O-RAN.WG10.TS.TE&IV-API.0-R004-v01.00

*Technical Specification*

**O-RAN Work Group 10 (OAM for O-RAN) Topology Exposure and Inventory Application Protocols**

**Specification - Stage 3**

Copyright © 2025 by the O-RAN ALLIANCE e.V.

The copying or incorporation into any other work of part or all of the material available in this specification in any form without the prior written permission of O-RAN ALLIANCE e.V. is prohibited, save that you may print or download extracts of the material of this specification for your personal use, or copy the material of this specification for the purpose of sending to individual third parties for their information provided that you acknowledge O-RAN ALLIANCE as the source of the material and that you inform the third party that these conditions apply to them and that they must comply with them.

O-RAN ALLIANCE e.V., Buschkauler Weg 27, 53347 Alfter, Germany Register of Associations, Bonn VR 11238, VAT ID DE321720189

Contents

[Foreword 3](#_TOC_250035)

[Modal verbs terminology 3](#_TOC_250034)

[Executive summary 3](#_TOC_250033)

1. [Scope 4](#_TOC_250032)
2. [References 4](#_TOC_250031)
   1. [Normative references 4](#_TOC_250030)
   2. [Informative references 4](#_TOC_250029)
3. [Definition of terms, symbols and abbreviations 5](#_TOC_250028)
   1. [Terms 5](#_TOC_250027)
   2. [Symbols 5](#_TOC_250026)
   3. [Abbreviations 5](#_TOC_250025)
4. [Application protocol for TE&IV services 5](#_TOC_250024)
   1. [Introduction 5](#_TOC_250023)
   2. [Version conventions for the present document 5](#_TOC_250022)
5. [RESTful TE&IV Service APIs 5](#_TOC_250021)
   1. [Overview 5](#_TOC_250020)
   2. [Versioning of RESTful TE&IV Service APIs 6](#_TOC_250019)
   3. [URI structure and supported content formats 6](#_TOC_250018)
   4. [General considerations for RESTful TE&IV Service APIs 7](#_TOC_250017)
      1. [Usage of HTTP header fields 7](#_TOC_250016)
      2. [Handling of large query results 7](#_TOC_250015)
      3. [Error reporting 7](#_TOC_250014)
6. [Topology Exposure and Inventory Management Services 7](#_TOC_250013)
   1. [Topology Inventory API 7](#_TOC_250012)
      1. [Introduction 7](#_TOC_250011)
      2. [API version 7](#_TOC_250010)
      3. [Resource structure and methods 8](#_TOC_250009)
      4. [Service operations 9](#_TOC_250008)
      5. [Resources 13](#_TOC_250007)
      6. [Custom operations without associated resources 24](#_TOC_250006)
      7. [Notifications 24](#_TOC_250005)
      8. [Data model 25](#_TOC_250004)
      9. [Error Handling 27](#_TOC_250003)

[Annex A (normative) OpenAPI Specifications 28](#_TOC_250002)

* 1. [Overview 28](#_TOC_250001)
  2. [Topology & Inventory API 28](#_TOC_250000)

Annex (informative): Change History 38

# Foreword

This Technical Specification (TS) has been produced by WGn of the O-RAN Alliance.

The content of the present document is subject to continuing work within O-RAN and may change following formal O-RAN approval. Should the O-RAN Alliance modify the contents of the present document, it will be re-released by O-RAN with an identifying change of version date and an increase in version number as follows:

version xx.yy.zz where:

xx: the first digit-group is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have xx=01). Always 2 digits with leading zero if needed.

yy: the second digit-group is incremented when editorial only changes have been incorporated in the document. Always 2 digits with leading zero if needed.

zz: the third digit-group included only in working versions of the document indicating incremental changes during the editing process. External versions never include the third digit-group. Always 2 digits with leading zero if needed.

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the O-RAN Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in O-RAN deliverables except when used in direct citation.

# Executive summary

The present document specifies the TE&IV APIs suited to realize the use cases of TE&IV Service Consumer as specified in TE&IV UCR [1].

# Scope

This document specifies the TE&IV APIs used to support TE&IV services within the SMO.

# References

## Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies. In the case of a reference to a 3GPP document, a non-specific reference implicitly refers to the latest version of that document in Release 18.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, O-RAN cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

1. O-RAN TS: "Topology Exposure and Inventory Management Services Use Cases and Requirement Specification" (“TE&IV UCR”)
2. 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3"
3. 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition"
4. Semantic Versioning 2.0.0: [https://semver.org](https://semver.org/)
5. IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format"
6. IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax"
7. ETSI GS NFV-SOL 013 Rel3: "Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs"
8. IETF RFC 7951: “JSON Encoding of Data Modeled with YANG”
9. O-RAN TS: "Topology Exposure & Inventory Common Information Models and Interface Specification - Stage 2" (“TE&IV CIMI”)
10. O-RAN TS: "Topology Exposure and Inventory Data Model Specification - Stage 3" (“TE&IV DM”)
11. OpenAPI: “OpenAPI 3.0.3 Specification“, <http://spec.openapis.org/oas/v3.0.3.html>

## Informative references

[i.1] IETF RFC 4229: "HTTP Header Field Registrations"

# Definition of terms, symbols and abbreviations

## Terms

For the purpose of the present document, the following terms apply:

**API Consumer**: The TE&IV Service Consumer consuming one or more services using APIs. **API Producer**: The TE&IV Service Producer that offers its services for consumption via APIs. **Domain:** A logical grouping of topology entities and/or topology relationships.

## Symbols

Void

## Abbreviations

Void

# Application protocol for TE&IV services

## Introduction

The present document contains a realization of the services identified in TE&IV UCR [1]. This definition of the TE&IV Application Protocols Interface (TE&IV API) defined in the present document is based on the 3GPP service framework for network functions specified in 3GPP TS 29.501 [3].

## Version conventions for the present document

The version number of the present document follows the "xx.yy " versioning scheme. There could be implications for the interoperability between TE&IV Service Consumers and TE&IV Service API implementations in the SMO that are based on different versions of the present document.

An incremented "xx" version field of the present document could indicate that a new major feature (e.g., a new TE&IV Service) has been added or that an incompatible change has been made to one or more TE&IV Service APIs. An incremented "yy" version field could indicate that an optional feature has been added, a technical issue has been fixed, or that clarifications or editorial corrections have been made.

The version conventions for RESTful TE&IV Service APIs are defined in clause 5.2.

# RESTful TE&IV Service APIs

## Overview

The design of the RESTful TE&IV Service APIs is based on the services and requirements defined in TE&IV UCR [1], and on the protocol design framework as specified in 3GPP TS 29.501 [3].

The present document defines the protocols for the TE&IV service APIs listed in table 5.1-1.

###### Table 5.1-1: RESTful TE&IV Service APIs and their versions defined in the present document.

|  |  |
| --- | --- |
| **Service API** | **API version** |
| Topology & Inventory API | 1.0.0 |

## Versioning of RESTful TE&IV Service APIs

Each RESTful TE&IV Service API is versioned independently. The API version number defined in the present document contains three numerical fields following a MAJOR.MINOR.PATCH pattern, and may contain a pre-release version field, according to SemVer [4].

The API version number held by an implementation may additionally include a build metadata field, according to SemVer [4], to indicate a specific deployment. The content of this field is implementation specific; it is provided by the deployment. The

<apiVersion> path segment used in URI structures indicate the MAJOR field of the API version number. The full API version number is visible in the "version" field of the "info" object of each OpenAPI document in Annex A.

To indicate the full API version the API Consumer intends to use, the API Consumer may include the "Version" HTTP header (see IETF RFC 4229 [i.1]) in an HTTP request, in which case the header shall contain the version identifier as defined above. It is optional to include the build metadata field.

The API Producer shall include in the response the "Version" HTTP header signalling the used API version, including the build metadata if available. If the build metadata have been omitted in the request, the API Producer shall use the combination of MAJOR, MINOR, PATCH, and pre-release indicator as requested and the highest supported value for the build metadata field for that combination, if available. In case the API Consumer has not sent a "Version" header in the request, the API Producer shall use the latest available version, and signal it in the "Version" header.

NOTE: In case multiple versions are supported by an API Producer under the URI for a major version, this allows the API Consumer to request a particular version. This mechanism is referred to as "microversioning".

If the API version signalled by the API Consumer in the "Version" request header is not supported by the API Producer, the API Producer shall respond with a "406 Not Acceptable" error and may include in the response payload body a Problem Details structure providing more information on the cause of the error.

## URI structure and supported content formats

This clause specifies the URI prefix and the supported content formats applicable to the RESTful TE&IV Service APIs. All resource URIs of the APIs shall have the following prefix:

###### {apiRoot}/<apiName>/<apiVersion>/

The request URIs used in HTTP requests from the API Consumer towards the API Producer shall have the resource URI structure defined in clause 4.4.1 of 3GPP TS 29.501 [3], i.e.:

###### {apiRoot}/<apiName>/<apiVersion>/<apiSpecificResourceUriPart>

with the following components:

* The {apiRoot} shall be set as described in clause 4.4.1 of 3GPP TS 29.501 [3]; however, the restrictions w.r.t the operator specific FQDN of the host portion defined there do not apply.
* The <apiName> indicates the API name of the service interface in an abbreviated form. It is defined in the clause specifying the corresponding RESTful TE&IV Service API.
* The <apiVersion> indicates the major version (see clause 5.2) of the API and is defined in the clause specifying the corresponding RESTful TE&IV ServiceAPI.
* Each <apiSpecificResourceUriPart> represents a specific resource of the API. It is defined in the corresponding RESTful TE&IV Service API for each one of the defined resources.

For HTTP requests and responses that have message content, the content format JSON (see IETF RFC 8259 [5]) shall be supported. The JSON format shall be signalled by the content type "application/json".

All resource URIs of the API shall comply with the URI syntax as defined in IETF RFC 3986 [6]. An implementation that dynamically generates resource URI parts (individual path segments, sequences of path segments that are separated by "/", query parameter values) shall ensure that these parts only use the character set that is allowed by IETF RFC 3986 [6] for these parts.

## General considerations for RESTful TE&IV Service APIs

### Usage of HTTP header fields

HTTP headers are components of the headers section of the HTTP request and response messages. The usage of HTTP header fields shall follow the definitions in clause 4.2 of ETSI GS NFV-SOL 013 [7].

### Handling of large query results

The handling of large query results shall be supported by RESTful TE&IV Service APIs as specified in clause 5.4.2 of ETSI GS NFV-SOL 013 [7].

### Error reporting

In RESTful interfaces, application errors are mapped to HTTP errors. Since HTTP error information is generally not enough to discover the root cause of the error, additional application specific error information is typically delivered in the message content based on the ProblemDetails data type.

HTTP error responses shall be supported as specified in clause 4.8 of 3GPP TS 29.501 [3]. Protocol errors and application errors specified in table 5.2.7.1-1 of 3GPP TS 29.500 [2] shall be supported for an HTTP method if the corresponding HTTP status codes are specified as mandatory for that HTTP method in table 5.2.7.1-1 of 3GPP TS 29.500 [2].

If an HTTP method is not defined for a particular resource in the present document, that method is not supported. When that method is requested on the resource, the API Producer shall return a "405 Method Not Allowed" response. The message content may include a ProblemDetails structure.

# Topology Exposure and Inventory Management Services

## Topology Inventory API

### Introduction

This API enables the API Consumer to query the topology entities, entity types and its relationship to other topology entities as defined in TE&IV CIMI [9].

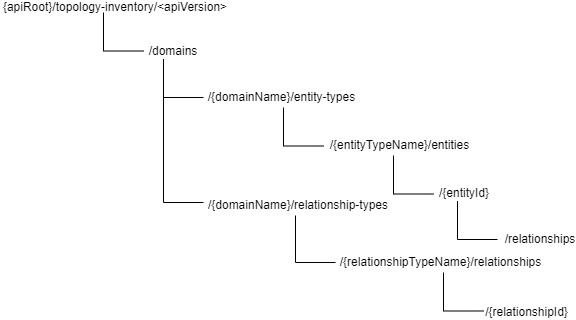
### API version

For the Topology Inventory API as specified in the present document, the MAJOR version field shall be 1, the MINOR version field shall be 0 and the PATCH version field shall be 0 (see clause 4.3.1.1 of 3GPP TS 29.501 [3] for a definition of the version fields). Consequently, the <apiVersion> URI path segment shall be set to “v1“.

### Resource structure and methods

The request URIs used in HTTP requests from the API Consumer towards the API Producer shall have the resource URI structure as defined in clause 5.2. The <apiName> resource URI variable shall be “topology-inventory”. The <apiSpecificResourceUriPart> for each resource shall be set as described in clause 6.1.5.

Figure 6.1.3-1 shows the overall resource URI structure defined for the Topology Inventory API.



###### Figure 6.1.3-1: Resource URI structure of the Topology Inventory API

Table 6.1.3-1 lists the individual resources defined for the API, the applicable HTTP methods, and the associated service operations.

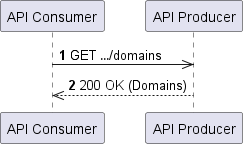
###### Table 6.1.3-1: Resources and methods overview of the Topology Inventory API

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource name** | **Resource URI** | **HTTP**  **method** | **Service Operation** |
| All topology domains | …/domains | GET | getAllDomains |
| All topology entity  types in a topology domain | …/domains/{domainName}/entity-types | GET | getTopologyEntityTypes |
| All instances of a  topology entity type in a topology domain | …/domains/{domainName}/entity- types/{entityTypeName}/entities | GET | getTopologyByEntityTypeName |
| Individual instance of  a topology entity type in a topology domain | …/domains/{domainName}/entity- types/{entityTypeName}/entities/{entityId} | GET | getTopologyById |
| All relationships of an instance of a topology entity type in a  topology domain | …/domains/{domainName}/entity- types/{entityTypeName}/entities/{entityId}  /relationships | GET | getAllRelationshipsForEntityId |
| All topology relationship types in a  topology domain | …/domains/{domainName}/relationship- types | GET | getTopologyRelationshipTypes |
| Individual topology relationship | …/domains/{domainName}/relationship- types/{relationshipTypeName}/relationshi  ps | GET | getRelationshipsByType |
| Individual topology relationship instance | …/domains/{domainName}/relationship- types/{relationshipTypeName}/relationshi  ps/{relationshipId} | GET | getRelationshipById |

### Service operations

#### Operation getAllDomains

The API Consumer uses this operation to discover the available topology domains



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer Consumer ->> Producer: GET …/domains Producer -->> Consumer: 200 OK (Domains) @enduml

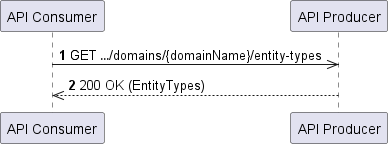
###### Figure 6.1.4.1-1: getAllDomains Operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains) to discover the available topology domains.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry a topology domain information. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getTopologyEntityTypes

The API Consumer uses this operation to discover the topology entity types in a topology domain.



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/entity-types Producer -->> Consumer: 200 OK (EntityTypes)

@enduml

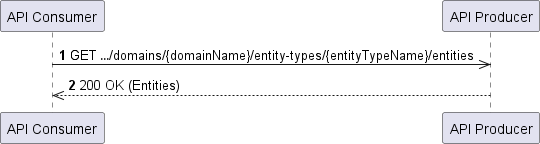
###### Figure 6.1.4.2-2: getTopologyEntityTypes Operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/entity-types) to discover the topology entities within a topology domain.
2. The API Producer shall return the HTTP GET response. On success, “200 OK“ shall be returned and the message content shall carry a topology entity types information within a topology domain. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getTopologyByEntityTypeName

The API Consumer uses this operation to query all instances of a topology entity type in a topology domain



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/entity-types/{entityTypeName}/entities Producer -->> Consumer: 200 OK (Entities)

@enduml

###### Figure 6.1.4.3-3: getTopologyByEntityTypeName Operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/entity-types/{entityTypeName}/entities) to query the instances of a specific topology entity type within a topology domain.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry the instances of a specific topology entity type information within a topology domain. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getTopologyById

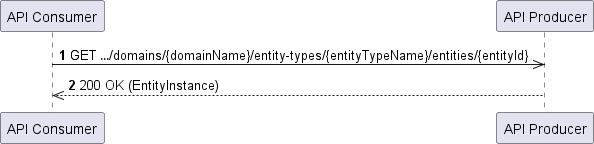
The API Consumer uses this operation to query an instance of a topology entity type in a topology domain

@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/entity- types/{entityTypeName}/entities/{entityId}

Producer -->> Consumer: 200 OK (EntityInstance) @enduml



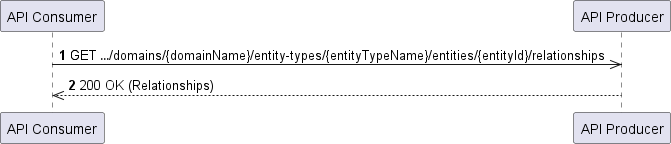
###### Figure 6.1.4.4-4: getTopologyById Operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/entity-types/{entityTypeName}/entities/{entityId}) to query an instance of a specific topology entity type within a topology domain.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry an instance of a specific topology entity type information within a topology domain. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getAllRelationshipsForEntityId

The API Consumer uses this operation to query the relationships of an instance of a topology entity type in a topology domain



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/entity- types/{entityTypeName}/entities/{entityId}/relationships Producer -->> Consumer: 200 OK (Relationships)

@enduml

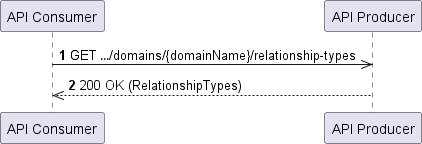
###### Figure 6.1.4.5-5: getAllRelationshipsForEntityId Operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/entity-types/{entityTypeName}/entities/{entityId}/relationships) to query the relationships of an instance of a topology entity type within a topology domain.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry the relationships information of an instance of a topology entity type within in a topology domain. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getTopologyRelationshipTypes

The API Consumer uses this operation to query all topology relationship types in a topology domain.



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/relationship-types Producer -->> Consumer: 200 OK (RelationshipTypes)

@enduml

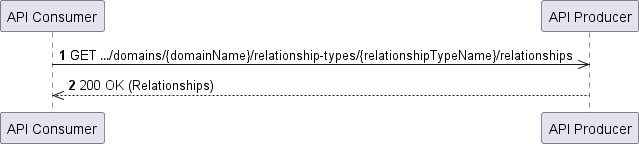
###### Figure 6.1.4.6-6: getTopologyRelationshipTypes operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/relationship-types) to query all the topology relationship types of a topology entity within a topology domain.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry the topology relationship types information. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getRelationshipsByType

The API Consumer uses this operation to query all topology relationship instances of a topology relationship type in a topology domain.



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/relationship- types/{relationshipTypeName}/relationships

Producer -->> Consumer: 200 OK (Relationships) @enduml

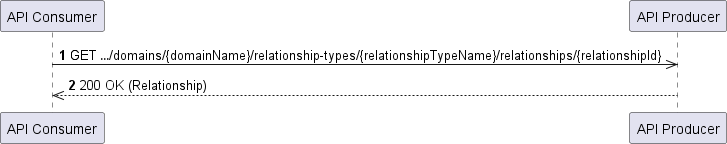
###### Figure 6.1.4.7-7: getRelationshipsByType operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/relationship-types/{relationshipTypeName}/relationships) to query all topology relationship instances of a topology relationship type.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry an array of relationship information. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

#### Operation getRelationshipById

The API Consumer uses this operation to query topology relationship instance of a topology relationship type in a topology domain.



@startuml autonumber

Participant “API Consumer“ as Consumer Participant “API Producer“ as Producer

Consumer ->> Producer: GET …/domains/{domainName}/relationship- types/{relationshipTypeName}/relationships/{relationshipId} Producer -->> Consumer: 200 OK (Relationship)

@enduml

###### Figure 6.1.4.8-8: getRelationshipById operation

The service operation is as follows:

1. The API Consumer shall send an HTTP GET request to the API Producer. The target URI shall identify the resource (…/domains/{domainName}/relationship-types/{relationshipTypeName}/relationships/{relationshipId}) to query topology relationship instance of a topology relationship type.
2. The API Producer shall return the HTTP GET response. On success, “200 OK” shall be returned and the message content shall carry relationship information of a topology relationship type. On failure, the appropriate error code shall be returned, and the response message content may contain additional error information.

### Resources

#### Overview

The following clause defines the resources for the Topology inventory API.

#### Resource: "All topology domains"

##### Description

The resource represents the available topology domains. Only the methods defined in clause 6.1.5.2.3 shall be supported by this resource.

##### Resource Definition

Resource URI: **{apiRoot}/topology-inventory/<apiVersion>/domains**

The resource URI variables supported by the resource are defined in Table 6.1.5.2.2-1.

###### Table 6.1.5.2.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2. |
| apiVersion | See clause 6.1.2. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.2.3.1-1, URI header parameters specified in 6.1.5.2.3.1-2, the request data structure specified in the table 6.1.5.2.3.1-3 and the response data structure and response code specified in the table 6.1.5.2.3.1-4.

###### Table 6.1.5.2.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| offsetParam | integer | O | 0..1 | This query parameter allows you to omit a specified number of entries before the beginning of the result set  for pagination. |
| limitParam | Integer | O | 0..1 | The query parameter provides to limit the number of entries returned for a  request for pagination. |

###### Table 6.1.5.2.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types that are acceptable. The default type acceptable for this header is  "application/json" |

###### Table 6.1.5.2.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.2.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| Domains | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries a domains structure. |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |

##### Resource Custom Operations

None.

#### Resource: "All topology entity types in a topology domain"

##### Description

The resource represents the available topology entity types in a topology domain. Only the methods defined in clause 6.1.5.3.3 shall be supported by this resource.

##### Resource Definition

###### Resource URI: {apiRoot}/topology-inventory/<apiVersion>/domains/{domainName}/entity-types

The resource URI variables supported by the resource are defined in Table 6.1.5.3.2-1.

###### Table 6.1.5.3.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2 . |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.3.3.1-1, URI header parameters specified in 6.1.5.3.3.1-2, the request data structure specified in the table 6.1.5.3.3.1-3 and the response data structure and response code specified in the table 6.1.5.3.3.1-4.

###### Table 6.1.5.3.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| offsetParam | integer | O | 0..1 | This query parameter allows you to omit a specified  number of entries before the beginning of the result set for pagination. |
| limitParam | Integer | O | 0..1 | The query parameter provides to limit the number of  entries returned for a request for pagination. |

###### Table 6.1.5.3.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types  that are acceptable. The default type acceptable for this header is "application/json" |

###### Table 6.1.5.3.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.3.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| EntityTypes | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries a EntityTypes structure. |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |

##### Resource Custom Operations

None.

#### Resource: "All instances of a topology entity type in a topology domain"

##### Description

The resource represents all the available instances of a topology entity types in a topology domain. Only the methods defined in clause 6.1.5.4.3 shall be supported by this resource.

##### Resource Definition

Resource URI:

###### {apiRoot}/topology-inventory/<apiVersion>/domains/{domainName}/entity-types/{entityTypeName}/entities

The resource URI variables supported by the resource are defined in Table 6.1.5.4.2-1.

###### Table 6.1.5.3.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2 . |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |
| entityTypeName | Name of a topology entity in a topology domain. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.4.3.1-1, URI header parameters specified in table 6.1.5.4.3.1-2 the request data structure specified in the table 6.1.5.4.3.1-3 and the response data structure and response code specified in the table 6.1.5.4.3.1-4.

###### Table 6.1.5.4.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| targetFilter | string | O | 0..1 | This query parameter specifies the entity type and attributes to be  returned in the REST response. |
| scopeFilter | string | O | 0..1 | This query parameter specifies the attributes to match on for specific Topology Entities for which the data is  to be produced. |
| offsetParam | integer | O | 0..1 | This query parameter allows you to omit a specified number of entries before the beginning of the result set  for pagination. |
| limitParam | Integer | O | 0..1 | The query parameter provides to limit the number of entries returned for a  request for pagination. |

###### Table 6.1.5.4.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types that are acceptable. The default type acceptable for  this header is "application/yang.data+json" |

###### Table 6.1.5.4.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.4.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| Entities | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries an Entities structure. |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |

##### Resource Custom Operations

None.

#### Resource: "Individual instance of a topology entity type in a topology domain"

##### Description

The resource represents the individual available instances of a topology entity types in a topology domain. Only the methods defined in clause 6.1.5.5.3 shall be supported by this resource.

##### Resource Definition

Resource URI:

###### {apiRoot}/topology-inventory/<apiVersion>/domain/{domainName}/entity-types/{entityTypeName}/entities/{entityId}

The resource URI variables supported by the resource are defined in Table 6.1.5.5.2-1.

###### Table 6.1.5.5.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2 . |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |
| entityTypeName | Name of a topology entity in a topology domain. |
| entityId | Identifier of an individual topology entity. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.5.3.1-1, URI header parameters specified in table 6.1.5.5.3.1-2, the request data structure specified in the table 6.1.5.5.3.1-3 and the response data structure and response code specified in the table 6.1.5.5.3.1-4.

###### Table 6.1.5.5.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.5.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types  that are acceptable. The default type acceptable for this header is "application/yang.data+json" |

###### Table 6.1.5.5.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.5.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| EntityInstance | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries an EntityInstance which is an encapsulated object reference to the data model for schema definition of Topology Entity. (NOTE). |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |
| NOTE: The attributes of a Topology Entity defined in the TE&IV Information Model [9] may be represented in multiple data modelling languages. Irrespective of the data modelling language specified in the Data Model specification [10], in the response, these attributes are encoded in JSON format. For example, attributes of the Topology Entity in the TE&IV Information Model [9] are represented in the YANG data modelling language. This YANG definition will be encoded as JSON using RFC 7951 and is encapsulated in this object. | | | | |

##### Resource Custom Operations

None.

#### Resource: "All relationships of an instance of a typology entity type in a topology domain"

##### Description

The resource represents the relationships of an instance of a topology entity types in a topology domain. Only the methods defined in clause 6.1.5.6.3 shall be supported by this resource.

##### Resource Definition

Resource URI:

###### {apiRoot}/topology-inventory/<apiVersion>/domains/{domainName}/entity- types/{entityTypeName}/entities/{entityId}/relationships

The resource URI variables supported by the resource are defined in Table 6.1.5.6.2-1.

###### Table 6.1.5.6.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2 . |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |
| entityTypeName | Name of a topology entity in a topology domain. |
| entityId | Identifier of an individual topology entity. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.6.3.1-1, URI header parameters specified in table 6.1.5.6.3.1-2, the request data structure specified in the table 6.1.5.6.3.1-3 and the response data structure and response code specified in the table 6.1.5.6.3.1-4.

###### Table 6.1.5.6.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| offsetParam | integer | O | 0..1 | This query parameter allows you to omit a specified number of entries before the beginning of the result set  for pagination. |
| limitParam | Integer | O | 0..1 | The query parameter provides to limit the number of entries returned for a  request for pagination. |

###### Table 6.1.5.6.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types  that are acceptable. The default type acceptable for this header is "application/yang.data+json" |

###### Table 6.1.5.6.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.6.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| Relationships | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries a Relationships structure. |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |

##### Resource Custom Operations

None.

#### Resource: "All topology relationship types in a topology domain"

##### Description

The resource represents all the topology relationship types in a topology domain. Only the methods defined in clause 6.1.5.7.3 shall be supported by this resource.

##### Resource Definition

Resource URI:

###### {apiRoot}/topology-inventory/<apiVersion>/domains/{domainName}/relationship-types

The resource URI variables supported by the resource are defined in Table 6.1.5.7.2-1.

###### Table 6.1.5.7.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2 . |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.7.3.1-1, URI header parameters specified in table 6.1.5.7.3.1-2, the request data structure specified in the table 6.1.5.7.3.1-3 and the response data structure and response code specified in the table 6.1.5.7.3.1-4.

###### Table 6.1.5.7.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| offsetParam | integer | O | 0..1 | This query parameter allows you to omit a specified number of entries before the beginning of the result set  for pagination. |
| limitParam | Integer | O | 0..1 | The query parameter provides to limit the number of entries returned for a  request for pagination. |

###### Table 6.1.5.7.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types that are acceptable. The default type acceptable for  this header is "application/json" |

###### Table 6.1.5.7.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request |

###### Table 6.1.5.7.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| RelationshipTypes | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries a RelationshipTypes structure. |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |

##### Resource Custom Operations

None.

#### Resource: "All topology relationship instances of a topology relationship type"

##### Description

The resource represents all the topology relationship instances of a topology relationship type in a topology domain. Only the methods defined in clause 6.1.5.8.3 shall be supported by this resource.

##### Resource Definition

Resource URI:

###### {apiRoot}/topology-inventory/<apiVersion>/domains/{domainName}/relationship- types/{relationshipTypeName}/relationships

The resource URI variables supported by the resource are defined in Table 6.1.5.8.2-1.

###### Table 6.1.5.8.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2. |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |
| relationshipTypeName | A topology relationship type name such as NF Deployment NF Relationship. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.8.3.1-1, URI header parameters specified in table 6.1.5.8.3.1-2, the request data structure specified in the table 6.1.5.8.3.1-3 and the response data structure and response code specified in the table 6.1.5.8.3.1-4.

###### Table 6.1.5.8.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| offsetParam | integer | O | 0..1 | This query parameter allows you to omit a specified number of entries before the beginning of the result set  for pagination. |
| limitParam | Integer | O | 0..1 | The query parameter provides to limit the number of entries returned for a  request for pagination. |
| targetFilter | string | O | 0..1 | This query parameter specifies the  entity type and attributes to be returned in the REST response. |
| scopeFilter | string | O | 0..1 | This query parameter specifies the attributes to match on for specific Topology Entities for which the data is  to be produced. |

###### Table 6.1.5.8.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types that are acceptable. The default type acceptable for  this header is "application/yang.data+json" |

###### Table 6.1.5.8.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request. |

###### Table 6.1.5.8.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| Relationships | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries a Relationships structure. |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |

##### Resource Custom Operations

None.

#### Resource: "Individual topology relationship instance of a topology relationship type"

##### Description

The resource represents all the topology relationship instances of a topology relationship type in a topology domain. Only the methods defined in clause 6.1.5.9.3 shall be supported by this resource.

##### Resource Definition

Resource URI:

###### {apiRoot}/topology-inventory/<apiVersion>/domains/{domainName}/relationship- types/{relationshipTypeName}/relationships/{relationshipId}

The resource URI variables supported by the resource are defined in Table 6.1.5.9.2-1.

###### Table 6.1.5.9.2-1: Resource URI variables for the resource

|  |  |
| --- | --- |
| **Name** | **Definition** |
| apiRoot | See clause 5.2 . |
| apiVersion | See clause 6.1.2. |
| domainName | Name of the Topology Domain. |
| relationshipTypeName | A topology relationship type name. |
| relationshipId | An identifier of an individual topology relationship instance. |

##### Resource Standard Methods

GET

This method shall support the URI query parameters specified in table 6.1.5.9.3.1-1, URI header parameters specified in table 6.1.5.9.3.1-2, the request data structure specified in the table 6.1.5.9.3.1-3 and the response data structure and response code specified in the table 6.1.5.9.3.1-4.

###### Table 6.1.5.9.3.1-1: URI query parameters supported by the GET method on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request. |

###### Table 6.1.5.9.3.1-2: URI header parameters supported by the GET method on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data type** | **P** | **Cardinality** | **Description** |
| Accept | string | M | 1 | This header is used to specify response media types that are acceptable. The default type acceptable for  this header is "application/yang.data+json". |

###### Table 6.1.5.9.3.1-3: Data structures supported by the GET request body on this resource.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Description** |
| N/A |  |  | There is no object in the message content of the GET request. |

###### Table 6.1.5.9.3.1-4: Data structures supported by the GET response body on this resource.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **P** | **Cardinality** | **Response**  **codes** | **Description** |
| Relationship | M | 1 | 200 OK | The operation was successful.  The message content of the GET response carries a Relationship instance which is an encapsulated object reference to the data model for schema definition of Topology Relationship.  (NOTE). |
| ProblemDetails | O | 0..1 | 4xx/5xx | The operation has failed, and the message  content may contain Problem description details. |
| NOTE: The attributes of a Topology Relationship defined in the TE&IV Information Model [9] may be represented in multiple data modelling languages. Irrespective of the data modelling language specified in the Data Model specification [10], in the response, these attributes are encoded in JSON format. For example, attributes of the Topology Relationship in the TE&IV Information Model [9] are represented in the YANG data modelling language. This YANG definition will be encoded as JSON using RFC 7951 and is encapsulated in this object. | | | | |

##### Resource Custom Operations

None.

### Custom operations without associated resources

None.

### Notifications

None.

### Data model

#### Structured data types

##### Overview

The following clauses define the structured data types and their attributes to be used by the service API.

##### Data type: Domains

The Domains data type represents domains. It contains the attributes defined in table 6.1.8.1.2-1.

###### Table 6.1.8.1.2-1: Definition of type Domains.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| items | array(Domain) | M | 0..N | A list of domains. |

##### Data type: Domain

The Domain data type represents the properties of a topology domain. It contains the attributes defined in table 6.1.8.1.3-1.

###### Table 6.1.8.1.3-1: Definition of type Domain.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| domainName | string | M | 1 | A name of the topology domain. |
| entityTypes | Href | M | 1 | A reference to entity types of the Topology  Entity. |
| relationshipTypes | Href | M | 1 | A reference to relationship types of the  Topology Relationship. |

##### Data type: EntityTypes

The EntityTypes data type represents entity type within a topology domain. It contains the attributes defined table 6.1.8.1.4-1.

###### Table 6.1.8.1.4-1: Definition of type EntityTypes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| items | array(EntityType) | M | 1..N | An entity type within a topology domain. |

##### Data type: EntityType

The EntityType data type represents a type within a topology domain.

###### Table 6.1.8.1.5-1: Definition of type EntityType.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| entityTypeName | string | M | 1 | A name of a topology Entity Type. |
| entities | Href | M | 1 | A reference to topology entities of the  specific topology Entity Type |

##### Data type: Entities

The Entities data type represents all instances of a specific topology entity type within a topology domain. It contains the attributes defined in table 6.1.8.1.6-1.

###### Table 6.1.8.1.6-1: Definition of type Entities.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| items | array(object) | M | 1..N | Topology Entity instances encapsulated as array of objects and refers to the data model for schema definition of Topology Entity. (NOTE). |
| NOTE: The attributes of a Topology Entity defined in the TE&IV Information Model [9] may be represented in multiple data modelling languages. Irrespective of the data modelling language specified in the Data Model specification [10], in the response, these attributes are encoded in JSON format. For example, attributes of the Topology Entity in the TE&IV Information Model [9] are represented in the YANG data modelling language. This YANG definition will be encoded as JSON using RFC 7951 and is encapsulated in this object. | | | | |

##### Data type: Relationships

The Relationships data type represents relationships information of an instance of a topology entity type within a topology domain. It contains the attributes defined in table 6.1.8.1.7-1.

###### Table 6.1.8.1.7-1: Definition of type Relationships.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| items | array(object) | M | 1..N | Encapsulated object reference to the data model for schema definition of Topology Relationship. (NOTE). |
| NOTE: The attributes of the Topology Relationship of a Topology Entity defined in the TE&IV Information Model  [9] may be represented in multiple data modelling languages. Irrespective of the data modelling language specified in the Data Model specification [10], in the response, these attributes are encoded in JSON format. For example, attributes of the Topology Relationship of a Topology Entity in a TE&IV REL- RAN-Cloud domain in the TE&IV Information Model [9] are represented in the YANG data modelling language. This YANG definition will be encoded as JSON using RFC 7951 and is encapsulated in this object. | | | | |

##### Data type: RelationshipTypes

The Relationships data type represents all the relationships information of a topology entity type within a topology domain. It contains the attributes defined in table 6.1.8.1.8-1.

###### Table 6.1.8.1.8-1: Definition of type RelationshipTypes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| items | array(RelationshipType) | M | 1..N | All the relationship type information  within a topology domain. |

##### Data type: RelationshipType

The relationship information in a topology domain. It contains the attributes defined in table 6.1.8.1.9-1.

###### Table 6.1.8.1.9-1: Definition of type RelationshipType.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| relationshipTypeName | string | M | 1 | Name of a Topology Relationship. |
| relationships | Href | M | 1 | URI to the relationships of the Topology  Relationship Type. |

##### Data type: Href

This attribute provide reference to entity type and relationship type. It contains the attributes defined in table 6.1.8.1.10-1.

###### Table 6.1.8.1.10-1: Definition of type Href

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Data type** | **P** | **Cardinality** | **Description** |
| href | string | M | 1 | The format of the string according to IETF  RFC 3986 [6]. |

#### Simple data types and enumerations

The following clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

##### Simple data types

For this service API, no simple data types are defined in the present document

##### Enumerations

For this service API, no enumerations are defined in the present document.

### Error Handling

#### General

For the Topology Inventory API, HTTP error responses shall be supported as specified in clause 4.8 of 3GPP TS 29.501 [3]. Protocol errors and application errors specified in table 5.2.7.2-1 of 3GPP TS 29.500 [2] shall be supported for an HTTP method if the corresponding HTTP status codes are specified as mandatory for that HTTP method in table 5.2.7.1-1 of 3GPP TS 29.500 [2].

In addition, the requirements in the following clauses are applicable for the Topology Inventory API.

#### Protocol Errors

No specific protocol errors are defined in the present document.

#### Application Errors

No additional application errors defined in the present document.

# Annex A (normative) OpenAPI Specifications

## Overview

This Annex formally specifies the RESTful Topology & Inventory API by defining OpenAPI documents in YAML format that comply with the OpenAPI 3.0.3 Specification [11].

The Open API specifications of the RESTful Topology & Inventory API provided in this annex are versioned as described in clause 5.2.

## Topology & Inventory API

openapi: 3.0.3 info:

title: 'Topology & Inventory API' version: 1.0.0

description: |

API for Topology and Inventory API.

Topology and Inventory data is the information that represents entities in a telecommunications network and the relationships between them that provide insight into a particular aspect of the network of importance to specific use cases. Topology and Inventory data can be derived from inventory, configuration, or other data.

Topology and Inventory supports several topology domains. A domain is a grouping of topology and inventory entities that handles topology and inventory data.

Entities are enabling the modelling and storage of complex network infrastructure and relationships.

A relationship is a bi-directional connection between two entities, one of which is the originating side (A-side) and the other is the terminating side (B-side). The order of the sides matters since it defines the relationship itself which must be unique.

Topology and Inventory API provides the capabilities to fetch topology data. Using the filtering options, it is possible to define more specific query requests.

© 2024, O-RAN ALLIANCE.

All rights reserved.

tags:

* name: Entities and relationships

description: "Provides the capability to retrieve topology and inventory entities and relationships."

externalDocs:

description: 'O-RAN.WG10.TE&IV-API.0-R004-v01.00'

url: ['https://www.o-ran.org/specifications'](http://www.o-ran.org/specifications%27)

servers:

* url: '{apiRoot}/topology-inventory/v1' variables:

apiRoot:

description: 'apiRoot as defined in clause 5.3 in O-RAN.WG10.TE&IV-API' default: 'https://example.com'

paths:

/domains: get:

description: Get all the available topology domains. tags:

- Entities and relationships

summary: Get all the available topology domains. operationId: "getAllDomains"

parameters:

* + - $ref: '#/components/parameters/acceptJsonInHeader'
    - $ref: '#/components/parameters/offsetParam'
    - $ref: '#/components/parameters/limitParam' responses:

'200':

description: OK content:

application/json: schema:

$ref: '#/components/schemas/Domains'

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/entity-types: get:

description: Get all the available topology entity types within a domain. tags:

* + - Entities and relationships

summary: Get all the available topology entity types within a domain. operationId: "getTopologyEntityTypes"

parameters:

* + - $ref: '#/components/parameters/acceptJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/offsetParam'
    - $ref: '#/components/parameters/limitParam' responses:

'200':

description: OK content:

application/json: schema:

$ref: '#/components/schemas/EntityTypes'

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/entity-types/{entityTypeName}/entities: get:

description: Get all Topology Entity instances of a specific Topology Entity type. tags:

* + - Entities and relationships

summary: Get all Topology Entity instances of a specific Topology Entity type. operationId: "getTopologyByEntityTypeName"

parameters:

* + - $ref: '#/components/parameters/acceptJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/entityTypeNameInPath'
    - $ref: '#/components/parameters/targetFilterOptionalInQuery'
    - $ref: '#/components/parameters/scopeFilterOptionalInQuery'
    - $ref: '#/components/parameters/offsetParam'
    - $ref: '#/components/parameters/limitParam' responses:

'200':

description: OK content:

application/json: schema:

$ref: '#/components/schemas/Entities'

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/entity-types/{entityTypeName}/entities/{entityId}: get:

description: Get a specific Topology Entity instance of a Topology Entity type tags:

* + - Entities and relationships

summary: Get a specific Topology Entity instance of a Topology Entity type operationId: "getTopologyById"

parameters:

* + - $ref: '#/components/parameters/acceptYangJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/entityTypeNameInPath'
    - $ref: '#/components/parameters/entityIdInPath' responses:

'200':

description: OK content:

application/yang.data+json: schema:

type: object

description: "Encapsulated object reference to the data model for schema definition of

Topology Entity"

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/entity-types/{entityTypeName}/entities/{entityId}/relationships: get:

description: Get all relationships for a specific Topology Entity instance of a Topology Entity type tags:

* + - Entities and relationships

summary: Get all relationships for a specific Topology Entity instance of a Topology Entity type operationId: "getAllRelationshipsForEntityId"

parameters:

* + - $ref: '#/components/parameters/acceptYangJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/entityTypeNameInPath'
    - $ref: '#/components/parameters/entityIdInPath'
    - $ref: '#/components/parameters/offsetParam'
    - $ref: '#/components/parameters/limitParam' responses:

'200':

description: OK content:

application/yang.data+json: schema:

$ref: '#/components/schemas/Relationships'

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/relationship-types: get:

description: Get all the available Topology Relationship types. tags:

* + - Entities and relationships

summary: Get all the available topology relationship types. operationId: "getTopologyRelationshipTypes"

parameters:

* + - $ref: '#/components/parameters/acceptJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/offsetParam'
    - $ref: '#/components/parameters/limitParam' responses:

'200':

description: OK content:

application/json: schema:

$ref: '#/components/schemas/RelationshipTypes'

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/relationship-types/{relationshipTypeName}/relationships: get:

description: Get all the available Topology Relationships of a specific relationship type name tags:

* + - Entities and relationships

summary: Get all the available Topology Relationships of a specific relationship type name. operationId: "getRelationshipsByType"

parameters:

* + - $ref: '#/components/parameters/acceptYangJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/relationshipTypeNameInPath'
    - $ref: '#/components/parameters/targetFilterOptionalInQuery'
    - $ref: '#/components/parameters/scopeFilterOptionalInQuery'
    - $ref: '#/components/parameters/offsetParam'
    - $ref: '#/components/parameters/limitParam' responses:

'200':

description: OK content:

application/yang.data+json: schema:

$ref: '#/components/schemas/Relationships'

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503'

/domains/{domainName}/relationship-types/{relationshipTypeName}/relationships/{relationshipId}:

get:

description: Get a specific Topology Relationship instance of a Topology Relationship type. tags:

* + - Entities and relationships

summary: Get a specific Topology Relationship instance of a Topology Relationship type. operationId: "getRelationshipById"

parameters:

* + - $ref: '#/components/parameters/acceptYangJsonInHeader'
    - $ref: '#/components/parameters/domainNameInPath'
    - $ref: '#/components/parameters/relationshipTypeNameInPath'
    - $ref: '#/components/parameters/relationshipIdInPath' responses:

'200':

description: OK content:

application/yang.data+json: schema:

type: object

description: "Encapsulated object reference to the data model for schema definition of Topology Relationship"

'400':

$ref: '#/components/responses/400' '401':

$ref: '#/components/responses/401' '403':

$ref: '#/components/responses/403' '404':

$ref: '#/components/responses/404' '406':

$ref: '#/components/responses/406' '414':

$ref: '#/components/responses/414' '429':

$ref: '#/components/responses/429' '500':

$ref: '#/components/responses/500' '502':

$ref: '#/components/responses/502' '503':

$ref: '#/components/responses/503' components:

schemas: Domains:

type: object title: Domains properties:

items:

type: array items:

type: object properties:

domainName: type: string

entityTypes:

$ref: '#/components/schemas/Href' relationshipTypes:

$ref: '#/components/schemas/Href'

required: ['domainName', 'entityTypes', 'relationshipTypes'] required: ['items']

EntityTypes: type: object

title: EntityTypes properties:

items:

type: array items:

type: object properties:

entityTypeName: type: string

entities:

$ref: '#/components/schemas/Href' required: ['entityTypeName', 'entities']

required: ['items'] Entities:

type: object title: Entities properties:

items:

type: array items:

type: object

description: 'Encapsulated object reference to the data model for schema definition of Topology Entity'

required: ['items'] RelationshipTypes:

type: object

title: RelationshipTypes properties:

items:

type: array items:

type: object properties:

relationshipTypeName: type: string

relationships:

$ref: '#/components/schemas/Href'

required: ['relationshipTypeName', 'relationships'] required: ['items']

Relationships: type: object

title: Relationships properties:

items:

type: array items:

type: object

description: "'Encapsulated object reference to the data model for schema definition of Topology Relationships"

required: ['items'] ProblemDetails:

description: 'A problem detail to carry details in an HTTP response according to RFC 7807' type: object

properties: type:

description: 'a URI reference according to IETF RFC 3986 that identifies the problem type' type: string

title:

description: 'human-readable summary of the problem type' type: string

status:

description: 'the HTTP status code' type: number

detail:

description: 'human-readable explanation ' type: string

instance:

description: 'URI reference that identifies the specific occurrence of the problem' type: string

Href:

type: object title: Href properties:

href:

type: string format: uri-template

responses: '400':

description: 'Bad Request' content:

application/problem+json:

schema:

$ref: '#/components/schemas/ProblemDetails'

'401':

description: 'Unauthorized' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'403':

description: 'Forbidden' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'404':

description: 'Not Found' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'405':

description: 'Method Not Allowed' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'406':

description: 'Not Acceptable' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'409':

description: 'Conflict' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'411':

description: 'Length Required' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'413':

description: 'Payload Too Large' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'414':

description: 'URI Too Large' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'415':

description: 'Unsupported Media Type' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'429':

description: 'Too Many Requests' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'500':

description: 'Internal Server Error' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'502':

description: 'Bad Gateway' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails'

'503':

description: 'Service Unavailable' content:

application/problem+json: schema:

$ref: '#/components/schemas/ProblemDetails' parameters:

acceptJsonInHeader: name: Accept

in: header required: true schema:

type: string

default: application/json acceptYangJsonInHeader:

name: Accept in: header required: true schema:

type: string

default: application/yang.data+json offsetParam:

name: offset in: query

description: This query parameter allows you to omit a specified number of entries before the beginning of the result set for pagination.

required: false schema:

type: integer default: 0

minimum: 0 limitParam:

name: limit in: query

description: The query parameter provides to limit the number of entries returned for a request for pagination.

required: false schema:

type: integer default: 500

minimum: 1

maximum: 500 domainNameInPath:

name: domainName in: path

description: domain name required: true

schema:

type: string entityIdInPath:

name: entityId in: path required: true schema:

type: string relationshipIdInPath:

name: relationshipId in: path

required: true schema:

type: string entityTypeNameInPath:

name: entityTypeName in: path

required: true schema:

type: string relationshipTypeNameInPath:

name: relationshipTypeName in: path

required: true schema:

type: string targetFilterOptionalInQuery:

name: targetFilter

description: Use \*targetFilter\* to specify the entity type and attributes to be returned in the REST response. The value for

\*targetFilter\* can also be a list of entity types and attributes. in: query

required: false schema:

type: string scopeFilterOptionalInQuery:

name: scopeFilter

description: Use \*scopeFilter\* to specify the attributes to match on. The value for \*scopeFilter\* can also be a list of entity types and attributes. scopeFilter returns a boolean.

in: query required: false schema:

type: string

# Annex (informative):

Change History

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision** | **Description** |
| 2024.10.01 | 00.00.01 | Initial proposed skeleton for the TE&IV Application Protocols Specification |
| 2024.11.21 | 01.00 | Implemented the following CRs for the November 2024 train: ERI-2024.08.14-WG10-CR- 0110-TE&IV\_API\_Content-v02, ERI-2024.09.23-WG10-CR-0115-TE&IV\_API\_Service  operation and Resources-v02, ERI-2024.10.16-WG10-CR-0125-TE&IVOpenAPI-v01,  ERI-2024.11.07-WG10-CR-0126-TE&IV\_API\_version\_update-v01 |