ERICSSON

Collection Management: Import and Export Collections REST Northbound Interface

Interwork Description

1/155 19-CNA 403 2993 Uen F

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

Contents

1 Introduction

This document describes the Network Explorer Import and Export Collections REST Northbound Interface

The Network Explorer Import and Export Collections REST Northbound Interface is a feature deployed in the ENM domain management for external systems; it offers a REST based interface for external systems to access collections via import and export operations

2 Target Audience

Developers implementing a client application that performs bulk transfers of collections into or out of ENM systems.

3 Base URL

This RESTful service is accessible from https://<customer-domain>/network-explorer-import/v1/collection/

4 Network Explorer Import and Export Collections Use Cases

4.1 Export Collections of Objects Use Case

The Export Collections of Objects use case allows the user to export to file a given list of collections of objects

The output file is a zip file **collections_<timestamp>.zip** containing one text file for each collection exported. The files are named with the convention: *<collection name>_<owner>_<sharing permission>.txt*

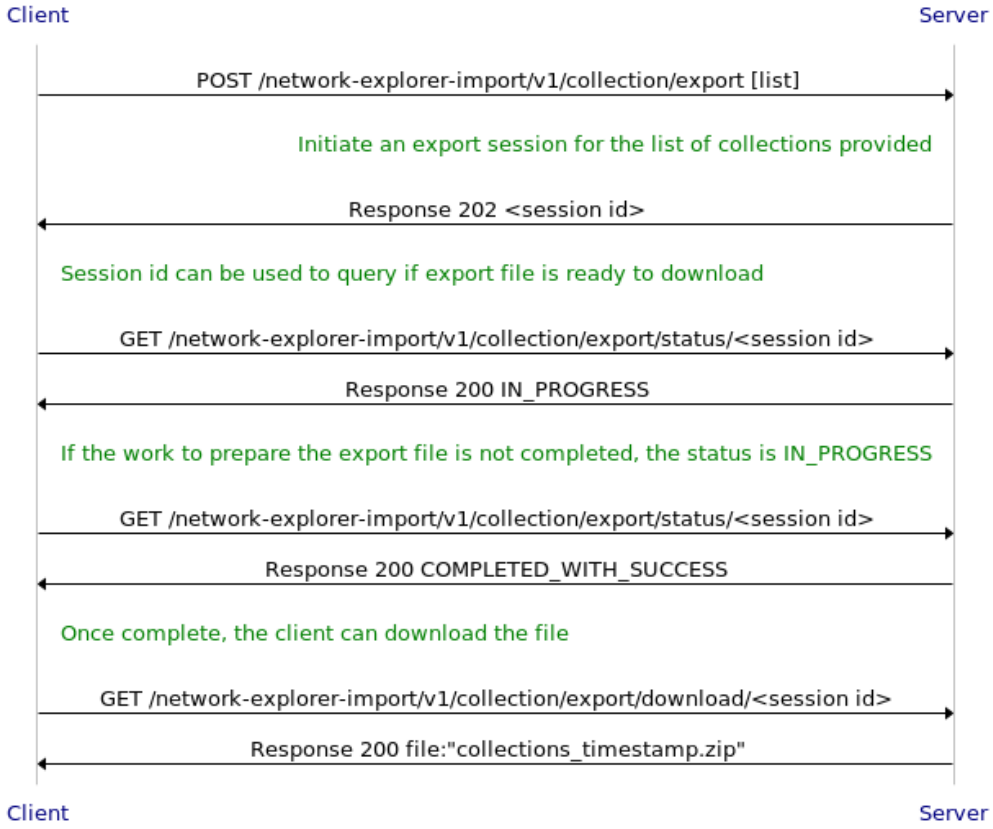
In order to perform the use case the following steps are executed:

— initiate the export providing the list of collection to export

— monitor the progress of the export session till the export completion

— download the export output file

The following picture provides an overview of the scenario



4.2 Export Collections of Collections Use Case

The Export Collections of Collections use case allows to export a given list of collections of collections to a file.

The output file is a zip file **collections_<timestamp>.zip** containing one text file for each collection exported. The files are named with the convention: *<collection name>_<owner>_<sharing permission>.txt*

The output zip file also includes the file topology.txt. The topology.txt describes the relationship of the exported collections.

In order to perform the use case the following steps are executed:

— initiate the export providing the list of collection to export

— monitor the progress of the export session till the export completion

— download the export output file

4.3 Import Collections of Objects Use Case

The Import Collections of Object use case allows the user to import a given list of collections of objects from a file.

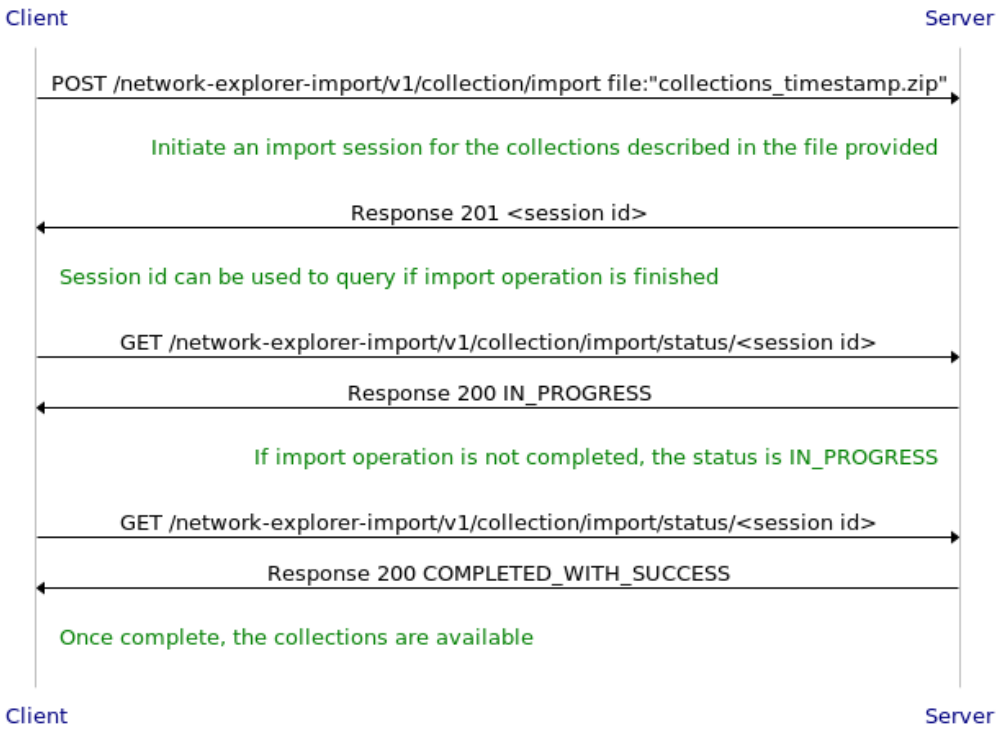
The input file is a zip file containing one text file for each collection to be imported.

In order to perform the use case the following steps are executed:

— initiate the import providing the file

— monitor the progress of the import session until completion

The following picture provides an overview of the scenario



4.4 Import Collections of Collections Use Case

The Import Collections of Collections use case allows the user to import a given list of collections of collections from a file.

The input file is a zip file containing one text file for each collection to be imported.

The input zip file also includes the file topology.txt. The topology.txt describes the relationship of the imported collections.

In order to perform the use case the following steps are executed:

— initiate the import providing the file

— monitor the progress of the import session until completion

5 Network Explorer Import and Export Collections REST Northbound Interface Prerequisites

5.1 Authorization

Users must have create capabilities for collections to import

Users must have read capabilities on the collections to export

5.2 Authentication

Prerequisites

— A valid Username and Password.

— A fully deployed ENM System

Authentication through NBI

Establish a session with the ENM system.

For details, refer to:*Export ENM PKI Root CA Certificate*" section in the *ENM Public Key Infrastructure System Administrator Guide, 2/1543-AOM 901 151-3*.

— "Export ENM PKI root CA Certificate" section in the ENM Public key Infrastructure System Administrator Guide,2/1543-AOM 901 151-3.

— "Establish a User Session over REST" and " cURL examples for Identity and Access Management" section in the ENM Identity and Access Management Programmers Guide, 19817-CNA 403 3016

6 Network Explorer Import and Export Collections REST Northbound Interface API Details

6.1 Initiate an Export Session for a given Collection of Objects

Initiates an export collections session for the given collections of objects starting the asynchronous process to produce the export file.

The session identifier of the created export session is included in the response and it shall be used for monitoring progress of the export operation.

6.1.1 Request

Request Method

The Request Method is POST

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/export

Request Body

The Request Body includes the attributes reported in the following table:

Name	Description	Type
collections	List of collection Ids of the collections of objects to be exported. The collection Id is a unique identifier referencing the collection of objects to be exported	Array of String

Request sample

```
POST /network-explorer-import/v1/collection/export

{   "collections":
    [
        "281474977035132",
        "645687876897442"
    ]
}
```

6.1.2 Response

The response includes the unique session identifier assigned to the newly created export session and the stat time of the export session as reported in the following table

Name	Description	Type
sessionId	The session identifier of the created export session.	String
startTime	The session start time in posix long format (millisec).	Number
timeZone	The ID of the default timezone for the server.	String

Response sample

```
application/json

{
    "sessionId": "r6732069-9938-4173-84b6-9e6666412047",
    "startTime": 1528450705214,
    "timeZone": "Europe/Dublin"
}
```

6.2 Initiate an Export Session for a given Collection of Collections

Initiates an export session for the given collections of collections of starting the asynchronous process to produce the export file.

The session identifier of the created export session is included in the response and it shall be used for monitoring progress of the export operation.

6.2.1 Request

Request Method

The Request Method is POST

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/export/nested

Request Body

The Request Body includes the attributes reported in the following table

Name	Description	Type
collections	List of collection Ids. The collection Id is a unique identifier referencing the collection of collection to be exported	Array of String

Request sample

```
POST /network-explorer-import/v1/collection/export/nested

{
   "collections": [
        "281474977035455",
        "645687876897389"
    ]
}
```

6.2.2 Response

The response includes the unique session identifier assigned to the newly created export session and the stat time of the export session as reported in the following table

Name	Description	Type
sessionId	The session identifier of the created export session.	String
startTime	The session start time in posix long format (millisec).	Number
timeZone	The ID of the default timezone for the server.	String

Response sample

```
application/json

{
    "sessionId": "r6732069-9938-4173-84b6-9e6666412047",
    "startTime": 1528450705214,
    "timeZone": "Europe/Dublin"
}
```

6.3 Retrieve the Status of a given Export Session

Retrieve the current status of the export session with the given session identifier. This API is valid for both monitoring the progress of sessions for both collections of objects and collections of collections.

The status "IN_PROGRESS" is returned when there are still collections to process.

The status "COMPLETED_WITH_SUCCESS" is returned when all the provided collections have been processed successfully.

The status "COMPLETED_WITH_ERRORS" is returned when all the provided collections have been processed but errors were faced processing some of them.

The status "ABORTED" is returned when the export session has been aborted by using abort API

The status "FAILED" is returned when the session has been interrupted by an unrecoverable issue.

The session identifier is provided as path parameter

6.3.1 Request

Request Method

The Request Method is GET

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/export/status/

Request sample

```
GET /network-explorer-import/v1/collection/export/status/56498462
```

6.3.2 Response

The response includes the unique session identifier assigned to the newly created session and the start time of the session as reported in the following table:

Name	Description	Type
status	The current status of the given session.	String (SessionStatus) Enum:"IN_PROGRESS" "COMPLETED_WITH_SUCCESS" "COMPLETED_WITH_ERRORS" "ABORTED" "FAILED"
failure	The list of any errors detected during the collection session processing. This attribute is included in the response only in case the status is COMPLETED_WITH_ERRORS	array of object (Failure) where : Failure { collectionId String ID of the failed collection. collectionName String Name of the failed collection collectionOwner String Owner of the failed collection collectionCategory String Category of the failed collection reason String Failure reason

Name	Description	Type
		internalErrorCode Integer Error code provided by the service }
processed	The number of the processed collections in the session at the time of the query.	Integer
total	The number of the provided collections.	Integer

Response sample

```
{
  "status": "COMPLETED_WITH_ERRORS",
  "failure": [
     {
       "collectionId": "276002926481297",
       "collectionName": "MyCollection01",
       "collectionOwner": "User1",
       "collectionCategory": "Private",
       "reason": "Collection already exists",
       "internalErrorCode": 11001
    },
    {
       "collectionId": "281474977035132",
       "collectionName": "",
       "collectionOwner": "",
       "collectionCategory": "",
       "reason": "Collection not found",
       "internalErrorCode": 11002

    },
    {
       "collectionId": "569341830415685",
       "collectionName": "MyCollection02",
       "collectionOwner": "User2",
       "collectionCategory": "Public",
       "reason": "Server Error. Check Logs",
       "internalErrorCode": 11003
    },
    {
        "collectionId": "845143123904782",
        "collectionName": "MyNestedCollection01",
        "collectionOwner": "User1",
        "collectionCategory": "Public",
        "reason": "Nested collection partially exported",
        "internalErrorCode": 11004
     }
    ],
   "processed": 85,
   "total": 85
  }
```

6.4 Download the Exported Collections File

Downloads the exported collections file given the export collection session identifier.

The provided session identifier shall refer to a completed export session otherwise an invalid request error is returned.

6.4.1 Request

Request Method

The Request Method is GET

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/export/download/

Request sample

```
GET /network-explorer-import/v1/collection/export/download/2549846556
```

6.4.2 Response

The export output file is returned

Name	Description	Type
file	The exported collections data file is a compressed file (zip format) named: *collections_"timestamp".zip*. The compressed file holds a formatted text file for each exported collection named: *"collectionName_collectionOwner_collectionCategory".txt*. In the case of exporting collections of collections, the zip file will also include the file *topology.txt*. The topology.txt describes the topology of the exported collections .	String (binary)

6.5 Abort an Ongoing Export Session

The export session with the given session identifier is stopped and marked for deletion.

The provided session identifier refers to an in progress export session otherwise an invalid request error is returned.

6.5.1 Request

Request Method

The Request Method is DELETE

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/export/

Request sample

```
DELETE /network-explorer-import/v1/collection/export/826697845
```

6.6 Initiate an Import Session for a given Collection of Objects

Initiates an import session for the given collections of objects, starting an asynchronous process to create all collections in the system.

The session identifier of the created import session is included in the response and it is used for monitoring progress of the import operation.

6.6.1 Request

Request Method

The Request Method is POST

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/import

Request Body

The Request Body includes the multipart fields reported in the following table

Field Name	Description	Type

importConfig	Configuration options
	parentId (optional) The parent Id references the collection id of a collection of collections into which the client wishes to import the file contents
		Object (ImportConfig) where:
		ImportConfig {
		parentId String
		}

file	A compressed file (zip format). The compressed file holds a formatted text file for each collection to import.	String (binary)

Request sample

```
POST /network-explorer-import/v1/collection/import/

multipart/form-data

importConfig
{
   "parentId": "7497703545545"
}

file
...(binary data)
```

Sample CURL Request

```
curl --location --request POST 'https://<customer-domain>/network-explorer-impor →
t/v1/collection/import' \

--header 'X-Tor-UserId: <enm_username>' \

--header 'x-tor-application: networkexplorer' \

--header 'accept: application/json' \

--form 'importConfig={"parentId": 1}' \

--form 'file=@collections_export_sample.zip'
```

Sample Collections Export File Contents

```
version:3.0

name:"exportcollectionssample"

sharing:PUBLIC

label:created_by_import

userId:"null"

contains:Objects

FDN:NetworkElement=LTE01:NetworkElement

FDN:NetworkElement=RadioNode01:NetworkElement
```

6.6.2 Response

The response includes the unique session identifier assigned to the newly created import session and the stat time of the import session as reported in the following table

Name	Description	Type
sessionId	The session identifier of the created import session.	String
startTime	The session start time in posix long format (millisec).	Number

Name	Description	Type
timeZone	The ID of the default timezone for the server.	String

Response sample

```
application/json

{
    "sessionId": "r7326069-9938-4173-84b6-9e6666412047",
    "startTime": 1538450705214,
    "timeZone": "Europe/Dublin"
}
```

6.7 Initiate an Import Session for a given Collection of Collections

Initiates an import session for the given collections of collections, starting an asynchronous process to create all collections in the system.

The session identifier of the created import session is included in the response and it shall be used for monitoring progress of the import operation.

6.7.1 Request

Request Method

The Request Method is POST

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/import

Request Body

The Request Body include the fields reported in the following table

Name	Description	Type

importConfig	Configuration options
	parentId (optional) The parent Id references the collection id of a collection of collections into which the client wishes to import the file contents
		Object (ImportConfig) where:
		ImportConfig {
		parentId String
		}

file	A compressed file (zip format). The compressed file holds a formatted text file for each collection to import. In the case of importing collections of collections, the zip	String (binary)

Name	Description	Type
	file must also include the file *topology.txt*. The topology.txt describes the topology of the collections to import.	

Request sample

```
POST  /network-explorer-import/v1/collection/import/

multipart/form-data

file
...(binary data)
```

6.7.2 Response

The response includes the unique session identifier assigned to the newly created import session and the stat time of the import session as reported in the following table

Name	Description	Type
sessionId	The session identifier of the created import session.	String
startTime	The session start time in posix long format (millisec).	Number
timeZone	The ID of the default timezone for the server.	String

Response sample

```
application/json

{
    "sessionId": "r7326069-9938-4173-84b6-9e6666412047",
    "startTime": 1538450705214,
    "timeZone": "Europe/Dublin"
}
```

6.8 Retrieve the Status of a given Import Session

Retrieve the current status of the import session with the given session identifier. This API is valid for monitoring the progress of sessions for both collections of objects and collections of collections.

The status "IN_PROGRESS" is returned when there are still collections to process.

The status "COMPLETED_WITH_SUCCESS" is returned when all the provided collections have been processed successfully.

The status "COMPLETED_WITH_ERRORS" is returned when all the provided collections have been processed but errors were faced processing some of them.

The status "ABORTED" is returned when the export session has been aborted by using abort API

The status "FAILED" is returned when the session has been interrupted by an unrecoverable issue.

The session identifier is provided as path parameter

6.8.1 Request

Request Method

The Request Method is GET

Request Path

The Request Path is formed by constructing a URI with the following schema:

/network-explorer-import/v1/collection/import/status/

Request sample

```
GET /network-explorer-import/v1/collection/import/status/56498462
```

6.8.2 Response

The response includes the unique session identifier assigned to the newly created session and the start time of the session as reported in the following table:

Name	Description	Type
status	The current status of the given session.	String (SessionStatus) Enum:"IN_PROGRESS" "COMPLETED_WITH_SUCCESS" "COMPLETED_WITH_ERRORS" "ABORTED" "FAILED"
failure	The list of any errors detected during the collection session processing. This attribute is included in the response only in case the status is COMPLETED_WITH_ERRORS	Array of Object (Failure) Failure { collectionName String Name of the failed collection collectionOwner String Owner of the failed collection collectionCategory String Category of the failed collection reason String Failure reason internalErrorCode Integer Error code provided by the service }
processed	The number of collections processed in the session at the time of the query.	Integer
total	The number of the provided collections.	Integer

Response sample

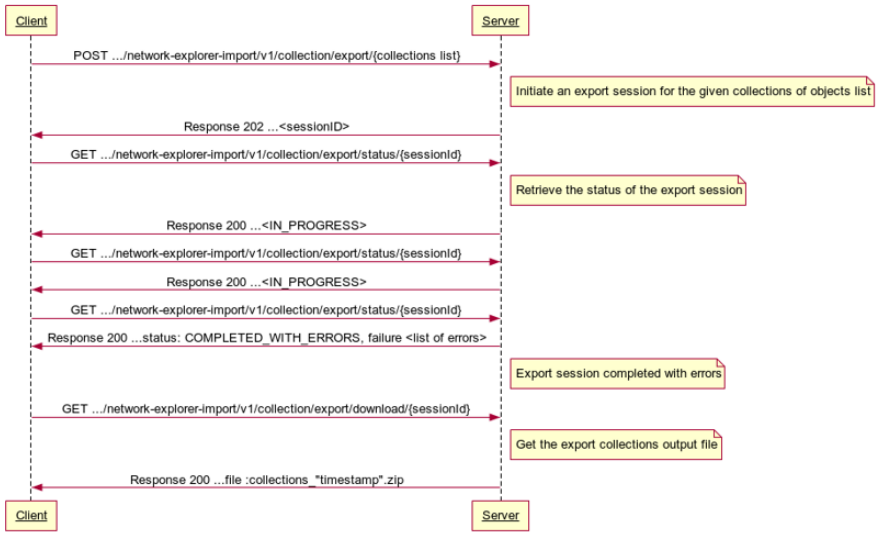

```
{
  "status": "COMPLETED_WITH_ERRORS",
  "failure": [
     {
       "collectionName": "MyCollection01",
       "collectionOwner": "User1",
       "collectionCategory": "Private",
       "reason": "Collection already exists",
       "internalErrorCode": 11001
     }
   ],
   "processed": 85,
   "total": 85
  }
```


6.9 Additional Export Collections Use Case Scenarios

The following sequence diagram refers to the export collections scenario in which errors are detected during export so only a subset of the given collections are exported .

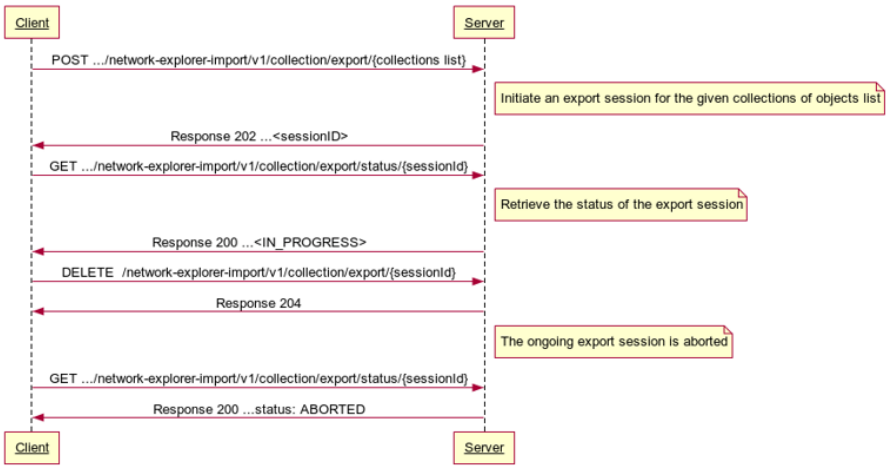
In this case the export session completes in status COMPLETED_WITH_ERRORS and the list of errors detected during export is included.

The export collections output file includes the collections successfully exported



The following sequence diagram reports the case of an ongoing export session being aborted by the client.

After aborting, the status of the export session is set to ABORTED


Client
Server
POST .../network-explorer-import/v1/collection/export/{collections list}
Initiate an export session for the given collections of objects list
Response 202 ...<sessionID>
GET .../network-explorer-import/v1/collection/export/status/{sessionId}
Retrieve the status of the export session
Response 200 ...<IN_PROGRESS>
DELETE /network-explorer-import/v1/collection/export/{sessionId}
Response 204
The ongoing export session is aborted
GET .../network-explorer-import/v1/collection/export/status/{sessionId}
Response 200 ...status: ABORTED
Client
Server