *optional*	If present, match NS instances that contain VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute.	< string > array

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 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)

Response 504

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)

Query multiple subscriptions.

```
GET /subscriptions
```

Description

Query Subscription Information. The GET method queries the list of active subscriptions of the functional block that invokes the method. It can be used e.g. for resynchronization after error situations.

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	**Version** *required*	Version of the API requested to use when responding to this request.	string
Query	**filter** *optional*	Attribute-based filtering expression according to clause 5.2 of ETSI GS NFV SOL 013. The NFVO shall support receiving this parameter as part of the URI query string. The OSS/BSS may supply this parameter. All attribute names that appear in the LccnSubscription and in data types referenced from it shall be supported by the NFVO in the filter expression.	string
Query	**nextpage_opaque_marker** *optional*	Marker to obtain the next page of a paged response. Shall be supported by the NFVO if the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV SOL 013 for this resource.	string

Responses

HTTP Code	Description	Schema
200	200 OK The list of subscriptions has been queried successfully. The response body shall contain the representations of all active subscriptions of the functional block that invokes the method. If the NFVO supports alternative 2 (paging) according to clause 5.4.2.1 of ETSI GS NFV SOL 013 for this resource, inclusion of the Link HTTP header in this response shall follow the provisions in clause 5.4.2.3 of ETSI GS NFV SOL 013. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response. `Link` (string) : Reference to other resources. Used for paging in the present document, see clause 4.7.2.1.	< Response 200 > array

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
callbackUri *required*	String formatted according to IETF RFC 3986.	string (uri)
filter *optional*	This type represents a subscription filter related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.3.8-1. At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).	filter
id *required*	An identifier with the intention of being globally unique.	string

_links

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

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

filter

Name	Description	Schema
lcmOpNameImpactingNsComponent *optional*	Match particular LCM operation names for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (VNF_INSTANTIATE, VNF_SCALE, VNF_SCALE_TO_LEVEL, VNF_CHANGE_FLAVOUR, VNF_TERMINATE, VNF_HEAL, VNF_OPERATE, VNF_CHANGE_EXT_CONN, VNF_MODIFY_INFO, NS_INSTANTIATE, NS_SCALE, NS_UPDATE, NS_TERMINATE, NS_HEAL) > array
lcmOpOccStatusImpactingNsComponent *optional*	Match particular LCM operation status values as reported in notifications of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (START, COMPLETED, PARTIALLY_COMPLETED, FAILED, ROLLED_BACK) > array
notificationTypes *optional*	Match particular notification types. Permitted values: - NsLcmOperationOccurenceNotification - NsIdentifierCreationNotification - NsIdentifierDeletionNotification - NsChangeNotification	< enum (NsLcmOperationOccurenceNotification, NsIdentifierCreationNotification, NsIdentifierDeletionNotification, NsChangeNotification) > array
nsComponentTypes *optional*	Match particular NS component types for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChang.	< enum (VNF, PNF, NS) > array

Name	Description	Schema
nsInstanceSubscriptionFilter *optional*	This type represents subscription filter criteria to match NS instances. NOTE 1: The attributes "nsdIds", "vnfdIds" and "pnfdIds" are alternatives to reference to NS instances that are created based on certain NSDs, or contain VNF instances that are based on certain VNFDs, or contain PNFs that are based on certain PNFDs in a filter. They should not be used together in the same filter instance, but one alternative should be chosen. NOTE 2: The attributes "nsInstanceIds" and "nsInstanceNames" are alternatives to reference to particular NS Instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.	nsInstanceSubscriptionFilter
operationStates *optional*	Match particular LCM operation state values as reported in notifications of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (PROCESSING, COMPLETED, PARTIALLY_COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK) > array
operationTypes *optional*	Match particular NS lifecycle operation types for the notification of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL) > array

nsInstanceSubscriptionFilter

Name	Description	Schema
nsInstanceIds *optional*	If present, match NS instances with an instance identifier listed in this attribute.	< string > array
nsInstanceNames *optional*	If present, match NS instances with a NS Instance Name listed in this attribute.	< string > array

Name	Description	Schema
nsdIds *optional*	If present, match NS instances that were created based on a NSD identified by one of the nsdId values listed in this attribute.	< string > array
pnfdIds *optional*	If present, match NS instances that contain PNFs that are represented by a PNFD identified by one of the pnfdId values listed in this attribute.	< string > array
vnfdIds *optional*	If present, match NS instances that contain VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute.	< string > array

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 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)

Response 504

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)

Read an individual subscription resource.

```
GET /subscriptions/{subscriptionId}
```

Description

The GET method retrieves information about a subscription by reading an individual subscription resource. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 6.4.17.3.2-1 and 6.4.17.3.2-2

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	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**subscriptionId** *required*	Identifier of this subscription.	string

Responses

HTTP Code	Description	Schema
200	200 OK The operation has completed successfully. The response body shall contain a representation of the subscription resource. **Headers** : `Content-Type` (string) : The MIME type of the body of the response.This header field shall be present if the response has a non-empty message body. `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. `Version` (string) : Version of the API used in the response.	Response 200

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403

HTTP Code	Description	Schema
404	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. This response code is not appropriate in case the resource addressed by the URI is a container resource which is designed to contain child resources, but does not contain any child resource at the time the request is received. For a GET request to an existing empty container resource, a typical response contains a 200 OK response code and a payload body with an empty array. **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. `Version` (string) : Version of the API used in the response.	Response 404
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406

HTTP Code	Description	Schema
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

Response 200

Name	Description	Schema
_links *required*	Links to resources related to this resource.	_links
callbackUri *required*	String formatted according to IETF RFC 3986.	string (uri)
filter *optional*	This type represents a subscription filter related to notifications about NS lifecycle changes. It shall comply with the provisions defined in Table 6.5.3.8-1. At a particular nesting level in the filter structure, the following applies: All attributes shall match in order for the filter to match (logical "and" between different filter attributes). If an attribute is an array, the attribute shall match if at least one of the values in the array matches (logical "or" between the values of one filter attribute).	filter
id *required*	An identifier with the intention of being globally unique.	string

_links

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

self

Name	Description	Schema
href *required*	URI of a resource referenced from a notification. Should be an absolute URI (i.e. a URI that contains {apiRoot}), however, may be a relative URI (i.e. a URI where the {apiRoot} part is omitted) if the {apiRoot} information is not available.	string (url)

filter

Name	Description	Schema
lcmOpNameImpactingNsComponent *optional*	Match particular LCM operation names for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (VNF_INSTANTIATE, VNF_SCALE, VNF_SCALE_TO_LEVEL, VNF_CHANGE_FLAVOUR, VNF_TERMINATE, VNF_HEAL, VNF_OPERATE, VNF_CHANGE_EXT_CONN, VNF_MODIFY_INFO, NS_INSTANTIATE, NS_SCALE, NS_UPDATE, NS_TERMINATE, NS_HEAL) > array
lcmOpOccStatusImpactingNsComponent *optional*	Match particular LCM operation status values as reported in notifications of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChangeNotification", and shall be absent otherwise.	< enum (START, COMPLETED, PARTIALLY_COMPLETED, FAILED, ROLLED_BACK) > array
notificationTypes *optional*	Match particular notification types. Permitted values: - NsLcmOperationOccurenceNotification - NsIdentifierCreationNotification - NsIdentifierDeletionNotification - NsChangeNotification	< enum (NsLcmOperationOccurenceNotification, NsIdentifierCreationNotification, NsIdentifierDeletionNotification, NsChangeNotification) > array
nsComponentTypes *optional*	Match particular NS component types for the notification of type NsChangeNotification. May be present if the "notificationTypes" attribute contains the value "NsChang.	< enum (VNF, PNF, NS) > array

Name	Description	Schema
nsInstanceSubscriptionFilter *optional*	This type represents subscription filter criteria to match NS instances. NOTE 1: The attributes "nsdIds", "vnfdIds" and "pnfdIds" are alternatives to reference to NS instances that are created based on certain NSDs, or contain VNF instances that are based on certain VNFDs, or contain PNFs that are based on certain PNFDs in a filter. They should not be used together in the same filter instance, but one alternative should be chosen. NOTE 2: The attributes "nsInstanceIds" and "nsInstanceNames" are alternatives to reference to particular NS Instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.	nsInstanceSubscriptionFilter
operationStates *optional*	Match particular LCM operation state values as reported in notifications of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (PROCESSING, COMPLETED, PARTIALLY_COMPLETED, FAILED_TEMP, FAILED, ROLLING_BACK, ROLLED_BACK) > array
operationTypes *optional*	Match particular NS lifecycle operation types for the notification of type NsLcmOperationOccurrenceNotification. May be present if the "notificationTypes" attribute contains the value "NsLcmOperationOccurrenceNotification", and shall be absent otherwise.	< enum (INSTANTIATE, SCALE, UPDATE, TERMINATE, HEAL) > array

nsInstanceSubscriptionFilter

Name	Description	Schema
nsInstanceIds *optional*	If present, match NS instances with an instance identifier listed in this attribute.	< string > array
nsInstanceNames *optional*	If present, match NS instances with a NS Instance Name listed in this attribute.	< string > array

Name	Description	Schema
nsdIds *optional*	If present, match NS instances that were created based on a NSD identified by one of the nsdId values listed in this attribute.	< string > array
pnfdIds *optional*	If present, match NS instances that contain PNFs that are represented by a PNFD identified by one of the pnfdId values listed in this attribute.	< string > array
vnfdIds *optional*	If present, match NS instances that contain VNF instances that were created based on a VNFD identified by one of the vnfdId values listed in this attribute.	< string > array

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 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)

Response 504

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)

Terminate a subscription.

```
DELETE /subscriptions/{subscriptionId}
```

Description

The DELETE method terminates an individual subscription. This method shall support the URI query parameters, request and response data structures, and response codes, as specified in the Tables 6.4.17.3.5-1 and 6.4.17.3.5-2.

Parameters

Type	Name	Description	Schema
Header	**Authorization** *optional*	The authorization token for the request. Reference: IETF RFC 7235.	string
Header	**Version** *required*	Version of the API requested to use when responding to this request.	string
Path	**subscriptionId** *required*	Identifier of this subscription.	string

Responses

HTTP Code	Description	Schema
204	204 NO CONTENT The subscription resource has been deleted successfully. 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. `Version` (string) : Version of the API used in the response.	No Content

HTTP Code	Description	Schema
400	400 BAD REQUEST 400 code can be returned in the following specified cases, the specific cause has to be proper specified in the "ProblemDetails" structure to be returned. If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or the payload body contains a syntactically incorrect data structure), 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 response to a GET request which queries a container resource would be so big that the performance of the API producer is adversely affected, and the API producer does not support paging for the affected resource, it 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 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. 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. The use of this HTTP error response code described above is applicable to the use of the OAuth 2.0 for the authorization of API requests and notifications, as defined in clauses 4.5.3.3 and 4.5.3.4. **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. `Version` (string) : Version of the API used in the response.	Response 400

HTTP Code	Description	Schema
401	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. `Version` (string) : Version of the API used in the response.	Response 401
403	403 FORBIDDEN If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **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. `Version` (string) : Version of the API used in the response.	Response 403
405	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. **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. `Version` (string) : Version of the API used in the response.	Response 405

HTTP Code	Description	Schema
406	406 NOT ACCEPTABLE If the "Accept" 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. **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. `Version` (string) : Version of the API used in the response.	Response 406
500	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 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. `Version` (string) : Version of the API used in the response.	Response 500
503	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 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. `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. `Version` (string) : Version of the API used in the response.	Response 503

HTTP Code	Description	Schema
504	504 GATEWAY TIMEOUT If the API producer encounters a timeout while waiting for a response from an upstream server (i.e. a server that the API producer communicates with when fulfilling a request), it should respond with this response code. **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. `Version` (string) : Version of the API used in the response.	Response 504

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 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

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 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
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

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 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

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 504

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)