Oracle® Communications

Cloud Native Core, Network Slice Selection Function REST Specification Guide

Release 25.1.201
G32781-02
October 2025

Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide, Release 25.1.201

G32781-02

Contents

Preface

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

- For Technical issues such as creating a new Service Request (SR), select **1**.
- For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.
- For Hardware, Networking and Solaris Operating System Support, select **3**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Acronyms

The following table provides information on the acronyms and terminology used in the document:

Table 1 Acronyms

Field	Description
3GPP	3rd Generation Partnership Project
AMF	Access and Mobility Management Function
ASM	Aspen Service Mesh
HTTPS	Hypertext Transfer Protocol Secure
KPI	Key Performance Indicator
NF	Network Function
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
Network Slice	A logical network that provides specific network capabilities and network characteristics.
PEI	Permanent Equipment Identifier
PLMN	Public Land Mobile Network
PDU	Protocol Data Unit
RAN	Radio Access Network
Requested NSSAI	NSSAI provided by the UE to the serving PLMN during registration.
Allowed NSSAI	NSSAI provided by the serving PLMN during a registration procedure, indicating the S-NSSAIs values the UE could use in the serving PLMN for the current registration area.
Configured NSSAI	NSSAI provisioned in the UE applicable to one or more PLMNs.
EANAN	Empty Authorized NSSAI Availability Notification
SEPP	Security Edge Protection Proxy
SBA	Service Based Architecture
SBI	Service Based Interface
SSC	Session and Service Continuity
SST	Slice or Service type
SD	Slice Differentiator
SMF	Session Management Function
S-NSSAI	Single Network Slice Selection Assistance Information
SUBMOD	Subscription Modification
Subscribed S-NSSAI	5G uses this as a default when the UE doesn’t send a Requested NSSAI
SUPI	Subscription Permanent Identifier
TA	Tracking Area
TAC	Tracking Area Code
TAI	Tracking Area Identifier
UDM	Unified Data Management

Table 1 (Cont.) Acronyms

Field	Description
UDR	Unified Data Repository
UE	User Equipment

What's New in This Guide

This section lists the documentation updates for Release 25.1.2xx in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

Release 25.1.201- G32781-02, October 2025

There are no updates to this document in this release.

Release 25.1.200- G32781-01, July 2025

- Performed the following updates in the Runtime Log Level Update APIs:
  - Updated the example API calls.
  - Added endpoint details for each microservice.
  - Added payload examples for each microservice.
  - Added payload details table for each microservice.
- Updated the descriptions and notes for Request or Response Body Parameters in the Policy Mapping Configuration to Enable Overload Configuration section.

1 Introduction

This document provides information about how to configure the services and manageable objects in Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF) using Representational State Transfer Application Program Interfaces (REST APIs).

1.1 Overview

Oracle Communications Network Slice Selection Function (NSSF) selects the network slicing instance (NSI), determines the allowed Network Slice Selection Assistance Information (NSSAI) and Access and Mobility Management Function (AMF) to serve the User Equipment (UE). AMF can retrieve NRF, NSI ID, and target AMFs as part of UE initial registration and Packet Data Unit (PDU) establishment procedure.

Network Slice Selection Function supports the following functionalities:

- NSSF enables the Access and Mobility Management Function (AMF) to perform initial registration and (Protocol Data Unit) PDU session establishment.
- AMF can retrieve NRF, NSI ID and target AMFs as part of UE initial registration and PDU establishment procedure.
- NSSF uses an NF Service Consumer (AMF) to update the S-NSSAI(s) the AMF supports and notify any change in status.
- NSSF selects the network slicing instance (NSI) and determines the authorized Network Slice Selection Assistance Information (NSSAIs) and AMF to serve the UE.
- NSSF interaction with NRF allows retrieving specific NF services to be used for the registration request.
- NSSF also allows a mechanism for registration and subsequent notification.

The NSSF supports the above functions through following services:

- Nnssf_NSSelection Service
- Nnssf_NSSAIAvailability Service

> **Note**
>
> The performance and capacity of the NSSF system may vary based on the call model, Feature or Interface configuration, and underlying CNE and hardware environment.

For more information about the NSSF supported services, see *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

1.2 OpenAPI Specification

NSSF supports the OpenAPI version 3.0.0.

1.3 References

See the following documents for more information about NSSF:

- *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*
- *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide*

2
NSSF Managed Objects

This chapter provides information about THE MANAGED OBJECT used in Oracle Communications Oracle Communication Cloud Native Core Network Selection Slice Function (NSSF).

2.1 Primitive Tables

Behavior

Table 2-1 Behavior

Attribute	Description	Data type
accessType	This is a mandatory parameter. For information, see Enumeration section.	accessType
nsiProfiles	This is a mandatory parameter. It contains an array of NsiProfile Map.	Array(NsiProfileMap)

TargetAmfSet

Table 2-2 TargetAmfSet

Attribute	Description	Data type
regionId	This is a mandatory parameter. Region id of TargetAmfSet.	String
setId	This is a mandatory parameter. Set id of TargetAmfSet.	String
saliance	This is an optional parameter. saliance of TargetAmfSet.	Integer

CandidateAmf

Table 2-3 CandidateAmf

Attribute	Description	Data type
instanceId	This is a mandatory parameter. It contains Instance id of the AMF.	String

Timespan

Table 2-4 Timespan

Attribute	Description	Data type
startTime	Start time in hh:mm:ss	Time

Table 2-4 (Cont.) Timespan

Attribute	Description	Data type
endTime	End time in hh:mm:ss	Time

Plmnid

Table 2-5 Plmnid

Attribute	Description	Data type
mcc	Mobile Country Code	String
mnc	Mobile Network Code	String

Snssai

Table 2-6 Snssai

Attribute	Description	Data type
sst	Slice or Service Type	Integer
sd	Slice Differentiator	String

TacRange

Table 2-7 TacRange

Attribute	Description	Data type
startTac	A 4/6 digit hexadecimal number that identifies starting value of a Tracking Area in a TAC range.	String
endTac	A 4/6 digit hexadecimal number that identifies ending value of a Tracking Area in a TAC range.	String

MappingOfSnssai

Table 2-8 MappingOfSnssai

Attribute	Description	Data type
servingSnssai	This IE contains the S-NSSAI value of serving network (5G).	Snssai
homeSnssai	This IE contains the mapped S-NSSAI value of home network (4G).	Snssai

2.2 Enumerations

Grant

Table 2-9 Grant

Value	Description
"ALLOWED"	Allowed signifies SNSSAI is allowed in TAI.
"RESTRICTED_TA"	S-NSSAI is not allowed for Tracking Area.
"RESTRICTED_PLMN	S-NSSAI is not allowed for PLMN.

Access Type

Table 2-10 Access Type

Value	Description
"3GPP_ACCESS"	Specifies 5G network
"NON_3GPP_ACCESS"	Specifies non 5G network

DayofWeek

Table 2-11 DayofWeek

Value	Description
"MONDAY"	Monday, day of the week
"TUESDAY"	Tuesday, day of the week
"WEDNESDAY"	Wednesday, day of the week
"THURSDAY"	Thursday, day of the week
"FRIDAY"	Friday, day of the week
"SATURDAY"	Saturday, day of the week
"SUNDAY"	Sunday, day of the week

3
NSSF REST Specifications

This chapter provides information about REST specifications used in Oracle Communications Oracle Communication Cloud Native Core, Network Selection Slice Function (NSSF).

NSSF can be configured using Helm configurations, REST APIs, and Cloud Native Core (CNC) Console. The NSSF deployment configurations are performed during NSSF installation using Helm, and a few configurations are modified using REST APIs. REST configurations can also be performed using the Cloud Native Configuration (CNC) Console.

For HELM configurations, see *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide*.

For the configurations using CNC Console, see *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

3.1 AMF Set

AMF Set is a mandatory Managed Object (MO). It contains information on supported AMF Sets per PLMN and their salience. This MO can also be configured automatically with default 0 salience by configuring NSI Profile. The operator must ensure that the salience of AMF Sets in a PLMN is not the same. If more than one AMF set supports an NSI, the set with the lowest salience is picked.

- Request_Type: POST
- URL: *http://{apiRoot}/nnssf-configuration/v1/amfset*
- REST message sample - AMF Set
- Content-Type: application/json
- Initiated By: API Invoker

AMF Set Supported REST APIs

An operator can configure AMF Set by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the AMF Set Managed Object:

Table 3-1 AMF Set - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/amfset	Create an AMF Set	**Data Type:** amfset **HTTP Method:** POST
/nnssf-configuration/v1/amfset	Read all AMF Sets	**Data Type:** amfset **HTTP Method:** GET
/nnssf-configuration/v1/amfset/{mcc}-{mnc}-{sst}-{sd}	Update an AMF Set	**Data Type:** amfset **HTTP Method:** PUT

Table 3-1 (Cont.) AMF Set - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/amfset/{mcc}-{mnc}-{sst}-{sd}	Read an AMF Set	**Data Type:** amfset **HTTP Method:** GET
/nnssf-configuration/v1/amfset/{mcc}-{mnc}-{sst}-{sd}	Delete an AMF Set	**Data Type:** amfset **HTTP Method:** DELETE

Request or Response Body Parameters

The AMF Set Managed Object is mandatory. It comprises the MCC (Mobile Country Code), MNC (Mobile Network Code), AMF Region ID, AMF Set ID and AMF Pointer. This AMF Set contains information on supported AMF Sets per PLMN and their salience. This Managed Object can also be configured automatically with default 0 salience by configuring NSI-Profile.

Table 3-2 AMF Set - Parameters

Parameter	Description	Details
`regionId`	This is a mandatory parameter. Region ID of the target AMF list.	**Data Type:** String
`setId`	This is a mandatory parameter. Set ID of the target AMF list.	**Data Type:** String
`mcc`	This is a mandatory parameter. Mobile Country Code.	**Data Type:** String
`mnc`	This is a mandatory parameter. Mobile Network Code.	**Data Type:** String
`salience`	This is an optional parameter. Order of importance (higher salience, more important). Default value is 0.	**Data Type:** String

Supported Response Codes

Table 3-3 Supported Response Codes

Response Code	Description
201	AMF Set configured successfully
200	Successful response
204	No Content
400	Bad Request
403	Forbidden
409	Constraint Violation
406	Not Acceptable client error
404	Config not found

Examples

The following example shows how an AMF Set is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/amfset -H 'Content-Type: application/json' -d '{ "regionId":
"01","setId":
      "101","plmnId": { "mcc": "100","mnc": "101" }, "salience": 1
    }'
```

Example of the Request Body
The following snippet shows an example of the request body:

```
{
    "regionId": "01",
    "setId": "101",
    "plmnId": {
        "mcc": "100",
        "mnc": "101"
    },
    "salience": 1
}
```

Example of the Response Body
The following example shows the contents of the response body in JSON format:

```
{
  "regionId": "01",
  "setId": "101",
  "plmnId": {
    "mcc": "100",
    "mnc": "101"
  },
  "salience": 1,
  "mccmncRegionidSetId": "100-101-01-101"
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/amfset/{setId}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/amfset/100-101-01-101"
```

3.2 Nss Rules

Nss Rules is a mandatory Managed Object. It maps SNSSAI+TAI+AMF-ID to the Network Slice Instance. If TOD is enabled, it adds in slice selection based on TOD. NSS Rules are policy rules which allow the operator to ALLOW or REJECT a request for a Network Slice Selection request, and if allowed, then map to a Network Slice.

- Request_Type: POST
- URL: *http://{apiRoot}/nnssf-configuration/v1/nssrules*
- REST message sample - Nss Rules
- Content-Type: application/json
- Initiated By: API Invoker

Nss Rules - Supported REST APIs

The operator can configure Nss Rules by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the Nss Rules Managed Object:

Table 3-4 Nss Rules - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/nssrules	Get all NssRules.	**Data Type:** array(NssRule) **HTTP Method:** GET
/nnssf-configuration/v1/nssrules	Create a network slice selection rule	**Data Type:** NssRule **HTTP Method:** POST
/nnssf-configuration/v1/nssrules/{name}	Read a network slice selection rule	**Data Type:** NssRule **HTTP Method:** GET
/nnssf-configuration/v1/nssrules/{name}	Delete a network slice selection rule	**Data Type:** NssRule **HTTP Method:** DELETE
/nnssf-configuration/v1/nssrules/{name}	Update a network slice selection rule	**Data Type:** NssRule **HTTP Method:** PUT

Nss Rules - Dependencies

To configure Nss Rules Managed Object, configure the following Managed Objects as a pre-requisite:

- **Nssai Auth:** An Auth with Grant "Allowed" encompassing range of TACs, default for PLMN, or a single matching TAC with rule is mandatory.
- **Nsi Profile:** Profile name being used in rule must be configured.
- **Time Profile:** If the rule is to return different profiles based on time of the day, then configuring the time profile is mandatory.

Request or Response Body Parameters

The Nss Rules Managed Object enables the configuration of policy rules. It allows an operator to allow, reject, or associate a Network slice based on NSSAI (SST and SD), PLMN (MCC and MNC), TAC, and AMF_ID. The operator can configure salience value to prioritize one rule over other.

Table 3-5 Nss Rules - Parameters

Parameter	Description	Details
name	**M/C/O:** This is a mandatory parameter. It contains Network Slice Selection Rule Name.	**Data Type:** String
amfId	**M/C/O:** This is an optional parameter. It contains AMF Identifier.	**Data Type:** String
plmnId	**M/C/O:** This is a mandatory parameter. It contains Public Land Mobile Network ID (MCC:MNC).	**Data Type:** String
tac	**M/C/O:** This is a conditional parameter. It contains Tracking Area Code (TAC).	**Data Type:** String
tacrange	**M/C/O:** This is a conditional parameter. Either tac or tacrange would be present. If both are not present, or if tac is null, then the rule corresponds to the PLMN.	**Data Type:** TacRange
snssai	**M/C/O:** This is a mandatory parameter. It contains Single Network Slice Selection Assistance Information.	**Data Type:** Snssai
salience	**M/C/O:** This is an optional parameter. It contains the order of importance (higher salience, more important). Note: salience should be greater that 5.	**Data Type:** Integer
behavior	**M/C/O:** This is a mandatory parameter. It contains the behavior of the parameter. To know more, see "Primitive Tables" in *Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide*.	**Data Type:** String

Supported Response Codes

Table 3-6 Supported Response Codes

Response Code	Description
201	NSS Rules configured successfully
200	Successful response
204	No Content
400	Bad Request
403	Forbidden
409	Constraint Violation
406	Not Acceptable client error
404	Config not found

Examples

The following example shows how an NSS Rules is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/nssrules -H 'Content-Type: application/json' -d '{ "name":
```

```
"TACRANGE-RULE-1", "amfId": "12345678-abcd-efAB-CDEF-123456789012", "plmnId":
{ "mcc": "100", "mnc": "101" }, "tac": null, "tacrange": { "starttac":
"100000", "endtac": "200000" }, "snssai": { "sst": "1", "sd": "EABB01" },
"salience": "6", "behavior": { "accessType": "3GPP_ACCESS", "nsiProfiles":
[ { "name": "NSI-PROFILE-1", "salience": 6 } ] } }'
```

Example of the Request Body

The following snippet shows an example of the request body:

```
{
    "name": "TACRANGE-RULE-1",
    "amfId": "12345678-abcd-efAB-CDEF-123456789012",
    "plmnId": {
        "mcc": "100",
        "mnc": "101"
    },
    "tac": null,
    "tacrange": {
        "starttac": "100000",
        "endtac": "200000"
    },
    "snssai": {
        "sst": "1",
        "sd": "EABB01"
    },
    "salience": "6",
    "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
            {
                "name": "NSI-PROFILE-1",
                "salience": 6
            }
        ]
    }
}
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
{
    "name": "TACRANGE-RULE-1",
    "amfId": "12345678-abcd-efAB-CDEF-123456789012",
    "plmnId": {
        "mcc": "100",
        "mnc": "101"
    },
    "tacrange": {
        "starttac": "100000",
        "endtac": "200000"
    },
    "snssai": {
        "sst": 1,
        "sd": "EABB01"
    },
```

```
    "salience": 6,
    "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
            {
                "name": "NSI-PROFILE-1",
                "salience": 6
            }
        ]
    }
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/nssrules/{name}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/nssrules/TACRANGE-RULE-1"
```

3.3 Time Profiles

Time profile is a conditional configuration only configured if TOD is enabled. This MO consists of time spans used to create a separate policy based on the time of day.

- Request_Type: POST
- URL: *http://host:port/nnssf-configuration/v1/timeprofiles*
- REST message sample - Time Profiles
- Content-Type: application/json
- Initiated By: API Invoker

Time Profile - Supported REST APIs

The operator can configure Time Profiles by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the Time Profile Managed Object.

Table 3-7 Time Profile - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/timeprofiles	Create a time profile	**Data Type:** TimeProfile **HTTP Method:** POST
/nnssf-configuration/v1/timeprofiles	Read all time profiles	**Data Type:** array(TimeProfile) **HTTP Method:** GET
/nnssf-configuration/v1/timeprofiles/{name}	Read a time profile by name	**Data Type:** TimeProfile **HTTP Method:** GET

Table 3-7 (Cont.) Time Profile - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/timeprofiles/{name}	Delete a time profile by name	**Data Type:** TimeProfile **HTTP Method:** DELETE
/nnssf-configuration/v1/timeprofiles/{name}	Update a time profile by name. If Time profile with name not found, return 404 Immutable Param: name Mutable Param: Start date, End date, Days of week, timespan(s)	**Data Type:** TimeProfile **HTTP Method:** PUT

Time Profile - Dependencies

This Managed Object is not dependent on other Managed Objects. Hence, configuration of any other Managed Objects is not a pre-requisite to configuring Time Profile.

> **Note**
>
> It is applicable only when "Time of the Day Based Network Slice Instance Selection" feature is enabled. For for information, see **Time of the Day Based Network Slice Instance Selection** section in *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

Request or Response Body Parameters

The Time Profile Managed Object enables the configuration of time or date based slice selection policies. This allows an operator to create a Time Profile and associate it to a network slice when creating a NSS Rule Managed Object.

Table 3-8 Time Profile - Parameters

Parameter	Description	Details
`name`	This is a mandatory parameter. Time Profile Name.	**Data Type:** String
`startDate`	This is a mandatory parameter. Date in the format of yy-mm-dd.	**Data Type:** Date
`endDate`	This is a mandatory parameter. Date in the format of yy-mm-dd.	**Data Type:** Date
`daysOfWeek`	This is an optional parameter. See Enumerations section.	**Data Type:** array(Daysofweek)
`timespans`	This is an optional parameter. See Primitive Tables section.	**Data Type:** array(TimeSpan)

Supported Response Codes

Table 3-9 Supported Response Codes

Response Code	Description
201	Time Profile configured successfully
200	Successful response
204	No Content
400	Bad Request
404	Config not found
406	Not Acceptable client error
500	Internal Server Error

Examples

The following example shows how an Time Profile is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/timeprofiles -H 'Content-Type: application/json' -d
'{ "name": "WEEKDAY-BUSY", "startDate": "2021-09-10", "endDate":
"2025-12-31", "daysOfWeek": [ "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY",
"FRIDAY" ], "timeSpans": [ { "startTime": "07:00:00", "endTime":
"09:00:00" }, { "startTime": "17:00:00", "endTime": "22:00:00" } ] }'
```

Example of the Request Body
The following snippet shows an example of the request body:

```
{
    "name": "WEEKDAY-BUSY",
    "startDate": "2021-09-10",
    "endDate": "2025-12-31",
    "daysOfWeek":
    [
        "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY"
    ],
    "timeSpans":
    [
        {
            "startTime": "07:00:00",
            "endTime": "09:00:00"
        },
        {
            "startTime": "17:00:00",
            "endTime": "22:00:00"
        }
    ]
}
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
{
    "name": "WEEKDAY-BUSY",
    "startDate": "2021-09-10",
    "endDate": "2025-12-31",
    "daysOfWeek": [
        "MONDAY",
        "TUESDAY",
        "WEDNESDAY",
        "THURSDAY",
        "FRIDAY"
    ],
    "timeSpans": [
        {
            "startTime": "07:00:00",
            "endTime": "09:00:00"
        },
        {
            "startTime": "17:00:00",
            "endTime": "22:00:00"
        }
    ]
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/timeprofiles/{name}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/timeprofiles/WEEKDAY-BUSY"
```

3.4 Configured NSSAI

Configured NSSAI is a mandatory Managed Object. It signifies default configured NSSAI per PLMN. NSSF enables operator to configure default SNSSAI per TAC also.

> **Note**
>
> Default Configured NSSAI signifies the Configured NSSAI for the PLMN by the operator. The Configured NSSAI that comes in the request is the UE's perspective of the same. The response from the NSSF is the intersection. The word "Default" is used to differentiate the perspectives (Operator and UE). This managed object signifies the Configured NSSAI for the PLMN, irrespective of the UE.

- Request_Type: POST
- URL:
- *http://{apiRoot}/nnssf-configuration/v1/configuredsnssais*

- REST message sample - Configured S-NSSAIs
- Content-Type: application/json
- Initiated by: API Invoker

Configured NSSAIs - Supported REST APIs

The operator can configure Configured NSSAIs by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the Configured NSSAIs Managed Object:

Table 3-10 Configured NSSAIs - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/configuredsnssais	Create a configured NSSAI.	**Data Type:** array(ConfiguredNssai) **HTTP Method:** POST
/nnssf-configuration/v1/configuredsnssais	Read all configured NSSAIs.	**Data Type:** array(ConfiguredNssai) **HTTP Method:** GET
/nnssf-configuration/v1/configuredsnssais/{mcc}-{mnc}	If combination of AmfId, MCC, MNC, TAC is not found, then responds with 404 Mutable parameters: Snssai List, Salience Immutable parameters: AMF-Id , MCC, MNC, TAC	**Data Type:** ConfiguredNssai **HTTP Method:** PUT
/nnssf-configuration/v1/configuredsnssais/{mcc}-{mnc}	Per AMF + TAI configured NSSAI Per TAI configured NSSAI Per PLMN configured NSSAI	**Data Type:** ConfiguredNssai ConfiguredNssai array(ConfiguredNssai) **HTTP Method:** GET
/nnssf-configuration/v1/configuredsnssais/{mcc}-{mnc}	Delete a configured NSSAI.	**Data Type:** ConfiguredNssai **HTTP Method:** DELETE
/nnssf-configuration/v1/configuredsnssais/{mcc}-{mnc}-{sst}-{sd}	Delete a configured NSSAI.	**Data Type:** ConfiguredNssai **HTTP Method:** DELETE

ConfiguredNssai - Dependencies

To configure **ConfiguredNssai** Managed Object, you need to first configure the following Managed Objects as a prerequisite:

- **NssaiAuth:** A default PLMN level Auth with grant ALLOWED.
- **NSSRule:** A default PLMN level Rule with AMF_ID as null (No dependency on AMF).

> **Note**
>
> For more information on default configuration, see Examples and steps for default Configured NSSAIs .

Request or Response Body Parameters

The Configured NSSAIs Managed Object enables the configuration of default NSSAI based on one or more of the following parameters: PLMN, TAC, and AMF-ID. The operator can configure default behavior when none of the rules match and UE has set default indication flag to true.

Table 3-11 Configured NSSAIs - Parameters

Parameter	Description	Details
`plmnid`	This is a mandatory parameter. Public Land Mobile Network ID (MCC:MNC).	**Data Type:** plmnid
`nssai`	This is a mandatory parameter. See Primitive Table section.	**Data Type:** array(Snssai)

Supported Response Codes

Table 3-12 Supported Response Codes

Response Code	Description
201	ConfiguredNssai configured successfully
200	Successful response
204	No Content
400	Bad Request
403	Forbidden
404	Not Found
406	Not Acceptable client error
500	Internal Server Error

Examples

The following example shows how an Configured NSSAI is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/configuredsnssais -H 'Content-Type: application/json' -d
'{ "plmn": { "mcc": "100", "mnc": "101" }, "nssai": [ { "sst": 1, "sd":
"EABB01" } ] }'
```

Example of the Request Body

The following snippet shows an example of the request body:

```
{
    "plmn": {
        "mcc": "100",
```

```
        "mnc": "101"
    },
    "nssai": [
        {
            "sst": 1,
            "sd": "EABB01"
        }
    ]
}
```

Example of the Response Body
The following example shows the contents of the response body in JSON format:

```
{
  "plmn": {
    "mcc": "100",
    "mnc": "101"
  },
  "nssai": [
    {
      "sst": 1,
      "sd": "EABB01"
    }
  ],
  "plmnId": "100-101"
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/configuredsnssais/{mcc}-{mnc}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/configuredsnssais/100-101"
```

Examples and steps for default Configured NSSAIs

Scenario: This is a sample scenario where we are confiuring S-NSSAI (1,EABB01) as default configured NSSAI. Perform the following prerequisite configurations for this scenario:

1. **Configure Auth**: It signifies that S-NSSAI (1,EABB01) is allowed on the whole PLMN. Sample cURL for configuring Auth:

   ```
   curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
   configuration/v1/nssaiauth -H 'Content-Type: application/json' -d '{
     "name": "NSSAI-AUTH-1",
     "plmnId": { "mcc": "100", "mnc": "101" },
     "snssai": { "sst": "1", "sd": "EABB01" },
     "grant": "ALLOWED"
   }'
   ```

2. **Configure Rule**: It signifies that S-NSSAI (1,EABB01) is allowed as per Auth, and also with that linked Auth Profile, NSI-PROFILE-1 is mapped.
   Sample cURL for configuring Rule:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/nssrules -H 'Content-Type: application/json' -d '{
  "name": "IR-RULE-1",
  "plmnId": { "mcc": "100", "mnc": "101" },
  "snssai": { "sst": 1, "sd": "EABB01" },
  "salience": "6",
  "behavior": {
    "accessType": "3GPP_ACCESS",
    "nsiProfiles": [{ "name": "NSI-PROFILE-1" }]
  }
}'
```

3. **Configure Configured NSSAI**: Now, we can have S-NSSAI (1, EABB01) in PLMN ("mcc": "100", "mnc": "101") set as the default. This configuration is because there is an Auth, S-NSSAI (1, EABB01), in PLMN ("mcc": "100", "mnc": "101"), and a rule that maps the Auth to a profile.
   Sample cURL to configure Configured NSSAI:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/configuredsnssais -H 'Content-Type: application/json' -d
'{
  "plmn": { "mcc": "100", "mnc": "101" },
  "nssai": [ { "sst": 1, "sd": "EABB01" } ]
}'
```

> **Note**
>
> - If more S-NSSAIs need to be part of Configured NSSAI, then a similar Auth and Rule pair must be configured for each S-NSSAI.
> - These steps assume that the NSI-PROFILE-1 profile already exists.

3.5 NSSAI Auth

NSSAI Auth is a mandatory Managed Object. It consists of network slice, TAI and Grant value. The configuration signifies a mapping of allowed and restricted S-NSSAI per TAI.

- Request_Type: POST
- URL: *http://localhost:5755/nnssf-configuration/v1/nssaiauth*
- REST message sample - NSSAI Auth
- Initiated by: API Invoker

NSSAI Auth - Supported REST APIs

An operator can configure NSSAI Auth by following the information provided in the table below. The supported operations are **POST**, **GET**, and **DELETE**. The following table provides information about the REST APIs supported by the NSSAI Auth Managed Object.

Table 3-13 NSSAI Auth - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/nssaiauth	Create an NSSAIAuth Profile.	**Data Type:** NssaiAuth **HTTP Method:** POST
nnssf-configuration/v1/nssaiauth	Return all configured NssaiAuth Managed Objects	**Data Type:** array(NssaiAuth) **HTTP Method:** GET
nnssf-configuration/v1/nssaiauth/{name}	Returns an NssaiAuth Managed Object if present with the name	**Data Type:** NssaiAuth **HTTP Method:** GET
nnssf-configuration/v1/nssaiauth/{name}	Updates an NssaiAuth Managed Object if present with the name	**Data Type:** NssaiAuth **HTTP Method:** PUT
/nnssf-configuration/v1/nssaiauth/{name}	Delete an NssaiAuth Managed Object if a Managed Object with name is found.	**Data Type:** NssaiAuth **HTTP Method:** DELETE

NSSAI Auth - Dependencies

This Managed Object is not dependent on other Managed Objects. Hence, configuration of any other Managed Objects is not a prerequisite to configure NSSAI Auth.

Request or Response Body Parameters

The NSSAI Auth Managed Object enables the configuration of network slice authentication rules by configuring Grant status (Allowed_PLMN, Rejected_PLMN, or Rejected_TAC) for S-NSSAI on a per TAI basis.

Table 3-14 NSSAI Auth - Parameters

Parameter	Description	Details
`name`	This is a mandatory parameter. Network Slice Authentication Rule Name.	**Data Type:** String
`plmnId`	This is a mandatory parameter. Public Land Mobile Network ID (MCC:MNC).	**Data Type:** plmnid
`tac`	This is a conditional parameter. Tracking Area Code.	**Data Type:** String
`tacrange`	This is a conditional parameter. Range of TAC represented by `starttac` and `endtac`. Either `tac` or `tacrange` would be present. If both are not present, then Auth corresponds to PLMN.	**Data Type:** TacRange
`snssai`	This is a mandatory parameter. Single Network Slice Selection Assistance Information.	**Data Type:** Snssai
`grant`	This is a mandatory parameter. Whether the requested s-NSSAI is allowed or restricted.	**Data Type:** Grant

Supported Response Codes

Table 3-15 Supported Response Codes

Response Code	Description
201	NSSAI Auth configured successfully
200	Successful response
204	No Content
400	Bad Request
403	Forbidden
409	Constraint Violation
406	Not Acceptable client error
422	Unprocessable entity
404	Config not found
500	Internal Server Error

Examples

The following example shows how an NSSAI Auth is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/nssaiauth -H 'Content-Type: application/json' -d '{ "name":
"NSSAI-AUTH-1", "plmnId": { "mcc": "100", "mnc": "101" }, "tac": null,
"tacrange": { "starttac": "100000", "endtac": "200000" }, "snssai":
{ "sst":"1", "sd":"EABB01" }, "grant": "ALLOWED" }'
```

Example of the Request Body
The following snippet shows an example of the request body:

```
{
    "name": "NSSAI-AUTH-1",
    "plmnId":
    {
        "mcc": "100",
        "mnc": "101"
    },
    "tac": null,
    "tacrange":
    {
        "starttac": "100000",
        "endtac": "200000"
    },
    "snssai":
    {
        "sst":"1",
        "sd":"EABB01"
    },
    "grant": "ALLOWED"
}
```

Example of the Response Body
The following example shows the contents of the response body in JSON format:

```
{
    "name": "NSSAI-AUTH-1",
    "plmnId":
    {
        "mcc": "100",
        "mnc": "101"
    },
    "tacrange":
    {
        "starttac": "100000",
        "endtac": "200000"
    },
    "snssai":
    {
        "sst":"1",
        "sd":"EABB01"
    },
    "grant": "ALLOWED"
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/nssaiauth/{name}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/nssaiauth/NSSAI-AUTH-1"
```

3.6 AMF Resolutions

AMF Resolutions is a conditional Managed Object, only configured if subscription to NRF helm parameter (nsconfig.nrf.subscription) is not set. AMF Resolutions enables operator to configure mapping candidate AMF list to a Target AMF set ID and Region ID. If a subscription to NRF is set, this MO is auto-populated.

- Request_Type: POST
- URL: *http: //{apiRoot}/nnssf-configuration/v1/amfresolutions*
- REST message sample - AMF Resolutions
- Content-Type: application/json
- Initiated By: API Invoker

AMF Resolutions - Supported REST APIs

An operator can configure AMF Resolutions by following the information provided in the table below. The supported operations are **POST** and **GET**. The following table provides information about the REST APIs supported by the AMF Resolutions Managed Object:

Table 3-16 AMF Resolutions - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/amfresolutions	Get all AMFs for all set-ids	**Data Type:** array(AmfResolution) **HTTP Method:** GET
/nnssf-configuration/v1/amfresolutions	Create a AMF Resolutions	**Data Type:** AmfResolution **HTTP Method:** POST
/nnssf-configuration/v1/amfresolutions/{mcc}-{mnc}-{sst}-{sd}	Get all AMFs inside the set-id	**Data Type:** AmfResolution **HTTP Method:** GET
/nnssf-configuration/v1/amfresolutions/{mcc}-{mnc}-{sst}-{sd}	Updates the `candidateAmfList` for the AMF with set-id{mcc}-{mnc}-{sst}-{sd}. **Note**: `candidateAmfList` should not be empty. An AMF set must contain at least one candidate AMF.	**Data Type:** AmfResolution **HTTP Method:** PUT

AMF Resolutions -Dependencies

This Managed Object is not dependent on other Managed Objects. Hence, configuration of any other Managed Objects is not a prerequisite to configure AMF Resolution.

Request or Response Body Parameters

The AMF Resolutions Managed Object enables the configuration of mapping list of candidate AMFs to a pair of Target AMF set ID and Region ID. The operator can provide a static candidate AMF list. This configuration is used in cases where customer has disabled discovery service with NRF.

Table 3-17 AMF Resolutions - Parameters

Attribute	Description	Details
`regionId`	This is a mandatory parameter. Region ID of the target AMF list.	**Data Type:** Integer
`setId`	This is a mandatory parameter. Set ID of the target AMF list.	**Data Type:** Integer
`mcc`	This is a mandatory parameter. Mobile Country Code.	**Data Type:** String
`mnc`	This is a mandatory parameter. Mobile Network Code.	**Data Type:** String
`candidateAmfList`	This is a conditional parameter. For more information, see in the Primitive Table section . **Note:** This parameter is mandatory, non empty, and valid when NRF subscription is not enabled. When NRF subscription is enabled, this parameter is ignored.	**Data Type:** Array(candidateAmf)

> **⚠ Caution**
>
> An AMF set must contain at least one AMF, and this requirement must be maintained during configuration. The NSSF does not allow the operator to delete all AMFs within an AMF set, as an empty AMF set has no operational value. Therefore, if the operator intends to remove all AMFs from an AMF set, it is recommended to delete the entire AMF set instead.

Supported Response Codes

Table 3-18 Supported Response Codes

Response Code	Description
201	AMF Resolutions configured successfully
200	Successful response
204	No Content
400	Bad Request
404	Not Found
406	Not Acceptable client error
500	Internal Server Error

Examples

The following example shows how an AMF Resolution is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/amfresolutions -H 'Content-Type: application/json' -d
'{ "regionId": "01", "setId": "102", "mcc":"100", "mnc":"101",
"candidateAmfList": [ { "instanceId": "12345678-abcd-efAB-
CDEF-123456789012" }, { "instanceId": "12345678-abcd-efAB-
CDEF-123456789013" } ] } }'
```

Example of the Request Body
The following snippet shows an example of the request body:

```
{
    "regionId": "01",
    "setId": "102",
    "mcc":"100",
    "mnc":"101",
    "candidateAmfList":
    [
        {
            "instanceId": "12345678-abcd-efAB-CDEF-123456789012"
        },
        {
            "instanceId": "12345678-abcd-efAB-CDEF-123456789013"
        }
    ]
```

```
}
}
```

Example of the Response Body
The following example shows the contents of the response body in JSON format:

```
{
    "regionId": "01",
    "setId": "102",
    "mcc":"100",
    "mnc":"101",
    "candidateAmfList":
    [
        {
            "instanceId": "12345678-abcd-efAB-CDEF-123456789012"
        },
        {
            "instanceId": "12345678-abcd-efAB-CDEF-123456789013"
        }
    ]
}
```

3.7 NSI Profiles

NSI Profiles is a mandatory Managed Object. It consists of network slice name and ID and NRF-ID, Target AMF lists associated with the slice. This Managed Object contains network slice information.

> **Note**
>
> If Target AMF sets associated with network slice is not present then NSSF adds AMF set with salience 0.

- Request_Type: POST
- URL: *http://{apiRoot}/nnssf-configuration/v1/nsiprofiles*
- REST message sample - NSI Profiles
- Content-Type: application/json
- Initiated By: API Invoker

NSI Profiles - Supported REST APIs

Customer can configure NSI Profiles by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the NSI Profiles Managed Object:

Table 3-19 NSI Profiles - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/nsiprofiles	Returns array of all configured NsiProfiles.	**Data Type:** array(NsiProfile) **HTTP Method:** GET
/nnssf-configuration/v1/nsiprofiles	Create a network slice instance profile.	**Data Type:** NsiProfile **HTTP Method:** POST
/nnssf-configuration/v1/nsiprofiles/{name}	Read a network slice instance profile. Returns NsiProfile based on name	**Data Type:** NsiProfile **HTTP Method:** GET
/nnssf-configuration/v1/nsiprofiles/{name}	Delete a network slice instance profile based on name.	**Data Type:** NsiProfile **HTTP Method:** DELETE
/nnssf-configuration/v1/nsiprofiles/{name}	Update a network slice instance profile based on name. Immutable parameter is `nsiId`. Rest all parameters, except name, are mutable. It performs a Delete and POST only for targetamfsets.	**Data Type:** NsiProfile **HTTP Method:** PUT

NSI Profiles - Dependencies

This Managed Object is not dependent on other Managed Objects. Hence, configuration of any other Managed Objects is not a prerequisite to configure NSI Profile.

Request or Response Body Parameters

The NSI Profiles Managed Object enables the configuration of Network Slice Instance profile. The operator can create a Network Slice by providing a name, id, NRF URL corresponding to the slice, and a list of Target AMF sets that support this slice.

Table 3-20 NSI Profiles - Parameters

Parameter	Description	Details
`name`	This is a mandatory parameter. Contains Network Slice Instance Profile Name.	**Data Type:** String
`nrfUri`	This is a mandatory parameter. Contains URI of the Network Repository Function. **Note:** It must contain NRF NF Discovery URL.	**Data Type:** String
`nsiId`	This is an optional parameter. Contains Network Slice Instance Identifier.	**Data Type:** String
`targetAmfSets`	This is a mandatory parameter. Contains an array of `TargetAmfSet` (see Primitive Tables section).	**Data Type:** Array (TargetAmfSet)
`nrfNfMgtUri`	This is a mandatory parameter. Contains Management URI of Network Resource Function.	**Data Type:** String
`nrfAccessTokenUri`	This is a mandatory parameter. Contains Access Token URI of Network Resource Function.	**Data Type:** String
`plmn`	This is a mandatory parameter. Contains PLMN ID.	**Data Type:** plmn

Supported Response Codes

Table 3-21 Supported Response Codes

Response Code	Description
201	NSI Profiles configured successfully
400	Bad Request
403	Forbidden
404	Config not found
406	Not Acceptable client error
500	Internal Server Error

Examples

The following example shows how an NSI Profiles is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/nsiprofiles -H 'Content-Type: application/json' -d
'{ "name": "NSI-PROFILE-1", "nrfUri": "https://nrf.slice1.oracle.com/nnrf-
disc/v1", "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
"nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token", "nsiId":
"SLICE1", "plmn": { "mcc": "100", "mnc": "101" }, "targetAmfSets":
[ { "regionId": "01", "setId": "101", "salience": "1" }, { "regionId": "01",
"setId": "102", "salience": "2" } ] }'
```

Example of the Request Body
The following snippet shows an example of the request body:

```
{
    "name": "NSI-PROFILE-1",
    "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
    "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
    "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
    "nsiId": "SLICE1",
    "plmn": {
        "mcc": "100",
        "mnc": "101"
    },
    "targetAmfSets": [
        {
            "regionId": "01",
            "setId": "101",
            "salience": "1"
        },
        {
            "regionId": "01",
            "setId": "102",
            "salience": "2"
        }
```

```
    ]
}
```

Example of the Response Body
The following example shows the contents of the response body in JSON format:

```
{
    "name": "NSI-PROFILE-1",
    "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
    "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
    "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
    "nsiId": "SLICE1",
    "plmn": {
        "mcc": "100",
        "mnc": "101"
    },
    "targetAmfSets": [
        {
            "regionId": "01",
            "setId": "101",
            "salience": 1
        },
        {
            "regionId": "01",
            "setId": "102",
            "salience": 2
        }
    ]
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/nsiprofiles/{name}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/nsiprofiles/NSI-PROFILE-1"
```

3.8 GrSite

GrSite is a conditional Managed Object. It is relevant only if georedundancy (`global.grEnabled`) is enabled. GrSite enables the customer to configure GrSite information.

- Request_Type: POST/GET /PUT
- URL: *http: //{apiRoot}/nnssf-configuration/v1/grsites*
- REST message sample - GrSite
- Content-Type: application/json
- Initiated By: API Invoker

GrSite - Supported REST APIs

Configure the GrSite by using the information provided in the table below. The supported operations are **POST**, **GET**, and **PUT**. The following table provides information about the REST APIs supported by the GrSite Managed Object:

Table 3-22 GrSite - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/grsites	It returns array of all configured GrSites.	**Data Type:** Array(GrSite) **HTTP Method:** GET
/nnssf-configuration/v1/grsites/{NfId}	It returns GrSite for corresponding nfId.	**Data Type:** Array(GrSite) **HTTP Method:** GET
/nnssf-configuration/v1/grsites	It creates a GrSite.	**Data Type:** GrSite **HTTP Method:** POST
/nnssf-configuration/v1/grsites/{nfId}	It updates the rank of GrSite.	**Data Type:** Integer(GrSite) **HTTP Method:** PUT
/nnssf-configuration/v1/grsites/{NfId}	It deletes GrSite for corresponding nfId.	**Data Type:** GrSite **HTTP Method:** DELETE

GrSite - Dependencies

This Managed Object is not dependent on other Managed Objects. Hence, configuration of any other Managed Objects is not a prerequisite to configure GrSite.

> **Note**
>
> It is applicable only when Georedundency is enabled in the helm options. For for information about enabling Georedundency, see **Georedundency** section in *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

Request or Response Body Parameters

The GrSite Managed Object enables the configuration of different sites for the NSSF. This Managed Object allows an operator to configure various sites with different priorities, ranks, and statuses.

> **Note**
>
> Currently, NSSF supports three-site georedundency.

Table 3-23 GrSite - Parameters

Parameter	Description	Details
nfId	This is a mandatory parameter. Contains the instance ID of the site NSSF.	**Data Type:** String
rank	This is a mandatory parameter. Contains the priority given by the operator to related georedundant sites. **Note**: A lower value indicates a higher priority.	**Data Type:** Integer
grSiteStatus	This is a mandatory parameter. It contains the current status of the site or NSSF. **Note:** To know more about data type, see enumerations section.	**Range:** ACTIVE or DOWN **Data Type:** enum

Supported Response Codes

Table 3-24 Supported Response Codes

Response Code	Description
201	GrSite configured successfully
200	Successful response
204	No Content
400	Bad Request
406	Not Acceptable client error
500	Internal Server Error

Examples

The following example shows how a GrSite is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"1", "nfId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1",
"grSiteStatus":"ACTIVE" }'
```

Example of the Request Body
The following snippet shows an example of the request body:

```
{
    "rank": "1",
    "nfId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1",
    "grSiteStatus":"ACTIVE"
}
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
{
    "rank": "1",
    "nfId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1",
    "grSiteStatus":"ACTIVE"
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-configuration/v1/grsites/{NfId}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-configuration/v1/grsites/fe7d992b-0541-4c7d-ab84-c6d70b1b01b1"
```

3.9 NSSF Backup and Restore Configurations

These APIs provide functionality to back up existing configuration, delete existing configuration, and to restore a (stored) back up.

- Request_Type: POST/GET/DELETE
- URL: *http://{apiRoot}/nnssf-configuration/v1/upload*
- Content-Type: application/json
- Initiated By: API Invoker

NSSF Backup and Restore Configurations - Supported REST APIs

Configure the ConfigurationBackup Managed Object by following the information provided in the table below. The supported operations are **POST**, **GET**, and **DELETE**. Perform the configuration of ConfigurationBackup Managed Object as follows:

Table 3-25 NSSF Backup and Restore Configurations - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/upload	Restores a backup configuration.	**Data Type:** ConfigBackup **HTTP Method:** POST
/nnssf-configuration/v1/backup	Returns backup of existing configuration.	**Data Type:** ConfigBackup **HTTP Method:** GET
/nnssf-configuration/v1/deleteConfiguration/	Deletes existing configuration.	**HTTP Method:** DEL

Request or Response Body Parameters

ConfigurationBackup consists of list of following parameters:

Table 3-26 NSSF Backup and Restore Configurations - Parameters

Parameter	Description	Details
plmnLevelNsiProfileDto	If PlmnLevelNsiProfile is configured, this parameter contains list of PlmnLevelNsiProfiles.	**Data Type**: Array(PLMNLevelNsiProfileDto)
nsiProfileDto	If NsiProfile is configured, this parameter contains list of NsiProfiles.	**Data Type:** Array(NsiProfileDto)
nssaiAuthDto	If NssaiAuth is configured, this parameter contains list of NssaiAuths.	**Data Type:** Array(NsiProfileDto)
timeProfileDto	If TimeProfile is configured, this parameter contains list of TimeProfile.	**Data Type:** Array(TimeProfileDto)
nssRuleDto	If NssRule is configured, this parameter contains list of NssRule.	**Data Type:** Array(NssRuleDto)
amfResolutionDto	If AmfResolution is configured, this parameter contains list of AmfResolution.	**Data Type:** Array(AmfResolutionDto)
configuredSnssaiDto	If ConfiguredSnssai is configured, this parameter contains list of ConfiguredSnssai.	**Data Type:** Array(ConfiguredSnssaiDto)
amfSetDto	If amfSetDto is configured, this parameter contains list of AmfSets.	**Data Type**: Array(AmfSetDto)
mappingOfNssaisDto	If mappingOfNssaisDto is configured, this parameter contains list of MappingOfNssais.	**Data Type**: Array(PlmnMappingOfNssai)
nssfSystemOptionDtoList	If nssfSystemOptionDtoList is configured, this parameter contains list of NssfSystemOptions.	**Data Type**: Array(NssfSystemOptionDto)
trustedAmfDtoList	If trustedAmfDtoList is configured, this parameter contains list of TrustedAmfs.	**Data Type**: Array(TrustedAmfDto)

Supported Response Codes

Table 3-27 Supported Response Codes

Response Code	Description
201	Configured successfully
400	Bad Request
404	Not Found
406	Not Acceptable client error
204	No Content

Examples

The following example shows how ConfigurationBackup is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/upload -H 'Content-Type: application/json' -d '
```

Example of the Request Body of Configuration Upload

The following snippet shows an example of the request body:

```
{
  "plmnLevelNsiProfileDto": [
    {
      "name": "NSI-PROFILE-3",
      "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
      "nsiId": "SLIC3",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "plmnId": "100-101"
    }
  ],
  "nsiProfileDto": [
    {
      "name": "NSI-PROFILE-1",
      "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
      "nsiId": "SLICE1",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "targetAmfSets": [
        {
          "regionId": "01",
          "setId": "101",
          "salience": 1
        },
        {
          "regionId": "01",
          "setId": "102",
          "salience": 2
        }
      ]
    },
    {
      "name": "NSI-PROFILE-2",
      "nrfUri": "https://nrf.slice2.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice2.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice2.oracle.com/oauth2/token",
      "nsiId": "SLICE2",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "targetAmfSets": [
        {
          "regionId": "01",
          "setId": "101",
          "salience": 1
```

```
        },
        {
          "regionId": "01",
          "setId": "102",
          "salience": 2
        },
        {
          "regionId": "01",
          "setId": "103",
          "salience": 3
        }
      ]
    }
  ],
  "nssaiAuthDto": [
    {
      "name": "NSSAI-AUTH-2",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 1,
        "sd": "EABB01"
      },
      "grant": "ALLOWED"
    },
    {
      "name": "NSSAI-AUTH-3",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 2,
        "sd": "EABB01"
      },
      "grant": "ALLOWED"
    },
    {
      "name": "NSSAI-AUTH-4",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 5,
        "sd": "EABB01"
      },
      "grant": "ALLOWED"
    }
  ],
  "timeProfileDto": [
    {
      "name": "WEEKDAY-BUSY",
      "startDate": "2021-09-10",
```

```
      "endDate": "2025-12-31",
      "daysOfWeek": [
        "MONDAY",
        "TUESDAY",
        "WEDNESDAY",
        "THURSDAY",
        "FRIDAY"
      ],
      "timeSpans": [
        {
          "startTime": "07:00:00",
          "endTime": "45:00:00"
        },
        {
          "startTime": "17:00:00",
          "endTime": "22:00:00"
        }
      ]
    }
  ],
  "nssRuleDto": [
    {
      "name": "TACRANGE-RULE-1",
      "amfId": "12345678-abcd-efAB-CDEF-123456789012",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "tacrange": {
        "starttac": "100000",
        "endtac": "200000"
      },
      "snssai": {
        "sst": 1,
        "sd": "EABB01"
      },
      "salience": 6,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-1",
            "salience": 6
          }
        ]
      }
    },
    {
      "name": "TACRANGE-RULE-2",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 2,
        "sd": "EABB01"
```

```
      },
      "salience": 5,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-2",
            "salience": 6
          }
        ]
      }
    },
    {
      "name": "TACRANGE-RULE-3",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 5,
        "sd": "EABB01"
      },
      "salience": 5,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-3",
            "salience": 6
          }
        ]
      }
    }
  ],
  "amfResolutionDto": [
    {
      "regionId": "01",
      "setId": "102",
      "mcc": "100",
      "mnc": "101",
      "mccmncRegionidSetId": "100-101-01-102",
      "candidateAmfList": [
        {
          "instanceId": "12345678-abcd-efAB-CDEF-123456789012"
        },
        {
          "instanceId": "12345678-abcd-efAB-CDEF-123456789013"
        }
      ]
    }
  ],
  "configuredSnssaiDto": [
    {
      "plmn": {
        "mcc": "100",
        "mnc": "101"
```

```
      },
      "nssai": [
        {
          "sst": 5,
          "sd": "EABB01"
        }
      ],
      "plmnId": "100-101"
    }
  ],
  "amfSetDto": [
    {
      "regionId": "01",
      "setId": "101",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "salience": 0,
      "mccmncRegionidSetId": "100-101-01-101"
    },
    {
      "regionId": "01",
      "setId": "102",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "salience": 0,
      "mccmncRegionidSetId": "100-101-01-102"
    },
    {
      "regionId": "01",
      "setId": "103",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "salience": 0,
      "mccmncRegionidSetId": "100-101-01-103"
    },
    {
      "regionId": "01",
      "setId": "104",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "salience": 1,
      "mccmncRegionidSetId": "100-101-01-104"
    }
  ],
  "mappingofNssaisDto": [
    {
      "plmnId": {
        "mcc": "100",
```

```
        "mnc": "101"
      },
      "mappingOfNssai": [
        {
          "servingSnssai": {
            "sst": 1,
            "sd": "EABB01"
          },
          "homeSnssai": {
            "sst": 7,
            "sd": "EABB07"
          }
        },
        {
          "servingSnssai": {
            "sst": 2,
            "sd": "EABB02"
          },
          "homeSnssai": {
            "sst": 8,
            "sd": "EABB08"
          }
        }
      ]
    }
  ],
  "nssfSystemOptionDtoList": [
    {
      "NssfSystemOptionService": [
        {
          "scope": "NsSelection",
          "PerPLMNConfiguration": [
            {
              "PlmnConfiguration": {
                "EnableEnhancedAllowedNSSAIComputation": true
              },
              "plmnId": {
                "mcc": "100",
                "mnc": "101"
              }
            }
          ]
        }
      ],
      "AutoConfigurationFromNsAvailability": "FromTrustedAMFs",
      "EnhancedPatchBehaviour": false
    }
  ],
  "trustedAmfDtoList": [
    {
      "amfSet": "100-101-01-101",
      "amfIdList": [
        "4faf1bbc-6e4a-4454-a507-aef01a101a01",
        "4faf1bbc-6e4a-4454-a507-aef01a101a02"
      ]
    },
```

```
    {
      "amfSet": "100-101-01-103",
      "amfIdList": [
        "4faf1bbc-6e4a-4454-a507-aef01a101a03",
        "4faf1bbc-6e4a-4454-a507-aef01a101a04"
      ]
    }
  ]
}
```

Example of the Response Body of Configuration Upload

The following example shows the contents of the response body in JSON format.

```
{
  "plmnLevelNsiProfileDto": [
    {
      "name": "NSI-PROFILE-3",
      "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
      "nsiId": "SLIC3",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "plmnId": "100-101"
    }
  ],
  "nsiProfileDto": [
    {
      "name": "NSI-PROFILE-1",
      "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
      "nsiId": "SLICE1",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "targetAmfSets": [
        {
          "regionId": "01",
          "setId": "101",
          "salience": 1
        },
        {
          "regionId": "01",
          "setId": "102",
          "salience": 2
        }
      ]
    },
    {
      "name": "NSI-PROFILE-2",
      "nrfUri": "https://nrf.slice2.oracle.com/nnrf-disc/v1",
```

```
      "nrfNfMgtUri": "https://nrf.slice2.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice2.oracle.com/oauth2/token",
      "nsiId": "SLICE2",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "targetAmfSets": [
        {
          "regionId": "01",
          "setId": "101",
          "salience": 1
        },
        {
          "regionId": "01",
          "setId": "102",
          "salience": 2
        },
        {
          "regionId": "01",
          "setId": "103",
          "salience": 3
        }
      ]
    }
  ],
  "nssaiAuthDto": [
    {
      "name": "NSSAI-AUTH-2",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 1,
        "sd": "EABB01"
      },
      "grant": "ALLOWED"
    },
    {
      "name": "NSSAI-AUTH-3",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 2,
        "sd": "EABB01"
      },
      "grant": "ALLOWED"
    },
    {
      "name": "NSSAI-AUTH-4",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
```

```
    },
    "snssai": {
      "sst": 5,
      "sd": "EABB01"
    },
    "grant": "ALLOWED"
  }
],
"timeProfileDto": [
  {
    "name": "WEEKDAY-BUSY",
    "startDate": "2021-09-10",
    "endDate": "2025-12-31",
    "daysOfWeek": [
      "MONDAY",
      "TUESDAY",
      "WEDNESDAY",
      "THURSDAY",
      "FRIDAY"
    ],
    "timeSpans": [
      {
        "startTime": "07:00:00",
        "endTime": "45:00:00"
      },
      {
        "startTime": "17:00:00",
        "endTime": "22:00:00"
      }
    ]
  }
],
"nssRuleDto": [
  {
    "name": "TACRANGE-RULE-1",
    "amfId": "12345678-abcd-efAB-CDEF-123456789012",
    "plmnId": {
      "mcc": "100",
      "mnc": "101"
    },
    "tacrange": {
      "starttac": "100000",
      "endtac": "200000"
    },
    "snssai": {
      "sst": 1,
      "sd": "EABB01"
    },
    "salience": 6,
    "behavior": {
      "accessType": "3GPP_ACCESS",
      "nsiProfiles": [
        {
          "name": "NSI-PROFILE-1",
          "salience": 6
        }
```

```
        ]
      }
    },
    {
      "name": "TACRANGE-RULE-2",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 2,
        "sd": "EABB01"
      },
      "salience": 5,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-2",
            "salience": 6
          }
        ]
      }
    },
    {
      "name": "TACRANGE-RULE-3",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 5,
        "sd": "EABB01"
      },
      "salience": 5,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-3",
            "salience": 6
          }
        ]
      }
    }
  ],
  "amfResolutionDto": [
    {
      "regionId": "01",
      "setId": "102",
      "mcc": "100",
      "mnc": "101",
      "mccmncRegionidSetId": "100-101-01-102",
      "candidateAmfList": [
        {
          "instanceId": "12345678-abcd-efAB-CDEF-123456789012"
```

```
        },
        {
          "instanceId": "12345678-abcd-efAB-CDEF-123456789013"
        }
      ]
    }
  ],
  "configuredSnssaiDto": [
    {
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "nssai": [
        {
          "sst": 5,
          "sd": "EABB01"
        }
      ],
      "plmnId": "100-101"
    }
  ],
  "amfSetDto": [
    {
      "regionId": "01",
      "setId": "104",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "salience": 1,
      "mccmncRegionidSetId": "100-101-01-104"
    }
  ],
  "mappingofNssaisDto": [
    {
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "mappingOfNssai": [
        {
          "servingSnssai": {
            "sst": 1,
            "sd": "EABB01"
          },
          "homeSnssai": {
            "sst": 7,
            "sd": "EABB07"
          }
        },
        {
          "servingSnssai": {
            "sst": 2,
            "sd": "EABB02"
          },
```

```
            "homeSnssai": {
              "sst": 8,
              "sd": "EABB08"
            }
          }
        ]
      }
    ],
    "nssfSystemOptionDtoList": [
      {
        "NssfSystemOptionService": [
          {
            "scope": "NsSelection",
            "PerPLMNConfiguration": [
              {
                "PlmnConfiguration": {
                  "EnableEnhancedAllowedNSSAIComputation": false
                },
                "plmnId": {
                  "mcc": "311",
                  "mnc": "480"
                }
              },
              {
                "PlmnConfiguration": {
                  "EnableEnhancedAllowedNSSAIComputation": true
                },
                "plmnId": {
                  "mcc": "100",
                  "mnc": "101"
                }
              }
            ]
          }
        ],
        "AutoConfigurationFromNsAvailability": "FromTrustedAMFs",
        "EnhancedPatchBehaviour": false
      }
    ],
    "trustedAmfDtoList": [
      {
        "amfSet": "100-101-01-101",
        "amfIdList": [
          "4faf1bbc-6e4a-4454-a507-aef01a101a01",
          "4faf1bbc-6e4a-4454-a507-aef01a101a02"
        ]
      },
      {
        "amfSet": "100-101-01-103",
        "amfIdList": [
          "4faf1bbc-6e4a-4454-a507-aef01a101a03",
          "4faf1bbc-6e4a-4454-a507-aef01a101a04"
        ]
      }
    ]
  }
```

Example of the Response Body of Configuation Backup

The following example shows the contents of the response body in JSON format:


```
{
  "plmnLevelNsiProfileDto": [
    {
      "name": "NSI-PROFILE-3",
      "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
      "nsiId": "SLIC3",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "plmnId": "100-101"
    }
  ],
  "nsiProfileDto": [
    {
      "name": "NSI-PROFILE-1",
      "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
      "nsiId": "SLICE1",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "targetAmfSets": [
        {
          "regionId": "01",
          "setId": "101",
          "salience": 1
        },
        {
          "regionId": "01",
          "setId": "102",
          "salience": 2
        }
      ]
    },
    {
      "name": "NSI-PROFILE-2",
      "nrfUri": "https://nrf.slice2.oracle.com/nnrf-disc/v1",
      "nrfNfMgtUri": "https://nrf.slice2.oracle.com/nnrf-nfm/v1",
      "nrfAccessTokenUri": "https://nrf.slice2.oracle.com/oauth2/token",
      "nsiId": "SLICE2",
      "plmn": {
        "mcc": "100",
        "mnc": "101"
      },
      "targetAmfSets": [
        {
          "regionId": "01",
          "setId": "101",
```

```
            "salience": 1
          },
          {
            "regionId": "01",
            "setId": "102",
            "salience": 2
          },
          {
            "regionId": "01",
            "setId": "103",
            "salience": 3
          }
        ]
      }
    ],
    "nssaiAuthDto": [
      {
        "name": "NSSAI-AUTH-2",
        "plmnId": {
          "mcc": "100",
          "mnc": "101"
        },
        "snssai": {
          "sst": 1,
          "sd": "EABB01"
        },
        "grant": "ALLOWED"
      },
      {
        "name": "NSSAI-AUTH-3",
        "plmnId": {
          "mcc": "100",
          "mnc": "101"
        },
        "snssai": {
          "sst": 2,
          "sd": "EABB01"
        },
        "grant": "ALLOWED"
      },
      {
        "name": "NSSAI-AUTH-4",
        "plmnId": {
          "mcc": "100",
          "mnc": "101"
        },
        "snssai": {
          "sst": 5,
          "sd": "EABB01"
        },
        "grant": "ALLOWED"
      }
    ],
    "timeProfileDto": [
      {
        "name": "WEEKDAY-BUSY",
```

```
      "startDate": "2021-09-10",
      "endDate": "2025-12-31",
      "daysOfWeek": [
        "MONDAY",
        "TUESDAY",
        "WEDNESDAY",
        "THURSDAY",
        "FRIDAY"
      ],
      "timeSpans": [
        {
          "startTime": "07:00:00",
          "endTime": "45:00:00"
        },
        {
          "startTime": "17:00:00",
          "endTime": "22:00:00"
        }
      ]
    }
  ],
  "nssRuleDto": [
    {
      "name": "TACRANGE-RULE-1",
      "amfId": "12345678-abcd-efAB-CDEF-123456789012",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "tacrange": {
        "starttac": "100000",
        "endtac": "200000"
      },
      "snssai": {
        "sst": 1,
        "sd": "EABB01"
      },
      "salience": 6,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-1",
            "salience": 6
          }
        ]
      }
    },
    {
      "name": "TACRANGE-RULE-2",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 2,
```

```
        "sd": "EABB01"
      },
      "salience": 5,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-2",
            "salience": 6
          }
        ]
      }
    },
    {
      "name": "TACRANGE-RULE-3",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "snssai": {
        "sst": 5,
        "sd": "EABB01"
      },
      "salience": 5,
      "behavior": {
        "accessType": "3GPP_ACCESS",
        "nsiProfiles": [
          {
            "name": "NSI-PROFILE-3",
            "salience": 6
          }
        ]
      }
    }
  ],
  "amfResolutionDto": [
    {
      "regionId": "01",
      "setId": "102",
      "mcc": "100",
      "mnc": "101",
      "mccmncRegionidSetId": "100-101-01-102",
      "candidateAmfList": [
        {
          "instanceId": "12345678-abcd-efAB-CDEF-123456789012"
        },
        {
          "instanceId": "12345678-abcd-efAB-CDEF-123456789013"
        }
      ]
    }
  ],
  "configuredSnssaiDto": [
    {
      "plmn": {
        "mcc": "100",
```

```
        "mnc": "101"
      },
      "nssai": [
        {
          "sst": 5,
          "sd": "EABB01"
        }
      ],
      "plmnId": "100-101"
    }
  ],
  "amfSetDto": [
    {
      "regionId": "01",
      "setId": "104",
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "salience": 1,
      "mccmncRegionidSetId": "100-101-01-104"
    }
  ],
  "mappingofNssaisDto": [
    {
      "plmnId": {
        "mcc": "100",
        "mnc": "101"
      },
      "mappingOfNssai": [
        {
          "servingSnssai": {
            "sst": 1,
            "sd": "EABB01"
          },
          "homeSnssai": {
            "sst": 7,
            "sd": "EABB07"
          }
        },
        {
          "servingSnssai": {
            "sst": 2,
            "sd": "EABB02"
          },
          "homeSnssai": {
            "sst": 8,
            "sd": "EABB08"
          }
        }
      ]
    }
  ],
  "nssfSystemOptionDtoList": [
    {
      "NssfSystemOptionService": [
```

```
        {
          "scope": "NsSelection",
          "PerPLMNConfiguration": [
            {
              "PlmnConfiguration": {
                "EnableEnhancedAllowedNSSAIComputation": false
              },
              "plmnId": {
                "mcc": "311",
                "mnc": "480"
              }
            },
            {
              "PlmnConfiguration": {
                "EnableEnhancedAllowedNSSAIComputation": true
              },
              "plmnId": {
                "mcc": "100",
                "mnc": "101"
              }
            }
          ]
        }
      ],
      "AutoConfigurationFromNsAvailability": "FromTrustedAMFs",
      "EnhancedPatchBehaviour": false
    }
  ],
  "trustedAmfDtoList": [
    {
      "amfSet": "100-101-01-101",
      "amfIdList": [
        "4faf1bbc-6e4a-4454-a507-aef01a101a01",
        "4faf1bbc-6e4a-4454-a507-aef01a101a02"
      ]
    },
    {
      "amfSet": "100-101-01-103",
      "amfIdList": [
        "4faf1bbc-6e4a-4454-a507-aef01a101a03",
        "4faf1bbc-6e4a-4454-a507-aef01a101a04"
      ]
    }
  ]
}
```

3.10 NRF-Client

The NRF-Client manages configuration of profiles by retrieving the NfProfiles and performing complete update of the NfProfiles used for registering with NRF.

NRF-Client - Supported REST APIs

The operator can configure the NRF-Client by following the information provided in the table below. The supported operations are **GET** and **PUT**. The following table provides information about the REST APIs supported by the NRF-Client Managed Object:

Table 3-28 NRF-Client - Supported REST APIs

URI	Description	Details
{nfType}/nf-common-component/v1/nrf-client-nfmanagement/nfProfileList	Retrieve the NfProfiles used to register with NRF	**Data Type:** Array(NFProfile) **HTTP Method:** GET
{nfType}/nf-common-component/v1/nrf-client-nfmanagement/nfProfileList	Complete update of the NfProfiles used to register with NRF	**Data Type:** Array(NFProfile) **HTTP Method:** PUT

Request or Response Body Parameters

Table 3-29 configmapApplicationConfig.profile.appProfiles Parameters

Parameter	Description	Details
`nfInstanceId`	This is a mandatory parameter. Unique identifier for the network function instance.	**Data Type:** String **Range:** UUID
`nfType`	This is a mandatory parameter. Type of the network function (for example, NSSF).	**Data Type:** String **Range:** Predefined types (for example, NSSF)
`nfStatus`	This is a mandatory parameter. Status of the network function.	**Data Type:** String **Range:** REGISTERED, UNREGISTERED, etc.
`heartBeatTimer`	This is a conditional parameter. Time in seconds expected between two consecutive heartbeat messages from an NF Instance to the NRF. It may be included in the registration request. When present in the request, it shall contain the heartbeat time proposed by the NF service consumer. It shall be included in responses from NRF to registration requests (PUT) or in NF profile updates (PUT or PATCH). If the proposed heartbeat time is acceptable by the NRF based on the local configuration, it shall use the same value as in the registration request. Otherwise the NRF shall override the value using a preconfigured value.	**Data Type:** Integer **Range:** 1-3600 seconds

Table 3-29 (Cont.) configmapApplicationConfig.profile.appProfiles Parameters

Parameter	Description	Details
fqdn	This is a conditional parameter. Fully Qualified Domain Name of the network function.	**Data Type:** String **Range:** Valid domain
priority	This is an optional parameter. Priority (relative to other NFs of the same type) in the range of 0-65535, to be used for NF selection; lower values indicate a higher priority. If priority is also present in the nfServiceList parameters, those will have precedence over this value. The NRF may overwrite the received priority value when exposing an NFProfile with the NRF Discovery service.	**Data Type:** Integer **Range:** 0-65535
capacity	This is an optional parameter. Static capacity information in the range of 0-65535, expressed as a weight relative to other NF instances of the same type; if capacity is also present in the nfServiceList parameters, those will have precedence over this value.	**Data Type:** Integer **Range:** 0-65535
load	This is an optional parameter. Dynamic load information, ranged from 0 to 100, indicates the current load percentage of the NF.	**Data Type:** Integer **Range:** 0-100
plmnList.mcc	This is a conditional parameter. Mobile Country Code (MCC) of the PLMN.	**Data Type:** String **Range:** 3-digit numeric code
plmnList.mnc	This is a conditional parameter. Mobile Network Code (MNC) of the PLMN.	**Data Type:** String **Range:** 2- or 3-digit numeric code
nfSetIdList	This is a conditional parameter. NF Set ID defined in clause 28.12 of 3GPP TS 23.003 [12]. At most one NF Set ID shall be indicated per PLMN of the NF.	**Data Type:** array(NfSetId) **Range:** Valid String
locality	This is an optional parameter. Operator defined information about the location of the NF instance (for example, geographic location, data center).	**Data Type:** String **Range:** Valid location String
nfServices.serviceInstanceId	This is a mandatory parameter. Unique ID of the service instance within a given NF Instance.	**Data Type:** String **Range:** UUID

Table 3-29 (Cont.) configmapApplicationConfig.profile.appProfiles Parameters

Parameter	Description	Details
`nfServices.serviceName`	This is a mandatory parameter. Name of the service provided by the NF.	**Data Type:** String **Range:** Predefined service names
`nfServices.versions.expiry`	This is an optional parameter. Expiry date and time of the NF service.	**Data Type:** DateTime **Range:** ISO 8601 timestamp or null
`nfServices.versions.apiFullVersion`	This is a mandatory parameter. Full version number of the API as specified in clause 4.3.1 of 3GPP 1 29.501.	**Data Type:** String **Range:** x.y.z (for example, 1.0.0)
`nfServices.versions.apiVersionInUri`	This is a mandatory parameter. Version of the service instance to be used in the URI for accessing the API (for example, "v1").	**Data Type:** String **Range:** Version String (for example, v1)
`nfServices.scheme`	This is a mandatory parameter. Communication scheme (for example, HTTP, HTTPS).	**Data Type:** String(UriScheme) **Range:** http, https
`nfServices.nfServiceStatus`	This is a mandatory parameter. Status of the NF Service Instance	**Data Type:** String **Range:** REGISTERED, UNREGISTERED
`nfServices.fqdn`	This is an optional parameter. Fully Qualified Domain Name of the service endpoint.	**Data Type:** String **Range:** Valid domain String
`nfServices.interPlmnFqdn`	This is a conditional parameter. If the NF service needs to be discoverable by other NFs in a different PLMN, then an FQDN that is used for inter PLMN routing as specified in 3GPP 1 23.003 [12] may be registered with the NRF. A change of this attribute shall result in triggering a "NF_PROFILE_CHANGED" notification from NRF towards subscribing NFs located in a different PLMN, but the new value shall be notified as a change in the "fqdn" attribute.	**Data Type**: String or null **Range**: Valid domain String or null
`nfServices.ipEndPoints.ipv4Address`	This is an optional parameter. IPv4 address of the service endpoint.	**Data Type:** String **Range:** Valid IPv4 address
`nfServices.ipEndPoints.transport`	This is an optional parameter. Transport protocol used by the service (for example, TCP).	**Data Type:** String **Range:** TCP, UDP
`nfServices.ipEndPoints.port`	This is an optional parameter. Port number for the service endpoint.	**Data Type:** Integer **Range:** 1-65535

Table 3-29 (Cont.) configmapApplicationConfig.profile.appProfiles Parameters

Parameter	Description	Details
nfServices.allowedNfTypes	This is an optional parameter. Type of the NFs allowed to access the service instance. The absence of this attribute indicates that any NF type is allowed to access the service instance. A change of this attribute shall not trigger a "NF_PROFILE_CHANGED" notification from NRF, and this attribute shall not be included in profile change notifications to subscribed NFs.	**Data Type:** Array of Strings **Range:** Valid NF types
nfServices.priority	This is an optional parameter. Priority (relative to other services of the same type) in the range of 0-65535, to be used for NF Service selection; lower values indicate a higher priority. The NRF may overwrite the received priority value when exposing an NFProfile to the NRF Discovery service.	**Data Type:** Integer **Range:** 0-65535
nfServices.capacity	This is an optional parameter. Static capacity information in the range of 0-65535, expressed as a weight relative to other services of the same type.	**Data Type:** Integer **Range:** 0-65535
nfServices.load	This is an optional parameter. Dynamic load information, ranging from 0 to 100, indicates the current load percentage of the NF Service.	**Data Type:** Integer **Range:** 0-100

nrfProfileList

Configuration Example

PUT Request

```
curl -L -X PUT 'http://ocnssf-nsconfig.ocnssf:8080/nssf/nf-common-
component/v1/nrf-client-nfmanagement/nfProfileList' -H 'Content-Type:
application/json' -d '<JsonRequestBody>'
```

Example of Request Body

```
[
  {
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
    "nfType": "NSSF",
```

```
"nfStatus": "REGISTERED",
"fqdn": "ocnssf-nsgateway.ocnssf.svc",
"priority": 1,
"capacity": 1,
"load": 2,
"plmnList": [
  {
    "mcc": "310",
    "mnc": "14"
  }
],
"locality": "rcnltxekloc1",
"nfServices": [
  {
    "serviceInstanceId": "92d59bfc-e5d6-47f5-a26b-3a03facdebcc",
    "serviceName": "nnssf-nsselection",
    "versions": [
      {
        "expiry": null,
        "apiFullVersion": "1.0.0",
        "apiVersionInUri": "v1"
      }
    ],
    "scheme": "http",
    "nfServiceStatus": "REGISTERED",
    "fqdn": "ocnssf1-ingress-gateway.ocnssf.svc",
    "interPlmnFqdn": null,
    "ipEndPoints": [
      {
        "ipv4Address": "10.224.45.178",
        "transport": "TCP",
        "port": 80
      }
    ],
    "allowedNfTypes": [
      "AMF",
      "NSSF"
    ],
    "priority": 1,
    "capacity": 1,
    "load": 2
  },
  {
    "serviceInstanceId": "d33728cd-6e21-434b-bc5a-ed69bc612377",
    "serviceName": "nnssf-nssaiavailability",
    "versions": [
      {
        "expiry": null,
        "apiFullVersion": "1.0.0",
        "apiVersionInUri": "v1"
      }
    ],
    "scheme": "http",
    "nfServiceStatus": "REGISTERED",
    "fqdn": "ocnssf2-ingress-gateway.ocnssf.svc",
    "interPlmnFqdn": null,
```

```
        "ipEndPoints": [
          {
            "ipv4Address": "10.224.45.179",
            "transport": "TCP",
            "port": 80
          }
        ],
        "allowedNfTypes": [
          "AMF",
          "NSSF"
        ],
        "priority": 1,
        "capacity": 1,
        "load": 2
      }
    ]
  }
]
```

3.11 Runtime Log Level Update

The Runtime Log Level Update Managed Object enables an operator to centralize the configuration and update it dynamically without restarting NSSF.

Table 3-30 Runtime Log Level Update - Parameters

Parameter	Description	Details
`commonCfgClient.enabled`	This is a mandatory parameter. It is used to enable or disable client.	**Data Type:** Boolean **Range:** true or false **Default Value:** true
`commonCfgClient.pollingInterval`	This is a mandatory parameter. It is used to set polling interval in Milliseconds.	**Data Type:** Integer **Default Value:** 5000

Runtime Log Level Update - Supported REST APIs

The operator can configure the Runtime Log Level Update by following the information provided in the table below. The supported operations are **GET** and **PUT**. The following table provides information about the REST APIs supported by the Runtime Log Level Update Managed Object:

Table 3-31 Runtime Log Level Update - Supported REST APIs

URI	Description	Details
http://<nssf-url>/nssf/nf-common-component/v1/services	Get the list of microservices registered with common config service for dynamic logging.	**Data Type:** array **HTTP Method:** GET
http://<nssf-url>/nssf/nf-common-component/v1/<microservice-name>/logging	To get the current Logging information of the microservice.	**Data Type:** array **HTTP Method:** GET
http://<nssf-url>/nssf/nf-common-component/v1/<microservice-name>/logging	To update the Logging information of the microservice.	**Data Type:** array **HTTP Method:** PUT

Examples API Calls

1. GET All Services

cURL: http://<nssf-url>/nssf/nf-common-component/v1/services

2. GET Log Level for a Microservice

cURL: http://<nssf-url>/nssf/nf-common-component/v1/<microservice-name>/logging

Example Response:

```
{
  "appLogLevel": "INFO",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
  ]
}
```

3. PUT to Update Log Level

cURL: http://<nssf-url>/nssf/nf-common-component/v1/<microservice-name>/logging

Example Request:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
  ]
}
```

Per-Microservice Logging Endpoints and Payloads

1. App-Info

- **GET**: `/nssf/nf-common-component/v1/app-info/logging`
- **PUT**: `/nssf/nf-common-component/v1/app-info/logging`

PUT Payload Example:

```
{

  "appLogLevel": "DEBUG"

}
```

Table 3-32 App-Info

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

2. Perf-Info

- **GET**: /nssf/nf-common-component/v1/perf-info/logging
- **PUT**: /nssf/nf-common-component/v1/perf-info/logging

PUT Payload Example:

```
{

  "appLogLevel": "WARN"

}
```

Table 3-33 Perf-Info

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

3. Ingress Gateway

- **GET**: /nssf/nf-common-component/v1/igw/logging
- **PUT**: /nssf/nf-common-component/v1/igw/logging

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "logDiscarding": {
    "enabled": false,
    "featureToThresholdMapping": [
      {
        "feature": "RATE_LIMITING",
        "thresholdFactor": 100
      },
      {
        "feature": "OVERLOAD_CONTROL",
        "thresholdFactor": 100
      },
      {
        "feature": "ROUTE_LEVEL_RATE_LIMITING",
        "thresholdFactor": 100
      },
      {
```

```
            "feature": "RSS_RATE_LIMITING",
            "thresholdFactor": 100
          },
          {
            "feature": "EGRESS_RATE_LIMITING",
            "thresholdFactor": 100
          }
        ]
      },
      "packageLogLevel": [
        {
          "packageName": "root",
          "logLevelForPackage": "ERROR"
        },
        {
          "packageName": "oauth",
          "logLevelForPackage": "ERROR"
        }
      ],
      "logSubscriberInfo": "DISABLED",
      "additionalErrorLogging": "DISABLED"
    }
```

Table 3-34 Ingress Gateway

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
logDiscarding.enabled	This is a mandatory parameter Enable/disable log discarding	**Data Type:** Boolean **Range:** true, false
featureToThresholdMapping	This is a mandatory parameter Feature thresholds list	**Data Type:** List of Objects **Range:** Each item contains feature and thresholdFactor
feature	This is a mandatory parameter Feature name	**Data Type:** String **Range:** • RATE_LIMITING • OVERLOAD_CONTROL • ROUTE_LEVEL_RATE_LIMITING • RSS_RATE_LIMITING • EGRESS_RATE_LIMITING
thresholdFactor	This is a mandatory parameter Threshold factor for feature	**Data Type:** Integer **Range:** 0–100
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name

Table 3-34 (Cont.) Ingress Gateway

Parameter Name	Description	Details
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
logSubscriberInfo	This is an optional parameter Subscriber info logging setting	**Data Type:** String **Range:** ENABLED, DISABLED
additionalErrorLogging	This is an optional parameter Additional error logging setting	**Data Type:** String **Range:** ENABLED, DISABLED

4. Egress Gateway

- **GET**: `/nssf/nf-common-component/v1/egw/logging`
- **PUT**: `/nssf/nf-common-component/v1/egw/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "logDiscarding": {
    "enabled": false,
    "featureToThresholdMapping": [
      {
        "feature": "RATE_LIMITING",
        "thresholdFactor": 100
      },
      {
        "feature": "NOTIFICATION_RATE_LIMITING",
        "thresholdFactor": 100
      }
    ]
  },
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    },
    {
      "packageName": "oauth",
      "logLevelForPackage": "INFO"
    }
  ],
  "logSubscriberInfo": "DISABLED",
  "additionalErrorLogging": "DISABLED"
}
```

Table 3-35 Egress Gateway

Parameter Name	Description	Details
`appLogLevel`	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
`logDiscarding.enabled`	This is a mandatory parameter Enable/disable log discarding	**Data Type:** Boolean **Range:** true, false
`featureToThresholdMapping`	This is a mandatory parameter Feature thresholds list	**Data Type:** List of Objects **Range:** Each item contains feature and thresholdFactor
`feature`	This is a mandatory parameter Feature name	**Data Type:** String **Range:** • RATE_LIMITING • OVERLOAD_CONTROL • ROUTE_LEVEL_RATE_LIMITING • RSS_RATE_LIMITING • EGRESS_RATE_LIMITING
`thresholdFactor`	This is a mandatory parameter Threshold factor for feature	**Data Type:** Integer **Range:** 0–100
`packageLogLevel`	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
`packageName`	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
`logLevelForPackage`	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
`logSubscriberInfo`	This is an optional parameter Subscriber info logging setting	**Data Type:** String **Range:** ENABLED, DISABLED
`additionalErrorLogging`	This is an optional parameter Additional error logging setting	**Data Type:** String **Range:** ENABLED, DISABLED

5. Alternate Route

- **GET**: `/nssf/nf-common-component/v1/alt-route/logging`
- **PUT**: `/nssf/nf-common-component/v1/alt-route/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
```

```
  ],
  "additionalErrorLogging": "DISABLED"
}
```

Table 3-36 Alternate Route

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
additionalErrorLogging	This is an optional parameter Additional error logging setting	**Data Type:** String **Range:** ENABLED, DISABLED

6. NRF Client - Discovery

- **GET**: `/nssf/nf-common-component/v1/nrf-client-nfdiscovery/logging`
- **PUT**: `/nssf/nf-common-component/v1/nrf-client-nfdiscovery/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
  ],
  "logSubscriberInfo": "DISABLED",
  "additionalErrorLogging": "DISABLED"
}
```

Table 3-37 NRF Client - Discovery

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

Table 3-37 (Cont.) NRF Client - Discovery

Parameter Name	Description	Details
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
logSubscriberInfo	This is an optional parameter Subscriber info logging setting	**Data Type:** String **Range:** ENABLED, DISABLED
additionalErrorLogging	This is an optional parameter Additional error logging setting	**Data Type:** String **Range:** ENABLED, DISABLED

7. NRF Client - Management

- **GET**: `/nssf/nf-common-component/v1/nrf-client-nfmanagement/logging`
- **PUT**: `/nssf/nf-common-component/v1/nrf-client-nfmanagement/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "WARN"
    }
  ],
  "logSubscriberInfo": "DISABLED",
  "additionalErrorLogging": "DISABLED"
}
```

Table 3-38 NRF Client - Management

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage

Table 3-38 (Cont.) NRF Client - Management

Parameter Name	Description	Details
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
logSubscriberInfo	This is an optional parameter Subscriber info logging setting	**Data Type:** String **Range:** ENABLED, DISABLED
additionalErrorLogging	This is an optional parameter Additional error logging setting	**Data Type:** String **Range:** ENABLED, DISABLED

8. NsAvailability

- **GET**: `/nssf/nf-common-component/v1/nsavailability/logging`
- **PUT**: `/nssf/nf-common-component/v1/nsavailability/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "ERROR"
    }
  ]
}
```

Table 3-39 NsAvailability

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

9. NsAuditor

- **GET**: `/nssf/nf-common-component/v1/nsaudit/logging`
- **PUT**: `/nssf/nf-common-component/v1/nsaudit/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "ERROR"
    }
  ]
}
```

Table 3-40 NsAuditor

Parameter Name	Description	Details
`appLogLevel`	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
`packageLogLevel`	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
`packageName`	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
`logLevelForPackage`	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

10. NsConfig

- **GET**: `/nssf/nf-common-component/v1/nsconfig/logging`
- **PUT**: `/nssf/nf-common-component/v1/nsconfig/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "ERROR"
    }
  ]
}
```

Table 3-41 NsConfig

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

11. NsSelection

- **GET**: `/nssf/nf-common-component/v1/nsselection/logging`
- **PUT**: `/nssf/nf-common-component/v1/nsselection/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "ERROR"
    }
  ]
}
```

Table 3-42 NsSelection

Parameter Name	Description	Details
appLogLevel	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
packageLogLevel	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range:** Each item contains packageName and logLevelForPackage
packageName	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
logLevelForPackage	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

12. NsSubscription

- **GET**: `/nssf/nf-common-component/v1/nssubscription/logging`
- **PUT**: `/nssf/nf-common-component/v1/nssubscription/logging`

PUT Payload Example:

```
{
  "appLogLevel": "DEBUG",
  "packageLogLevel": [
    {
      "packageName": "root",
      "logLevelForPackage": "ERROR"
    }
  ]
}
```

Table 3-43 NsSubscription

Parameter Name	Description	Details
`appLogLevel`	This is a mandatory parameter Application-wide log level	**Data Type:** String **Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF
`packageLogLevel`	This is a mandatory parameter Specific package log settings	**Data Type:** List of Objects **Range**: Each item contains packageName and logLevelForPackage
`packageName`	This is a mandatory parameter Name of the package	**Data Type**: String **Range**: Any valid Java package name
`logLevelForPackage`	This is a mandatory parameter Log level assigned to the package	**Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF, OFF

3.12 PLMN Level NSI Profiles

PLMN Level NSI Profiles is a mandatory Managed Object. It is Default Network Slice Instance Profile per PLMN and consists of network slice ID, NRF-ID, and PLMN.

It is important to configure Rules based on NsAvailability Update, as all autoconfigured NSS_Rules are dependent on PLMN Level NSI Profiles. If this configuration is not provided for autoconfiguration of the Rules, the NsAvailability update will fail.

> **Note**
>
> Without this configuration, PDU establishment can fail for autoconfigured Auth.

- Request_Type: POST
- URL: *http://{apiRoot}/nnssf-configuration/v1/plmnlevelnsiprofiles*
- REST message sample - PLMN Level NSI Profile

- Content-Type: application/json
- Initiated by: API Invoker

PLMN Level NSI Profile - Supported REST APIs

The operator can configure PLMN Level NSI Profile by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the PLMN Level NSI Profile Managed Object:

Table 3-44 PLMN Level NSI Profile - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/plmnlevelnsiprofiles	Returns PLMNLevelNsiProfile for all supported PLMNs.	**Data Type:** array(PLMNLevelNsiProfile) **HTTP Method:** GET
/nnssf-configuration/v1/plmnlevelnsiprofiles/{mcc-mnc}	Create a generic fall back NSI profile (Nrf, Network slice instance).	**Data Type:** PLMNLevelNsiProfile **HTTP Method:** POST
/nnssf-configuration/v1/plmnlevelnsiprofiles/{mcc-mnc}	Get PLMNLevelNsiProfile for supported PLMN.	**Data Type:** PLMNLevelNsiProfile **HTTP Method:** GET
/nnssf-configuration/v1/plmnlevelnsiprofiles/{mcc-mnc}	Delete PLMNLevelNsiProfile for supported PLMN.	**Data Type:** PLMNLevelNsiProfile **HTTP Method:** DELETE
/nnssf-configuration/v1/plmnlevelnsiprofiles/{mcc-mnc}	Update PLMNLevelNsiProfile for supported PLMN.	**Data Type:** PLMNLevelNsiProfile **HTTP Method:** PUT

PLMN Level NSI Profiles- Dependencies

This managed object is not dependent on other managed objects.

Request or Response Body Parameters

The PLMN Level NSI Profiles managed object enables the configuration of Network Slice Instance profile. This allows an operator to create an Network Slice, by providing a name, id, NRF URL corresponding to the slice and list of Target AMF sets which support this slice.

Table 3-45 PLMN Level NSI Profiles - Parameters

Parameter	Description	Details
`nrfUri`	**M/C/O:** This is a mandatory parameter. Contains URI of the Network Repository Function. **Note**: It must contain NRF NF Discovery URL.	**Data Type:** String
`nsiId`	**M/C/O:** This is an optional parameter. Contains Network Slice Instance Identifier.	**Data Type:** String
`nrfNfMgtUri`	**M/C/O:** This is a mandatory parameter. Contains Management URI of Network Resource Function.	**Data Type:** String
`nrfAccessTokenUri`	**M/C/O:** This is an optional parameter. Contains Access Token URI of Network Resource Function.	**Data Type:** String
`plmn`	**M/C/O:** This is a mandatory parameter. Contains PLMN ID.	**Data Type:** plmn

Table 3-45 (Cont.) PLMN Level NSI Profiles - Parameters

Parameter	Description	Details
plmnid	**M/C/O:** This is a mandatory parameter. Contains value of MCC and MNC in the format "MCC"-"MNC"	**Data Type:** String

Supported Response Codes

Table 3-46 Supported Response Codes

Response Code	Description
201	PLMN Level NSI Profiles configured successfully
200	Successful response
204	No Content
400	Bad Request
403	Forbidden
404	Config not found
406	Not Acceptable client error

Examples

The following example shows how a PLMN Level NSI Profile is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl --http2-prior-knowledge -X POST 'http://{{host}}:{{port}}/nnssf-
configuration/v1/plmnlevelnsiprofiles' --header 'Content-Type: application/
json' --data-raw '{"name":"NSI-PROFILE-1","nrfUri":"https://
nrf.slice1.oracle.com/nnrf-disc/v1","nrfNfMgtUri":"https://
nrf.slice1.oracle.com/nnrf-nfm/v1","nrfAccessTokenUri":"https://
nrf.slice1.oracle.com/oauth2/token","nsiId":"SLICE1","plmn":
{"mcc":"100","mnc":"101"}}'
```

Example of the Request Body

The following snippet shows an example of the request body:

```
{
  "name": "NSI-PROFILE-1",
  "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
  "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
  "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
  "nsiId": "SLICE1",
  "plmn": {
    "mcc": "100",
    "mnc": "101"
  }
}
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
{
  "name": "NSI-PROFILE-1",
  "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v1",
  "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v1",
  "nrfAccessTokenUri": "https://nrf.slice1.oracle.com/oauth2/token",
  "nsiId": "SLICE1",
  "plmn": {
    "mcc": "100",
    "mnc": "101"
  }
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/plmnlevelnsiprofiles/{mcc-mnc}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-
configuration/v1/plmnlevelnsiprofiles/100-101"
```

3.13 MappingOfNssai

MappingOfNssai is a mandatory Managed Object to support request mapping. It is added to support EPS to 5G handover and contains mapping of 4G S-NSSAI to 5G S-NSSAI for a given PLMN.

> **Note**
>
> This Managed Object must be configured for each supported PLMN.

- Request_Type: POST
- URL: *http://{apiRoot}/nnssf-configuration/v1/mappingOfNssais*
- REST message sample - MappingOfNssai
- Content-Type: application/json
- Initiated by: API Invoker

MappingOfNssai - Supported REST APIs

The operator can configure MappingOfNssai by following the information provided in the table below. The supported operations are **POST**, **GET**, **DELETE**, and **PUT**. The following table provides information about the REST APIs supported by the MappingOfNssai managed object:

Table 3-47 MappingOfNssai - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/mappingOfNssais/{mcc-mnc}	Returns MappingOfNssais for the PLMN in request parameters.	**Data Type:** array(MappingOfNssai **HTTP Method:** GET
/nnssf-configuration/v1/mappingOfNssais	Returns MappingOfNssais for all supported PLMNs.	**Data Type:** array(PlmnMappingOfNssai) **HTTP Method:** GET
/nnssf-configuration/v1/mappingOfNssais	Create MappingOfNssais for the PLMN.	**Data Type:** array(MappingOfNssai) **HTTP Method:** POST
/nnssf-configuration/v1/mappingOfNssais/{mcc-mnc}	Update MappingOfNssais for the PLMN.	**Data Type:** array(MappingOfNssai) **HTTP Method:** PUT
/nnssf-configuration/v1/mappingOfNssais/{mcc-mnc}	Delete MappingOfNssais for the PLMN.	**Data Type:** NA **HTTP Method:** DELETE
/nnssf-configuration/v1/ mappingOfNssais{mcc=?mnc=?homeSnssai=? }	Delete a MappingOfNssai for homeSnssai the PLMN.	**Data Type:** NA **HTTP Method:** DELETE
/nnssf-configuration/v1/ mappingOfNssais{mcc=?mnc=?homeSnssai=? }	Update ServingSnssai for homeSnssai the PLMN.	**Data Type:** MappingOfNssai **HTTP Method:** PUT
/nnssf-configuration/v1/ mappingOfNssaisrequestparam	Create MappingOfNssais for the PLMN.	**Data Type:** array(PlmnMappingOfNssai) **HTTP Method:** POST
/nnssf-configuration/v1/ mappingOfNssaisrequestparam	Returns MappingOfNssais for all supported PLMNs.	**Data Type:** array(PlmnMappingOfNssai) **HTTP Method:** GET ALL
/nnssf-configuration/v1/ mappingOfNssaisrequestparam{mcc=?mnc=?}	Returns MappingOfNssais for the PLMN in request parameters.	**Data Type:** array(MappingOfNssai) **HTTP Method:** GET
/nnssf-configuration/v1/ mappingOfNssaisrequestparam{mcc=?mnc=?}	Update MappingOfNssais for the PLMN.	**Data Type:** array(MappingOfNssai) **HTTP Method:** PUT
/nnssf-configuration/v1/ mappingOfNssaisrequestparam{mcc=?mnc=?}	Delete MappingOfNssais for the PLMN.	**Data Type:** NA **HTTP Method:** DELETE
/nnssf-configuration/v1/ mappingOfNssaisrequestparam{mcc=?mnc=? homeSnssai=? }	Delete a MappingOfNssai for homeSnssai the PLMN.	**Data Type:** NA **HTTP Method:** DELETE
/nnssf-configuration/v1/ mappingOfNssaisrequestparam{mcc=?mnc=? homeSnssai=? }	Update ServingSnssai for homeSnssai the PLMN.	**Data Type:** MappingOfNssai **HTTP Method:** PUT

MappingOfNssai- Dependencies

This managed object is not dependent on other managed objects.

Request or Response Body Parameters

Table 3-48 MappingOfNssai - Parameters

Parameter	Description	Details
`mcc`	This is a mandatory parameter. Specifies Mobile Country Code.	**Data Type:** Integer
`mnc`	This is a mandatory parameter. Specifies Mobile Network Code.	**Data Type:** Integer
`mappingOfNssai`	This is a mandatory parameter. Specifies an array of MappingOfSnssai. For more information, see MappingOfSnssai.	**Data Type:** array(MappingOfSnssai)

Supported Response Codes

Table 3-49 Supported Response Codes

Response Code	Description
201	API invoker onboarded successfully
200	Successful response
204	No Content
400	Bad Request
403	Forbidden
404	Config not found
406	Not Acceptable client error
500	Internal Server Error

Examples

The following example shows how a MappingOfNssai is onboarded by submitting a POST request on the REST resource using cURL:

cURL Command:

```
curl -v --http2-prior-knowledge -X POST "http://{{host}}:{{port}}/nnssf-
configuration/v1/mappingOfNssais" -H 'Content-Type: application/json' -d
'{"plmnId": {"mcc": "100","mnc": "102"},"mappingOfNssai": [{"servingSnssai":
{"sst": 1,"sd": "EABB01"},"homeSnssai": {"sst": 7,"sd": "EABB07"}},
{"servingSnssai": {"sst": 2,"sd": "EABB02"},"homeSnssai": {"sst": 8,"sd":
"EABB08"}}]}'
```

Example of the Request Body

The following snippet shows an example of the request body:

```
{
        "plmnId": {
            "mcc": "100",
            "mnc": "102"
        },
        "mappingOfNssai": [
            {
```

```
                "servingSnssai": {
                    "sst": 1,
                    "sd": "EABB01"
                },
                "homeSnssai": {
                    "sst": 7,
                    "sd": "EABB07"
                }
            },
            {
                "servingSnssai": {
                    "sst": 2,
                    "sd": "EABB02"
                },
                "homeSnssai": {
                    "sst": 8,
                    "sd": "EABB08"
                }
            }
        ]
}
```

Example of the Response Body

The following example shows the contents of the response body in JSON format:

```
{
    "plmnId": {
        "mcc": "100",
        "mnc": "102"
    },
    "mappingOfNssai": [
        {
            "servingSnssai": {
                "sst": 1,
                "sd": "EABB01"
            },
            "homeSnssai": {
                "sst": 7,
                "sd": "EABB07"
            }
        },
        {
            "servingSnssai": {
                "sst": 2,
                "sd": "EABB02"
            },
            "homeSnssai": {
                "sst": 8,
                "sd": "EABB08"
            }
        }
    ]
}
```

Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-configuration/v1/mappingOfNssais/{mcc-mnc}"
```

Example of Delete cURL

```
curl  --http2-prior-knowledge -X DELETE "http://{{host}}:{{port}}/nnssf-configuration/v1/mappingOfNssais/100-102"
```

3.14 NSSF System Options

Managed object to configure `NssfSystemOptions`. It contains parameters to configure multiple features such as, enhanced computation of `AllowedNSSAI` and enable AutoConfiguration from Trusted AMFs etc.

URI: *{apiRoot}/nnssf-configuration/v1/nssfsystemoptions*

Method:

- GET: Service Method to get the values of `EnableEnhancedAllowedNSSAIComputation`, `EnhancedPatchBehaviour`, and `AutoConfigurationFromNsAvailability`.
- PUT: Service Method to update the values of `EnableEnhancedAllowedNSSAIComputation`, `EnhancedPatchBehaviour`, and `AutoConfigurationFromNsAvailability`.

Content-Type: application/json

Request or Response Body Parameters

Table 3-50 Request or Response Body Parameters

Parameter	Description	Details
`NssfSystemOptionService`	**M/C/O:** This is a mandatory parameter. **Description**: It is the main object that contains other objects required for enhanced computation of allowedNSSAI in NSSF.	**Data Type:** Object **Range:** NA **Default Value:** NA
`NssfSystemOptionService.scope`	**M/C/O:** This is a mandatory parameter. **Description**: Name of the NSSF Service that the feature applies to. Currently, it is limited to NsSelection only.	**Data Type:** String **Range:** NA **Default Value:** NsSelection

Table 3-50 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
`NssfSystemOptionService.PerPLMNConfiguration`	**M/C/O:** This is a mandatory parameter. **Description**: It is the object that contains objects for PLMN level configurations required for enhanced computation of allowedNSSAI in NSSF.	**Data Type:** Object **Range:** NA **Default Value:** NA
`NssfSystemOptionService.PerPLMNConfiguration.PlmnConfiguration`	**M/C/O:** This is a mandatory parameter. **Description**: It contains objects for PLMN level configurations required for enhanced computation of allowedNSSAI in NSSF.	**Data Type:** Object **Range:** NA **Default Value:** NA
`NssfSystemOptionService.PerPLMNConfiguration.PlmnConfiguration.EnableEnhancedAllowedNSSAIComputation`	**M/C/O:** This is a mandatory parameter. **Description**: Parameter to enable or disable the feature for a PLMN	**Data Type:** Boolean **Range:** true or false **Default Value:** false
`NssfSystemOptionService.PerPLMNConfiguration.plmnId`	**M/C/O:** This is a mandatory parameter. **Description**: Object that contains PLMN details	**Data Type:** Object **Range:** NA **Default Value:** NA
`NssfSystemOptionService.PerPLMNConfiguration.plmnId.mnc`	**M/C/O:** This is a mandatory parameter. **Description**: MNC of the PLMN	**Data Type:** Integer **Range:** Valid MNC **Default Value:** NA
`NssfSystemOptionService.PerPLMNConfiguration.plmnId.mcc`	**M/C/O:** This is a mandatory parameter. **Description**: MCC of the PLMN	**Data Type:** Integer **Range:** Valid MCC **Default Value:** NA

Table 3-50 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
`AutoConfigurationFromNsAvailability`	**M/C/O**: This is a mandatory parameter. **Description**: This parameter is used to configure NSSF's behavior with respect to "Auto-Population of Configuration Based on NSAvailability Update" feature. It is an enum with the following possible values: • `Disable`: This signifies the feature is disabled and is the default value. There will be no configuration update based on NsAvailability data if the value is disabled. • `FromTrustedAMFs`: This signifies NsAvailability Update only from trusted AMFs may lead to an update in NSSF configuration. • `FromAllAMFs`: This signifies that NsAvailability Update from any AMF may lead to an update in NSSF configuration. **Note**: Assuming that NsAvailability Update is processed by NSSF. That is, the AMF is authorized to send the availability update.	**Data Type:** String **Range:** Disable FromTrustedAMFs FromAllAMFs **Default Value:** Disable

Table 3-50 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
EnhancedPatchBehaviour	**M/C/O**: This is a mandatory parameter. **Description**: This parameter is used to allow patch remove request to remove last element of supportedSnssaiList and update AMF with empty list in database. If the value of the flag is true, then we should allow the patch remove request to remove the last element of supportedSnssaiList.	**Data Type**: Boolean **Range**: true or false **Default Value**: false

Supported Response Codes

Table 3-51 Supported Response Codes

Response Code	Description
200	Successful response
204	No content
400	Bad Request
404	Config not found

Example

Sample cURL:

```
curl -i --http2-prior-knowledge -X GET http://{{host}}:{{port}}/nnssf-configuration/v1/nssfsystemoptions
```

Example of GET Request or Response Body

```
{
  "NssfSystemOptionService": [
    {
      "scope": "NsSelection",
      "PerPLMNConfiguration": [
        {
          "PlmnConfiguration": {
            "EnableEnhancedAllowedNSSAIComputation": false
          },
          "plmnId": {
            "mcc": "311",
            "mnc": "480"
          }
        }
      ]
    }
```

```
  ],
  "AutoConfigurationFromNsAvailability": "Disable",
  "EnhancedPatchBehaviour": false
}
```

Example of PUT request for FromTrustedAMFs

```
{"EnhancedPatchBehaviour": true,
  "AutoConfigurationFromNsAvailability": "FromTrustedAMFs",
  "NssfSystemOptionService": [
    {
      "scope": "NsSelection",
      "PerPLMNConfiguration": [
        {
          "PlmnConfiguration": {
            "EnableEnhancedAllowedNSSAIComputation": true
          },
          "plmnId": {
            "mnc": "480",
            "mcc": "311"
          }
        }
      ]
    }
  ]
}
```

Example of PUT request for FromAllAMFs

```
{"EnhancedPatchBehaviour": true,
  "AutoConfigurationFromNsAvailability": "FromAllAMFs",
  "NssfSystemOptionService": [
    {
      "scope": "NsSelection",
      "PerPLMNConfiguration": [
        {
          "PlmnConfiguration": {
            "EnableEnhancedAllowedNSSAIComputation": true
          },
          "plmnId": {
            "mnc": "480",
            "mcc": "311"
          }
        }
      ]
    }
  ]
}
```

3.15 Trusted AMF

`TrustedAmf` is a managed object that lists an AMF as trusted and authorized to make configuration changes in the NSSF. Any update from these AMFs, as part of the NS-Availability

Update service from this operator-approved list of AMFs, will not require authorization from the NSSF (as mandated in 29.531). These updates will be automatically authorized, and the corresponding table entries will be auto-updated without operator intervention.

- **Request Type**: GET/POST/PUT/DELETE
- **URL**: *http://{apiRoot}/nnssf-configuration/v1/trustedamf*
- **Content-Type**: application/json
- **Initiated By**: API Invoker

TrustedAmf - Dependencies

To configure `TrustedAmf` Managed Object, configure the following Managed Objects as a prerequisite:

- **AmfSet**: The `AmfSet` under which the `TrustedAmf` falls must be preconfigured before creating or updating a `TrustedAmf`.

> **Note**
>
> Before an `AmfSet` is deleted then operator needs to remove the corresponding `TrustedAmf`.

TrustedAmf - Supported REST APIs

The operator can configure `TrustedAmf` by following the information provided in the table below. The supported operations are POST, GET, DELETE, and PUT. The following table provides information about the REST APIs supported by the `TrustedAmf` Managed Object:

Table 3-4 TrustedAmf - Supported REST APIs

Table 3-52 TrustedAmf - Supported REST APIs

URI	Description	Details
/nnssf-configuration/v1/trustedamf	Get all TrustedAmfs	**Data Type**: array(TrustedAmf) **HTTP Method**: GET
/nnssf-configuration/v1/trustedamf	Creates a list of TrustedAmfs	**Data Type**: array(TrustedAmf) **HTTP Method**: POST
/nnssf-configuration/v1/trustedamf	Update the whole list of TrustedAmfs	**Data Type**: array(TrustedAmf) **HTTP Method**: PUT
/nnssf-configuration/v1/trustedamf/{amfSet}	Update TrustedAmfs based on a particular AmfSet	**Data Type**: TrustedAmf **HTTP Method**: PUT
/nnssf-configuration/v1/trustedamf/{amfId}	Delete a TrustedAmf by amfId	**Data Type**: NA **HTTP Method**: DELETE

Request or Response Body Parameters

Table 3-53 Request or Response Body Parameters

Parameter	Description	Details
`amfSet`	**M/C/O:** This is a mandatory parameter. **Description**: It contains the AMF Set.	**Data Type:** String **Range:** NA **Default Value:** NA
`amfIdList`	**M/C/O:** This is a mandatory parameter. **Description**: It contains a list of AMF Identifiers.	**Data Type:** array(String) **Range:** NA **Default Value:** NA

Supported Response Codes

Table 3-54 Supported Response Codes

Response Code	Description
201	TrustedAmf configured successfully
200	TrustedAmf updated successfully
204	TrustedAmf deleted successfully
400	Bad Request
404	Not Found
409	Constraint Violation
406	Not Acceptable client error

Examples

The following example shows how an `TrustedAmf` is onboarded by submitting a GET/POST/PUT/DELETE request on the REST resource using cURL:

Sample GET cURL Command:

```
curl -i --http2-prior-knowledge -X GET http://{{host}}:{{port}}/nnssf-configuration/v1/trustedamf
```

Example of Response Body

```
[
  {
    "amfSet": "100-101-01-101",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a01",
      "4faf1bbc-6e4a-4454-a507-aef01a101a02"
    ]
  },
  {
    "amfSet": "100-101-01-103",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a03",
      "4faf1bbc-6e4a-4454-a507-aef01a101a04"
    ]
```

```
  }
]
```

Sample PUT cURL Command 1:

```
curl -i --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/nnssf-
configuration/v1/trustedamf -H 'Content-Type: application/json' -d '[ { "amfSet":
"100-101-01-101", "amfIdList": [ "4faf1bbc-6e4a-4454-a507-aef01a101a01",
"4faf1bbc-6e4a-4454-a507-aef01a101a02" ] }, { "amfSet": "100-101-01-103",
"amfIdList": [ "4faf1bbc-6e4a-4454-a507-aef01a101a03", "4faf1bbc-6e4a-4454-a507-
aef01a101a04" ] } ]'
```

Example of Response Body

```
[
  {
    "amfSet": "100-101-01-101",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a01",
      "4faf1bbc-6e4a-4454-a507-aef01a101a02"
    ]
  },
  {
    "amfSet": "100-101-01-103",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a03",
      "4faf1bbc-6e4a-4454-a507-aef01a101a04"
    ]
  }
]
```

Sample PUT cURL Command 2:

```
curl -i --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/nnssf-
configuration/v1/trustedamf -H 'Content-Type: application/json' -d '[ { "amfSet":
"100-101-01-101", "amfIdList": [ "4faf1bbc-6e4a-4454-a507-aef01a101a08" ] },
{ "amfSet": "100-101-01-103", "amfIdList": [ "4faf1bbc-6e4a-4454-a507-
aef01a101a09", "4faf1bbc-6e4a-4454-a507-aef01a101a10" ] }, { "amfSet":
"100-101-02-102", "amfIdList": [ "4faf1bbc-6e4a-4454-a507-aef01a101a11",
"4faf1bbc-6e4a-4454-a507-aef01a101a12" ] } ]'
```

Example of Response Body

```
[
  {
    "amfSet": "100-101-01-101",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a08"
    ]
  },
  {
    "amfSet": "100-101-01-103",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a09",
      "4faf1bbc-6e4a-4454-a507-aef01a101a10"
    ]
  },
```

```
  {
    "amfSet": "100-101-02-102",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a11",
      "4faf1bbc-6e4a-4454-a507-aef01a101a12"
    ]
  }
]
```

Sample PUT cURL Command for a particular AmfSet:

```
curl -i --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/nnssf-
configuration/v1/trustedamf/{amfSet} -H 'Content-Type: application/json' -d
'<Request-Body>'
```

Example of Request Body

```
{
  "amfSet": "100-101-02-102",
  "amfIdList": [
    "4faf1bbc-6e4a-4454-a507-aef01a101a12",
    "4faf1bbc-6e4a-4454-a507-aef01a101a13",
    "4faf1bbc-6e4a-4454-a507-aef01a101a14"
  ]
}
```

Sample cURL:

```
curl -i --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/nnssf-
configuration/v1/trustedamf/100-101-02-102 -H 'Content-Type: application/
json' -d '{ "amfSet": "100-101-02-102", "amfIdList": [ "4faf1bbc-6e4a-4454-
a507-aef01a101a12", "4faf1bbc-6e4a-4454-a507-aef01a101a13",
"4faf1bbc-6e4a-4454-a507-aef01a101a14" ] }'
```

Example of Response Body

```
[
  {
    "amfSet": "100-101-01-101",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a08"
    ]
  },
  {
    "amfSet": "100-101-01-103",
    "amfIdList": [
      "4faf1bbc-6e4a-4454-a507-aef01a101a09",
      "4faf1bbc-6e4a-4454-a507-aef01a101a10"
    ]
  },
  {
    "amfSet": "100-101-02-102",
    "amfIdList": [
```

```
      "4faf1bbc-6e4a-4454-a507-aef01a101a12",
      "4faf1bbc-6e4a-4454-a507-aef01a101a13",
      "4faf1bbc-6e4a-4454-a507-aef01a101a14"
    ]
  }
]
```

DELETE cURL Command:

```
curl -i --http2-prior-knowledge -X DELETE http://{{host}}:{{port}}/nnssf-
configuration/v1/trustedamf/{amfId}
```

Example of Delete cURL

```
curl -i --http2-prior-knowledge -X DELETE http://{{host}}:{{port}}/nnssf-
configuration/v1/trustedamf/4faf1bbc-6e4a-4454-a507-aef01a101a14
```

4

OAuth Validator REST API Configuration

This REST API configuration is required for enabling access token validation using NRF Instance ID and key-ID (K-ID).

Before this configuration, perform the prerequisite steps and helm configuration explained in **OAuth Access Token Based Authorization Using Key-ID and NRF Instance ID** section of *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

After Helm configuration, send the REST requests to use configured public key certificates. Using REST-based configuration, you can distinguish between the certificates configured on different NRFs and can use these certificates to validate the token received from a specific NRF.

OAuth Validator Configuration - Supported REST APIs

Table 4-1 OAuth Validator Configuration - Supported REST APIs

URI	Description	Details

/{nfType}/nf-common-component/v1/{serviceName}/{resource}
With service name as "igw" and resource as "oauthvalidatorconfiguration".
	These URIs can be used to update or add oauth configuration that will be used for validating token sent in request to Ingress Gateway.
	Note: By default, `instanceIdList`, `keyIdList` are null and validation mode is `INSTANCEID_ONLY`. This configuration is only applicable when oauth feature is enabled via helm chart.
		`{"keyIdList": [ { "keyId": "664b344e74294c8fa5d2e7dfaaaba407", "kSecretName": "oauthsecret", "certName": "4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt", "certAlgorithm": "ES256" } ], "oauthValidationMode": "KID_ONLY" }`
		
		or
		
		`{ "instanceIdList": [ { "instanceId": "4bc0c762-0212-416a-bd94-b7f1fb348bd4", "certName": "4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt", "kSecretName": "oauthsecret", "certAlgorithm": "ES256" } ], "oauthValidationMode": "INSTANCEID_ONLY" }`
		
		Note: Multiple OAuth and K-IDs can also be configured together in a single body.

Here, we can configure multiple Key-ID and instance ID objects of different NRFs. Validation of token is done based on the NRF Instance ID received in the token.

We have configurations like `instanceId` of NRF or `keyId` configured in NRF, secret name in which certificate is stored with certificate name and certificate algorithm. Once this request is sent, Oauth validator will pick appropriate certificate from secret configured.

We have "`oAuthValidationMode`" where we can select mode of validation. That is, `INSTANCEID_ONLY`, `KID_ONLY`, or `KID_PREFERRED` It will check for `keyIdList` or `instanceIdList` for validation of token received based on mode selected.

"`KID_PREFERRED`" is a fall back mode where it checks for `keyId` in token, If token is present then validation mode is `KID_ONLY`, else it will fall back to `INSTANCEID_ONLY`.

Example:

For K-ID Only mode, a sample curl command could be the following:

```
curl -i --http2-prior-knowledge -X PUT 'http://ocnssf-nsconfig:{{port}}/
nssf/nf-common-component/v1/igw/oauthvalidatorconfiguration' -H 'Content-
Type: application/json' -d '{"keyIdList": [{"keyId":
"664b344e74294c8fa5d2e7dfaaaba407","kSecretName": "oauthsecret","certName":
"4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt","certAlgorithm":
"ES256"}],"oauthValidationMode":"KID_ONLY"}'
```

For INSTANCEID_ONLY mode, a sample curl command could be the following:

```
curl -i --http2-prior-knowledge -X PUT 'http://ocnssf-nsconfig:{{port}}/
nssf/nf-common-component/v1/igw/oauthvalidatorconfiguration' -H 'Content-
Type: application/json' -d '{"instanceIdList":
[{"instanceId":"4bc0c762-0212-416a-bd94-
b7f1fb348bd4","certName":"4bc0c762-0212-416a-bd94-
b7f1fb348bd4.crt","kSecretName":"oauthsecret","certAlgorithm":"ES256"}],"oauth
ValidationMode":"INSTANCEID_ONLY"}'
```

Using tokens generated, you can send requests to different NSSF services. Given below are the sample curl commands to send requests:

Request supportedNssaiAvailabilityData:

```
curl -v --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/nnssf-
nssaiavailability/v1/nssai-availability/12345678-abcd-efAB-CDEF-123456789012 -
H 'Content-Type: application/json'  -H 'Authorization: Bearer
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjY2NGIzNDRlNzQyOTRjOGZhNWQyZTdkZm
FhYWJhNDA3In0.eyJhdWQiOiJOU1NGIiwiZXhwIjo5OTk5OTk5OTk5OSwiaXNzIjoiNjY0YjM0NGU3
NDI5NGM4ZmE1ZDJlN2RmYWFhYmE0MDciLCJzY29wZSI6Im5uc3NmLWNvbmZpZ3VyYXRpb24iLCJzdW
IiOiIwMjYzNjYzZC1mNWMyLTRkMWItOTE3MC1mN2IxYTkxMTYzMzcifQ.K9GoneRxHBn9vrun6VVyF
pBhe0xv4icVuJdQLoPppx_3KB1qUTIYkLFHEz2K4O_DPByDNHjPz7Fhe4PMfk9xXw' -d
'{ "supportedNssaiAvailabilityData": [ { "tai": { "plmnId": { "mcc": "100",
"mnc": "101" }, "tac": "001001" }, "supportedSnssaiList": [ { "sd": "EABB01",
"sst": 1 }, { "sd": "EABB02", "sst": 2 } ] } ] }'
```

Request using instance id token:

```
curl -v --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/nssaiauth -H 'Content-Type: application/json' -H
'Authorization: Bearer
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJOU1NGIiwiZXhwIjo5OTk5OTk5OTk5O
SwiaXNzIjoiNGJjMGM3NjItMDIxMi00MTZhLWJkOTQtYjdmMWZiMzQ4YmQ0Iiwic2NvcGUiOiJubnN
zZi1jb25maWd1cmF0aW9uIiwic3ViIjoiMDI2MzY2M2QtZjVjMi00ZDFiLTkxNzAtZjdiMWE5MTE2M
zM3In0.OngSuKL2a2EquyQaEutrbaaz18SKXJUnr0hTP22rDvG4eXCUPA-
Q-9Kz2L3zWxp3_oQDIJbn5mC5BUEA5UIB_Q ' -d '{"name":"NSSAI-AUTH-15","plmnId":
{"mcc":"100","mnc": "101"},"tac":"1001","snssai":{"sst":
"7","sd":"EABB01"},"grant": "ALLOWED"}'
```

5
Common Services REST APIs

This chapter provides information about the REST specifications for Common Services used in Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF).

You can use these APIs to update configurations related to the Overload Control feature. For more information on Overload Control Feature, see *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

5.1 Ingress Gateway REST APIs

This section explains REST API configurations required at Ingress Gateway for various features.

5.1.1 serverheaderdetails Configurations to Enable Server Header

This API can be used for adding Server Header in the error responses sent from Ingress Gateway, depending on the values provided in the JSON. By default, this feature is disabled. To enable the feature, invoke the following REST API and update the enable or disable flag.

URI:

/{nfType}/nf-common-component/v1/{serviceName}/serverheaderdetails

Method: GET, PUT, PATCH

Content-Type: application/json

Request or Response Body Parameters

Table 5-1 Request or Response Body Parameters

Parameter	Description	Details
`enabled`	This is a mandatory parameter. This parameter specifies whether the feature is enabled or disabled.	**Data Type**: Boolean **Range**: true or false **Example Value**: false
`errorCodeSeriesId`	This is an optional parameter. Specifies the error list IDs	**Data Type**: String **Range**: NA **Example Value**: NA
`configuration.nfType`	This is a mandatory parameter. Specifies the type of network function	**Data Type**: String **Range**: NA **Example Value**: NSSF

Table 5-1 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
configuration.nfInstanceId	This is a mandatory parameter. This parameter represents the UUID of the NSSF deployment that is used to generate the Server Header.	**Data Type**: String **Range**: NA **Example Value**: Valid NSSF Instance ID

Example

Example of Request or Response Body to Enable Server Header:

```
{
  "enabled": true,
  "errorCodeSeriesId": "E1",
  "configuration": {
    "nfType": "NSSF",
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01"
  }
}
```

Note

- In the mentioned configuration, when sending a response to AMF, the Server Header will be appended by the NSSF with the value "NSSF-9faf1bbc-6e4a-4454-a507-aef01a101a01"
- The values in the above example are samples. Ensure that you update the values of the following parameters according to your deployment:
  - nfType must be NSSF.
  - errorCodeSeriesId: A valid configured value
  - nfInstanceId: Valid NSSF's instance value. It must be same as NSSF's instance ID.
  - Ensure that an errorCodeSeries exists corresponding to the errorCodeSeriesId.

Sample cURL to Enable Server Header:

```
curl --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/{nfType}/nf-common-component/v1/{serviceName}/serverheaderdetails -H "Content-Type: application/json" -d '{"enabled":true,"errorCodeSeriesId":"E1","configuration":{"nfType":"NSSF","nfInstanceId":"9faf1bbc-6e4a-4454-a507-aef01a101a01"}}' -v
```

Example of Request or Response Body to Disable Server Header:

```
{
  "enabled": false,
  "errorCodeSeriesId": "E1",
  "configuration": {
    "nfType": "NSSF",
```

```
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01"
  }
}
```

Sample cURL to Disable Server Header:

```
curl --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/{nfType}/nf-common-component/v1/{serviceName}/serverheaderdetails -H "Content-Type: application/json" -d '{"enabled":false,"errorCodeSeriesId":"E1","configuration":{"nfType":"NSSF","nfInstanceId":"9faf1bbc-6e4a-4454-a507-aef01a101a01"}}' -v
```

Note

- `enabled` is used to enable or disable the feature.
- `nfType` and `nfInstanceId` are used to form the Server Header.
- Ensure that an `errorCodeSeries` exists corresponding to the `errorCodeSeriesId`.

5.1.2 Configurations to enable Ingress Gateway Pod Protection

This API is used to enable and configure Ingress Gateway Pod Protection Feature in NSSF. To enable the feature, REST API needs to be invoked and update the enabled flag to `true`.

URI:

/NSSF/nf-common-component/v1/{serviceName}/podprotection

Method: GET, PUT

Note

Although NSSF does not fully support the PATCH method, it has been partially implemented. The responsibility lies with the operator to abstain from using PATCH and to adhere strictly to the supported methods outlined in the documentation.

Content-Type: application/json

Request or Response Body Parameters

Table 5-2 Request or Response Body Parameters

Field Name	Description	Details
`enabled`	This attribute helps enable or disable the "Pod Protection Defense" feature. Setting this attribute to true triggers resource monitoring in the next schedule.	**Data Type**: Boolean **Range**: true or false **Example Value**: false
`monitoringInterval`	This is the interval (in milliseconds) at which the state of the resource will be checked and updated accordingly.	**Data Type**: integer **Range**: minimum 0 **Example Value**: 0

Table 5-2 (Cont.) Request or Response Body Parameters

Field Name	Description	Details
`congestionControl.enabled`	This attribute helps enable or disable the `congestionControl` mechanism from the next schedule.	**Data Type**: Boolean **Range**: true or false **Example Value**: false
`congestionControl.stateChangeSampleCount`	The number of times the pod has to remain in the same state in order to transition to a different state. For example, if the pod is in the normal state (weight 0) and in order to transition from normal to DOC, the pod has to remain in the DOC state for the specified count or time.	**Data Type**: integer **Range**: minimum 0 **Example Value**: 0
`congestionControl.actionSamplingPeriod`	The `isSchedulerTimeSensitiveAction` action will be executed only at the specified time, `actionSamplingPeriod * monitoringInterval`.	**Data Type**: integer **Range**: minimum 0 **Example Value**: 0
`congestionControl.states.name`	The attribute is used to define different states of congestion.	**Data Type**: String **Range**: Normal, DoC, Congested **Example Value**: NA
`congestionControl.states.weight`	Each state is identified by its weight, which defines the level of criticality. The lower the weight, the lower the criticality. For example, the resource threshold for weight 0 (Normal) should be lower than the resource threshold for weight 1 (DOC).	**Data Type**: integer **Range**: minimum 0 **Example Value**: 0
`congestionControl.actionSamplingPeriod.resourceThreshold`	The CPU and pending message thresholds for a given state are listed within this array. If the resource consumption exceeds the listed threshold, the corresponding entry action is triggered.	**Data Type**: array **Range**: NA **Example Value**: NA
`incrementByActionSamplingPeriod`	The value of `incrementBy` increases at a specific interval, which is determined by `incrementByActionSamplingPeriod * monitoringInterval`. If the value is not specified, the `actionSamplingPeriod` value will be used as the default.	**Data Type**: integer **Range**: minimum 0 **Example Value**: 0
`decrementByActionSamplingPeriod`	The value of `decrementBy` decreases at a specific interval, which is determined by `decrementByActionSamplingPeriod * monitoringInterval`. If the value is not specified, the `actionSamplingPeriod` value will be used as the default.	**Data Type**: integer **Range**: minimum 0 **Example Value**: 0

Example

Example of Request or Response Body:

> **Note**
>
> - For an ASM setup, it is recommend to set the threshold of `pendingMessage` at DoC state to 500 and at Congested state to 600.
> - As of now, NSSF does not support memory threshold.

```
{
        "enabled": false,
        "monitoringInterval": 100,
        "congestionControl": {
            "enabled": false,
            "stateChangeSampleCount": 10,
            "actionSamplingPeriod": 3,
            "states": [
                {
                    "name": "Normal",
                    "weight": 0,
                    "entryAction": [
                        {
                            "action": "MaxConcurrentStreamsUpdate",
                            "arguments": {
                                "incrementBy": 30,
                                "incrementByActionSamplingPeriod": 3,
                                "maxConcurrentStreamsPerCon": 100
                            }
                        },
                        {
                            "action": "AcceptIncomingConnections",
                            "arguments": {
                                "accept": true
                            }
                        }
                    ]
                },
                {
                    "name": "DoC",
                    "weight": 1,
                    "resourceThreshold": {
                        "cpu": 60,
                        "pendingMessage": 900
                    },
                    "entryAction": [
                        {
                            "action": "AcceptIncomingConnections",
                            "arguments": {
                                "accept": false
                            }
                        },
                        {
                            "action": "MaxConcurrentStreamsUpdate",
                            "arguments": {
                                "incrementBy": 30,
                                "incrementByActionSamplingPeriod": 3,
```

```
                                "decrementBy": 30,
                                "decrementByActionSamplingPeriod": 1,
                                "maxConcurrentStreamsPerCon": 50
                            }
                        }
                    ]
                },
                {
                    "name": "Congested",
                    "weight": 2,
                    "resourceThreshold": {
                        "cpu": 75,
                        "pendingMessage": 1000
                    },
                    "entryAction": [
                        {
                            "action": "AcceptIncomingConnections",
                            "arguments": {
                                "accept": false
                            }
                        },
                        {
                            "action": "MaxConcurrentStreamsUpdate",
                            "arguments": {
                                "decrementBy": 30,
                                "decrementByActionSamplingPeriod": 1,
                                "maxConcurrentStreamsPerCon": 5
                            }
                        }
                    ]
                }
            ]
        }
}
```

5.1.3 Error Code Profile Configuration in Ingress Gateway

This URI can be used to update the errorCodeProfiles that is used in Overload Control feature for populating details in error responses when a request is discarded. By default, the `errorCodeProfiles` remains null.

URI:

{apiRoot}/nssf/nf-common-component/v1/igw/errorcodeprofiles

Method:

- PUT: Update Error Code configuration of the required service.

Content-Type: application/json

Dependency

`errorcodeprofiles` is used in `ocdiscardpolicies` to define how different overload levels trigger rejection with specific errors.

Request or Response Body Parameters

Field Name	Description	Details
name	This is a mandatory parameter. Error name.	**Data Type:** string
errorCause	This is an optional parameter. errorCause field in an errorScenario determines the error cause that needs to be populated in ProblemDetails (Cause field) response from IGW when the exception occurred at IGW matches the configured errorScenario's exceptionType parameter.	**Data Type:** string
errorCode	This is a mandatory parameter. errorCode field in an errorScenario determines the HttpStatusCode that needs to be populated in ProblemDetails (HttpStatus field) response from IGW when the exception occurred at IGW matches the configured errorScenario's exceptionType field.	**Data Type:** integer
errorDescription	This is an optional parameter. errorDescription field in an errorScenario determines the description that needs to be populated in ProblemDetails (Detail field) response from IGW when the exception occurred at IGW matches the configured errorScenario's exceptionType field.	**Data Type:** string
errorTitle	This is an optional parameter. errorTitle field in an errorScenario determines the title that needs to be populated in ProblemDetails (Title field) response from IGW when the exception occurred at IGW matches the configured errorScenario's exceptionType parameter.	**Data Type:** string
redirectURL	This is an optional parameter. redirectUrl field in an errorScenario determines the redirection URL, this value is populated in LOCATION header while sending response from IGW. The header is populated only when the exception occurred at IGW matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the redirectUrl field for the particular errorScenario is configured appropriately.	**Data Type:** string

Field Name	Description	Details
retry-after	This is an optional parameter. retryAfter field in an errorScenario determines the value in seconds or particular date after which the service should be retried, this value is populated in retry-after header while sending response from IGW. The header is populated only when the exception occurred at IGW matches the configured errorScenario's exceptionType parameter, the errorCode configured for the particular errorScenario lies in 3xx error series and the retryAfter field for the particular errorScenario is configured appropriately in seconds.	**Data Type:** string

Example

Example of Request or Response Body:

```
[
    {
        "name": "error429",
        "errorCode": 429,
        "errorCause": "Too many requests",
        "errorTitle": "Too many requests",
        "redirectURL": "",
        "retry-after": "",
        "errorDescription": "Too many requests"
    },
    {
        "name": "error503",
        "errorCode": 503,
        "errorCause": "Backend not able to handle traffic",
        "errorTitle": "Backend not able to handle traffic",
        "redirectURL": "",
        "retry-after": "",
        "errorDescription": "Backend not able to handle traffic"
    }
]
```

Sample cURL:

```
curl-i --http2-prior-knowledge-X PUT 'http://ocnssf-nsconfig:{{port}}/nssf/nf-
common-component/v1/igw/errorcodeprofiles'-H 'Content-Type: application/json'
--data-raw '[{"name":"error429","errorCode":429,"errorCause":"Too many
requests","errorTitle":"Too many requests","redirectURL":"","retry-
after":"","errorDescription":"Too many requests"},
{"name":"error503","errorCode":503,"errorCause":"Backend not able to handle
traffic","errorTitle":"Backend not able to handle
traffic","redirectURL":"","retry-after":"","errorDescription":"Backend not
able to handle traffic"}]'
```

5.1.4 Discard Policy Configuration in Ingress Gateway

This URI can be used to update discard policies that will be used in overload control to select the appropriate policy from the configured list based on the load level of a particular service. By default, `ocDicardPolicies` is null.

URI:

{apiRoot}/nssf/nf-common-component/v1/igw/ocdiscardpolicies

- PUT: Update discard policy configuration of the required service.

Content-Type: application/json

Dependency

- `ocdiscardpolicies` use `errorcodeprofiles` to decide the error message when rejecting requests.
- `ocdiscardpolicies` are used by `ocpolicymapping` to associate services with specific overload handling policies.

Request or Response Body Parameters

Field Name	Description	Details
`name`	This is a mandatory parameter. Name of the discarded policy. **Note**: name must be the value configured in `policyName` under `ocpolicymapping`.	**Data Type:** string
`policies.action`	This is an optional parameter. Defines the action to be taken on selected requests rejection based on error code.	**Data Type:** string **Example Value:** `RejectWithErrorCode`
`policies.errorCodeProfile`	This is an optional parameter. The error code profiles.	**Data Type:** string
`policies.level`	This is an optional parameter. Defines the overload level.	**Data Type:** string
`policies.value`	This is a mandatory parameter. Value of priority above which requests are considered as potential candidates for drop. Percentage of requests to drop in the current sampling period over the calculated rate in the previous sampling period.	**Data Type:** integer
`scheme`	This is a mandatory parameter. Discarded policy scheme based on percentage.	**Data Type:** string

Example

Example of Request or Response Body:

```
[
    {
        "name": "nsselectionPolicy",
        "scheme": "PercentageBased",
        "policies": [
            {
                "level": "Warning",
                "value": 0,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error429"
            },
            {
                "level": "Minor",
                "value": 10,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error429"
            },
            {
                "level": "Major",
                "value": 25,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error429"
            },
            {
                "level": "Critical",
                "value": 50,
```

```
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error503"
            }
        ]
    },
    {
        "name": "nsavailabilityPolicy",
        "scheme": "PercentageBased",
        "policies": [
            {
                "level": "Warning",
                "value": 0,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error429"
            },
            {
                "level": "Minor",
                "value": 10,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error429"
            },
            {
                "level": "Major",
                "value": 25,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error429"
            },
            {
```

```
                "level": "Critical",
                "value": 50,
                "action": "RejectWithErrorCode",
                "errorCodeProfile": "error503"
            }
        ]
    }
]
```

Sample cURL:

```
curl -i --http2-prior-knowledge -X PUT 'http://ocnssf-nsconfig:{{port}}/
nssf/nf-common-component/v1/igw/ocdiscardpolicies' -H 'Content-Type:
application/json' --data-raw
'[{"name":"nsselectionPolicy","scheme":"PercentageBased","policies":
[{"level":"Warning","value":0,"action":"RejectWithErrorCode","errorCodeProfile
":"error429"},
{"level":"Minor","value":10,"action":"RejectWithErrorCode","errorCodeProfile":
"error429"},
{"level":"Major","value":25,"action":"RejectWithErrorCode","errorCodeProfile":
"error429"},
{"level":"Critical","value":50,"action":"RejectWithErrorCode","errorCodeProfil
e":"error503"}]},
{"name":"nsavailabilityPolicy","scheme":"PercentageBased","policies":
[{"level":"Warning","value":0,"action":"RejectWithErrorCode","errorCodeProfile
":"error429"},
{"level":"Minor","value":10,"action":"RejectWithErrorCode","errorCodeProfile":
"error429"},
{"level":"Major","value":25,"action":"RejectWithErrorCode","errorCodeProfile":
"error429"},
{"level":"Critical","value":50,"action":"RejectWithErrorCode","errorCodeProfil
e":"error503"}]}]'
```

5.1.5 Policy Mapping Configuration to Enable Overload Configuration

This URI can be used to update service names and corresponding policy names for the service which is mapped to "ocDiscardPolicies" based on "policyName" and also to enable or disable the Overload Control feature and the sampling period in overload control. By default, the Overload Control feature is disabled and the sampling period is 6000. To enable the feature, REST API needs to be invoked and update the enabled flag to `true`.

URI:

{apiRoot}/nssf/nf-common-component/v1/igw/ocpolicymapping

Method:

- GET: Get Policy mapping value of the required service

- PUT: Update the Policy mapping value of the required service
- PACTH: Update specific Policy mapping value of the required service

Content-Type: application/json

Dependency

Depends on `ocdiscardpolicies` to apply the right rejection policy to each service.

Request or Response Body Parameters

Table 5-3 Request or Response Body Parameters

Field Name	Description	Details
`enabled`	This is a mandatory parameter. To enable or disable the Overload Control feature. Set values to `true` or `false`, respectively.	**Data Type:** boolean
`mappings`	This is an optional parameter. **Note**: A value for `ocPolicyMapping.mappings` is required when `ocPolicyMapping.enabled` is set to `true`. If `ocPolicyMapping.mappings` is empty, the Overload Control feature will behave as if it is disabled.	**Data Type:** Array
`mappings.policyName`	This is a conditional parameter. The discard policy entry determines a mapping between the service and discards policy name per service. **Note**: If a value for `mappings.svcName` is provided, then `mappings.policyName` is mandatory.	**Data Type:** string
`mappings.svcName`	This is a conditional parameter. The service entry to determine a mapping between service and discard policy name per `service.svcName` must be added in the following format: <deployment-name>-<servicename> **Note**: servicename is fixed and cannot be changed. **Note**: If a value for `mappings.policyName` is provided, then `mappings.svcName` is mandatory.	**Data Type:** string
`samplingPeriod`	This is a mandatory parameter. The time frame for each cycle of Overload Control per service. Its value is in milliseconds.	**Data Type:** integer

Example

Example of Request or Response Body:

```
{
    "enabled": true,
    "mappings": [
```

```
        {
            "svcName": "ocnssf-nsselection",
            "policyName": "nsselectionPolicy"
        },
        {
            "svcName": "ocnssf-nsavailability",
            "policyName": "nsavailabilityPolicy"
        }
    ],
    "samplingPeriod": 6000
}
```

Sample cURL:

```
curl -i --http2-prior-knowledge -X PUT 'http://ocnssf-nsconfig:{{port}}/
nssf/nf-common-component/v1/igw/ocpolicymapping' -H 'Content-Type:
application/json' --data-raw '{"enabled": true, "mappings": [{"svcName":
"ocnssf-nsselection", "policyName": "nsselectionPolicy"},{"svcName": "ocnssf-
nsavailability","policyName": "nsavailabilityPolicy"}],"samplingPeriod":
6000}'
```

5.1.6 Error Code Series Configuration in Ingress Gateway

This URI can be used to update the `errorcodeserieslist` that are used in Overload Control feature and Server Header feature to list the configurable exception or error for an error scenario in Ingress Gateway.

URI:

{apiRoot}/nssf/nf-common-component/v1/igw/errorcodeserieslist

Method: PUT to update Error Code Series configuration of the required service.

Content Type: application/json

Dependency

Not directly linked to other configurations but enhances error handling by classifying errors.

Request or Response Body Parameters

Table 5-4 Request or Response Body Parameters

Parameter	Description	Details
`id`	This is a mandatory parameter. Indicates the error code identifier.	**Data Type**: string
`errorCodeSeries`	This is a mandatory parameter. List the error codes for a specific service. **Note**: "`ErrorCodeSeries`" is configured only if a set of error responses with specific error codes is expected in server header. If it is not configured then all the error responses will have server header.	**Data Type**: string

Table 5-4 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
exceptionList	This is an optional parameter. Lists the configurable exception or error for an error scenario in Ingress Gateway.	**Data Type**: string **Range:** • ConnectionTimeout • RequestTimeout • UnknownHostException • ConnectException • NotFoundException
errorSet	This is a mandatory parameter. Possible values for "errorSet" attribute: 5xx, 4xx, 3xx	**Data Type**: string
errorCodes	This is a mandatory parameter. Possible values include all error codes in the respective HttpSeries value assigned for "errorSet". **Note**: Use single value of "-1" if all error codes in that HttpSeries are to be considered. For example: • If the series is 4XX, the values should be between 400-499. • If the series is 5XX, the values should be between 500-599. Onus is on operator to configure the errorCodes properly as explained above.	**Data Type**: string

Example

Example of Request or Response Body:

```
[
  {
    "id": "E1",
    "exceptionList": [
      "RequestTimeout",
      "ConnectionTimeout",
      "UnknownHostException",
      "NotFoundException"
    ],
    "errorCodeSeries": [
      {
        "errorSet": "4xx",
        "errorCodes": [
          400,
          408
        ]
      },
```

```
      {
        "errorSet": "5xx",
        "errorCodes": [
          500,
          503
        ]
      }
    ]
  }
]
```

Example with recommended configuration to enable Server Headers for all 5xx and 4xx error codes:

```
[
    {
        "id": "E1",
        "errorCodeSeries":
        [
            {
                "errorSet": "4xx",
                "errorCodes": [-1]
            },
            {
                "errorSet": "5xx",
                "errorCodes": [-1]
            }
        ]
    }
]
```

Sample cURL:

```
curl --http2-prior-knowledge -X PUT http://{{host}}:{{port}}/{nfType}/nf-common-component/v1/{serviceName}/{instanceId}/errorcodeserieslist -H "Content-Type: application/json" -d '[{"id":"E1","exceptionList":["RequestTimeout","ConnectionTimeout","UnknownHostException","NotFoundException"],"errorCodeSeries":[{"errorSet":"4xx","errorCodes":[400,429]},{"errorSet":"5xx","errorCodes":[500,503]}]}]' -v
```

5.1.7 Routes Configuration in Ingress Gateway

The configuration of "routesconfiguration" is required for Server Header and Overload control feature to map route ID and its corresponding route-level configuration. By default, this configuration is null.

URI:

/{nfType}/nf-common-component/v1/{serviceName}/routesconfiguration

> **Note**
>
> Ensure that route ID (`routesConfig[0].id`) used in REST configuration is same as the route ID used in `values.yaml` file.

Method: GET, PUT, PATCH

Content-Type: application/json

Request or Response Body Parameters

Dependency

`routesconfiguration` defines routing behaviors and associates services with error code series. It references `errorCodeSeriesId` which is defined in `id` attribute in `errorcodeserieslist`.

Table 5-5 Request or Response Body Parameters

Parameter	Description	Details
`id`	This is a mandatory parameter. Specifies the route IDs for which you need to define server header	**Data Type**: string `Range`: NA **Example Value**: "availability_mapping"
`serverHeaderDetails.enabled`	This is an optional parameter. Set the value for this parameter to true if you want to define server header at the route level.	**Data Type**: boolean **Range**: true or false **Example Value**: true
`serverHeaderDetails.errorCodeSeriesId`	This is an optional parameter. Specifies the error list IDs **Note**: Ensure that an `errorCodeSeries` exists corresponding to the `errorCodeSeriesId`	**Data Type**: String **Range**: NA **Example Value**: NA

Example

Example of Request or Response Body

```
[
  {
    "id": "availability_mapping",
    "serverHeaderDetails": {
      "enabled": true,
      "errorCodeSeriesId": "E2"
    }
  },
  {
    "id": "nsselection_mapping",
    "serverHeaderDetails": {
      "enabled": false,
      "errorCodeSeriesId": "E2"
```

```
    }
  }
]
```

Sample cURL:

```
curl --http2-prior-knowledge -X PUT http://ocnssf-nsconfig:8080/{nfType}/nf-common-component/v1/igw/routesconfiguration -H "Content-Type: application/json" -d '[ { "id": "availability_mapping", "serverHeaderDetails": { "enabled": true, "errorCodeSeriesId": "E2" } }, { "id": "nsselection_mapping", "serverHeaderDetails": { "enabled": false, "errorCodeSeriesId": "E2" } } ]' -v
```

5.1.8 Perf-Info REST APIs

This section explains REST API configurations required at Perf-Info to enable Overload control feature:

5.1.8.1 Overload Level Threshold Configuration in Perf-Info

URI:

{apiRoot}/nssf/nf-common-component/v1/perf-info/overloadLevelThreshold

Method:

- GET: Get the Overload Threshold Value of the required service (Backend service).
- PUT: Update the Overload Threshold Value of the required service (Backend service).
- DELETE: Delete the Overload Threshold Value of the required service (Backend service).

Content-Type: application/json

Dependency

Determines when a service is considered overloaded. It triggers `ocdiscardpolicies` when thresholds are exceeded, causing requests to be rejected with predefined error responses.

Request or Response Body Parameters

Table 5-6 Request or Response Body Parameters

Parameter	Description	Details
`svcName`	This is a mandatory parameter. Name of the backend service.	**Data Type:** string
`metricsThresholdList.metricsName`	This is a mandatory parameter. Name of overload indicator such as `cpu`, `svc_failure_count`, `svc_pending_count`, and `memory`.	**Data Type:** string
`metricsThresholdList.levelThresholdList.onsetValue`	This is a mandatory parameter. The `metricsThresholdList.levelThresholdList.level` is considered to be breached if the `metricsThresholdList.metricsName` value has crossed this onset value.	**Data Type:** integer

Table 5-6 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
`metricsThresholdList.levelThresholdList.level`	This is a mandatory parameter. Specifies the name of the level. The name specified in this parameter must match the level name in Ingress Gateway's `ocdiscardpolicies`.	**Data Type:** string
`metricsThresholdList.levelThresholdList.abatementValue`	This is a mandatory parameter. The overload condition is considered as cleared this level, if the `metricsThresholdList.metricsName` is below the abatement value.	**Data Type:** integer
`metricsThresholdList.levelThresholdList`	This is a mandatory parameter. List of threshold values.	**Data Type:** array
`metricsThresholdList`	This is a mandatory parameter. List of criteria used to calculate the load level.	**Data Type:** array

Example

Example of Request or Response Body:

```
[
  {
    "svcName": "ocnssf-nsselection",
    "metricsThresholdList": [
      {
        "metricsName": "svc_failure_count",
        "levelThresholdList": [
          {
            "level": "Warning",
            "onsetValue": 250,
            "abatementValue": 200
          },
          {
            "level": "Minor",
            "onsetValue": 400,
            "abatementValue": 350
          },
          {
            "level": "Major",
            "onsetValue": 600,
            "abatementValue": 480
          },
          {
            "level": "Critical",
            "onsetValue": 900,
            "abatementValue": 720
          }
        ]
      },
      {
        "metricsName": "memory",
```

```
    "levelThresholdList": [
      {
        "level": "Warning",
        "onsetValue": 87,
        "abatementValue": 85
      },
      {
        "level": "Minor",
        "onsetValue": 90,
        "abatementValue": 87
      },
      {
        "level": "Major",
        "onsetValue": 95,
        "abatementValue": 90
      },
      {
        "level": "Critical",
        "onsetValue": 99,
        "abatementValue": 95
      }
    ]
  },
  {
    "metricsName": "cpu",
    "levelThresholdList": [
      {
        "level": "Warning",
        "onsetValue": 80,
        "abatementValue": 75
      },
      {
        "level": "Minor",
        "onsetValue": 85,
        "abatementValue": 80
      },
      {
        "level": "Major",
        "onsetValue": 90,
        "abatementValue": 85
      },
      {
        "level": "Critical",
        "onsetValue": 95,
        "abatementValue": 90
      }
    ]
  },
  {
    "metricsName": "svc_pending_count",
    "levelThresholdList": [
      {
        "level": "Warning",
        "onsetValue": 100,
        "abatementValue": 60
      },
```

```
            {
              "level": "Minor",
              "onsetValue": 200,
              "abatementValue": 125
            },
            {
              "level": "Major",
              "onsetValue": 260,
              "abatementValue": 201
            },
            {
              "level": "Critical",
              "onsetValue": 400,
              "abatementValue": 300
            }
          ]
        }
      ]
    },
    {
      "svcName": "ocnssf-nsavailability",
      "metricsThresholdList": [
        {
          "metricsName": "svc_failure_count",
          "levelThresholdList": [
            {
              "level": "Warning",
              "onsetValue": 65,
              "abatementValue": 60
            },
            {
              "level": "Minor",
              "onsetValue": 130,
              "abatementValue": 120
            },
            {
              "level": "Major",
              "onsetValue": 260,
              "abatementValue": 240
            },
            {
              "level": "Critical",
              "onsetValue": 520,
              "abatementValue": 480
            }
          ]
        },
        {
          "metricsName": "memory",
          "levelThresholdList": [
            {
              "level": "Warning",
              "onsetValue": 60,
              "abatementValue": 55
            },
            {
```

```
            "level": "Minor",
            "onsetValue": 65,
            "abatementValue": 60
          },
          {
            "level": "Major",
            "onsetValue": 70,
            "abatementValue": 65
          },
          {
            "level": "Critical",
            "onsetValue": 75,
            "abatementValue": 70
          }
        ]
      },
      {
        "metricsName": "cpu",
        "levelThresholdList": [
          {
            "level": "Warning",
            "onsetValue": 60,
            "abatementValue": 55
          },
          {
            "level": "Minor",
            "onsetValue": 65,
            "abatementValue": 60
          },
          {
            "level": "Major",
            "onsetValue": 70,
            "abatementValue": 65
          },
          {
            "level": "Critical",
            "onsetValue": 75,
            "abatementValue": 70
          }
        ]
      },
      {
        "metricsName": "svc_pending_count",
        "levelThresholdList": [
          {
            "level": "Warning",
            "onsetValue": 3,
            "abatementValue": 1
          },
          {
            "level": "Minor",
            "onsetValue": 6,
            "abatementValue": 3
          },
          {
            "level": "Major",
```

```
            "onsetValue": 12,
            "abatementValue": 6
          },
          {
            "level": "Critical",
            "onsetValue": 24,
            "abatementValue": 12
          }
        ]
      }
    ]
  }
]
```

Sample cURL:

```
curl -i --http2-prior-knowledge -X GET 'http://ocnssf-nsconfig:{{port}}/
nssf/nf-common-component/v1/perf-info/overloadLevelThreshold'
```

5.1.9 Configuration To Check If Overload Control in Enabled

This section explains REST API to check if overload control is enabled:

5.1.9.1 To Check Current Load Level

This API is used to check current load level on the NSSF services.

API:

http://ocnssf-ingress-gateway:80/igw/load-level?svcName=ocnssf-nsselection

http://ocnssf-ingress-gateway:80/igw/load-level?svcName=ocnssf-nsavailability

Method: GET

Content Type: application/json

cURL Command:

NsSelection:

```
curl -i --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:
{{port}}/igw/load-level?svcName=ocnssf-nsselection
```

NsAvailability:

```
curl -i --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:
{{port}}/igw/load-level?svcName=ocnssf-nsavailability
```

Response: `Normal`

5.1.9.2 To Check Pending Count

This API is used to check pending counts of requests with the NSSF services.

API:

http://ocnssf-ingress-gateway:80/igw/pending-req-count?svcName=ocnssf-nsselection

http://ocnssf-ingress-gateway:80/igw/pending-req-count?svcName=ocnssf-nsavailability

Method: GET

Content Type: application/json

cURL Command:

NsSelection:

```
curl -i --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:
{{port}}/igw/pending-req-count?svcName=ocnssf-nsselection
```

NsAvailability:

```
curl -i --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:
{{port}}/igw/pending-req-count?svcName=ocnssf-nsavailability
```

Response:

```
{
  "svc_pending_count": [
    {
      "svcName": "ocnssf-nsselection",
      "count": 52
    }
  ]
}
```

5.1.9.3 To Check Failure Count

This API is used to check failure counts of requests with the NSSF services.

API:

http://ocnssf-ingress-gateway:80/igw/failed-req-count?svcName=ocnssf-nsselection

http://ocnssf-ingress-gateway:80/igw/failed-req-count?svcName=ocnssf-nsavailability

Method: GET

Content Type: application/json

cURL Command:

NsSelection:

```
curl -i --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:
{{port}}/igw/failed-req-count?svcName=ocnssf-nsselection
```

NsAvailability:

```
curl -i --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:
{{port}}/igw/failed-req-count?svcName=ocnssf-nsavailability
```

Response:

```
{
  "svc_failure_count": [
    {
      "svcName": "ocnssf-nsselection",
      "count": 7780
    }
  ]
}
```

5.1.9.4 To Check NF_CONGESTION_RISK for NsAvailability

This API is used to check the risk of congestion on NsAvailability service.

API:

http://ocnssf-ingress-gateway:80/nnssf-nssaiavailability/v1/nssai-availability/1

Method: PUT

Content Type: application/json

cURL Command:

NsAvailability:

```
curl -v --http2-prior-knowledge -X PUT http://ocnssf-ingress-gateway:{{port}}/
nnssf-nssaiavailability/v1/nssai-availability/12345678-abcd-efAB-
CDEF-123456789015 -H 'Content-Type: application/json' -d
'{ "supportedNssaiAvailabilityData": [ { "tai": { "plmnId": { "mcc": "100",
"mnc": "101" }, "tac": "323052" }, "supportedSnssaiList": [ { "sd": "EABB05",
"sst": 5 }, { "sd": "EABB04", "sst": 4 } ] } ] }'
```

Response:

```
{
  "type": null,
  "title": "NF_CONGESTION_RISK",
  "status": 429,
  "detail": "Too many requests",
  "instance": null,
  "cause": "NF_CONGESTION_RISK",
  "invalidParams": null
}
```

5.1.9.5 To Check NF_CONGESTION_RISK for NsSelection

This API is used to check the risk of congestion on NsSelection service.

API:

http://ocnssf-ingress-gateway:80/nnssf-nsselection/v2/network-slice-information

Method: GET

Content Type: application/json

cURL Command:

NsSelection:

```
curl -v --http2-prior-knowledge -X GET http://ocnssf-ingress-gateway:{{port}}/
nnssf-nsselection/v2/network-slice-information?nf-id=12345678-abcd-efAB-
CDEF-123456789012&nf-
type=AMF&tai=%7B%0A%09%22plmnId%22%3A%0A%09%09%7B%0A%09%09%09%22mcc%22%3A%2210
0%22%2C%0A%09%09%09%22mnc%22%3A%22101%22%0A%09%09%7D%2C%0A%09%22tac%22%3A%2210
0001%22%0A%7D&slice-info-request-for-pdu-
session=%7B%0A%22sNssai%22%3A%20%0A%09%09%7B%0A%09%09%09%22sst%22%3A%221765675
7%22%2C%0A%09%09%09%22sd%22%3A%22EABB01%22%0A%09%09%7D%2C%0A%09%22roamingIndic
ation%22%3A%20%22NON_ROAMING%22%0A%7D
```

Response:

```
{
  "type": null,
  "title": "NF_CONGESTION_RISK",
  "status": 429,
  "detail": "Too many requests",
  "instance": null,
  "cause": "NF_CONGESTION_RISK",
  "invalidParams": null
}
```

5.2 Egress Gateway REST APIs

This section explains REST API configurations required at Egress Gateway for various features.

5.2.1 Peer Configuration

This resource is used in `SBIRouting` feature. This URI can be used to add or update the list of peers wherein each peer consists of ID, host, port or virtualHost, and apiPrefix. The ID of each peer is mapped to `peerIdentifier` in "`peersetconfiguration`" resource. The default value is null.

URI:

{apiRoot}/nssf/nf-common-component/v1/egw/peerconfiguration

Method supported: PUT, PATCH, GET

Resource: array (peerConfiguration)

Table 5-7 PeerConfiguration

Parameter	Data Type	Constraints	Description
`id`	string	Unique value in peer configuration.	Identifier for Peer

Table 5-7 (Cont.) PeerConfiguration

Parameter	Data Type	Constraints	Description
host	string	NA	Host details of the Peer. In case of SLF, it is SCP. It can be IPv4, IPv6, and FQDN details.
virtualHost	string	NA	FQDN details of the peer. This FQDN is sent to Alternate Route Service for DNS-SRV resolution.
port	string	NA	Port details for Peer.
apiPrefix	string	Keep the value as / only for NRF	API prefix details for Peer.
healthApiPath	string	Valid path	Parameter to support SCP health check API. It contains path of the health API.

Dependency: It is mandatory to configure `peerconfiguration` with `healthApiPath` if you want to enable `peermonitoringconfiguration`.

Example:

```
 curl -v -X PUT "http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/
peerconfiguration" -H  "Content-Type: application/json" -d peer.json

Sample peer.json:-
[
  {
    "id": "peer1",
    "host": "scp1",
    "port": "8080",
    "apiPrefix": "/",
    "healthApiPath": "/health/v1"
  },
  {
    "id": "peer2",
    "host": "scp2",
    "port": "8080",
    "apiPrefix": "/",
    "healthApiPath": "/health/v2"
  },
  {
    "id": "peer3",
    "host": "scp3",
    "port": "8080",
    "apiPrefix": "/",
    "healthApiPath": "/health/v3"
  },
  {
    "id": "peer4",
    "host": "scp4",
    "port": "8080",
    "apiPrefix": "/",
```

```
    "healthApiPath": "/health/v4"
  },
  {
    "id": "peer5",
    "virtualHost": "xyz.test.com",
    "apiPrefix": "/",
    "healthApiPath": "/health/v5"
  },
  {
    "id": "peer6",
    "virtualHost": "abc.test.com",
    "apiPrefix": "/",
    "healthApiPath": "/health/v6"
  }
]
```

5.2.2 Peer Set Configuration

This resource is used in `SBIRouting` feature. This URI can be used to add or update the list of peer sets wherein each peer set consists of ID and list of http or https instances. Each instance consists of priority and peer identifier that is mapped to `id` in "`peerconfiguration`" resource. The `id` of each peer set is mapped to `peerSetIdentifier` in `routesconfiguration` resource. The default value is null.

API:

nssf/nf-common-component/v1/egw/peersetconfiguration

Method supported: PUT, PATCH, GET

Resource: array (peerSetConfiguration)

Table 5-8 PeerSetConfiguration

Parameter	Data Type	Constraints	Description
`id`	string	Unique value in peerset configuration.	Identifier for peer set.
`httpConfiguration`	array(PeerIdentifierConfiguration)	Both http and https need to be configured irrespective of which scheme is supported.	Configuration for HTTP based peers. This value will be selected if 3GPPAPIRootScheme value is http.
`httpsConfiguration`	array(PeerIdentifierConfiguration)	Both http and https need to be configured irrespective of which scheme is supported.	Configuration for HTTPs based Peers. This value will be selected if 3GPPAPIRootScheme value is https.

Table 5-9 Peer Identifier Configuration

Parameter	Data Type	Constraints	Description
`priority`	integer	Priority should be unique	Priority of peer to be used in a peer set.

Table 5-9 (Cont.) Peer Identifier Configuration

Parameter	Data Type	Constraints	Description
peerIdentifier	string	NA	Peer Identifier is the value of Peer configured during PeerConfiguration

Dependency: `PeerConfiguration` must be done before adding `PeerSetConfiguration`.

Example:

```
curl -v -X PUT "http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/
peersetconfiguration" -H  "Content-Type: application/json" -d @peerset.json

sample peerset.json
[
  {
    "id": "set0",
    "httpConfiguration": [
      {
        "priority": 1,
        "peerIdentifier": "peer1"
      },
      {
        "priority": 2,
        "peerIdentifier": "peer2"
      },
      {
        "priority": 3,
        "peerIdentifier": "peer3"
      },
      {
        "priority": 4,
        "peerIdentifier": "peer4"
      }
    ],
    "httpsConfiguration": [
      {
        "priority": 1,
        "peerIdentifier": "peer1"
      },
      {
        "priority": 2,
        "peerIdentifier": "peer2"
      },
      {
        "priority": 3,
        "peerIdentifier": "peer3"
      },
      {
        "priority": 4,
        "peerIdentifier": "peer4"
      }
    ]
```

```
  },
  {
    "id": "set1",
    "httpConfiguration": [
      {
        "priority": 1,
        "peerIdentifier": "peer5"
      }
    ],
    "httpsConfiguration": [
      {
        "priority": 1,
        "peerIdentifier": "peer6"
      }
    ]
  }
]
```

5.2.3 Error Criteria Sets

API: *nssf/nf-common-component/v1/egw/sbiroutingerrorcriteriasets*

Resource: array (sbiroutingerrorcriteriasets)

Table 5-10 Error Criteria Sets

Parameter	Data Type	Constraints	Description
`id`	string	Unique value of Route.	Unique `id` for a `sbiRoutingErrorCriteriaSet`
`method`	array	GET, POST, PUT, DELETE, PATCH	The type of methods for which the re-route need to be attempted.
`exceptions`	array string	NA	Specific exceptions for which reroute or retry will be triggered.
`statusSeries`	string	4xx/5xx	Http `statusSeries` for which reroute or retry will be triggered when we get error response from downstream.
`status`	NA	401/404 or -1	Specific HTTP statuses that belongs to above mentioned `statusSeries` for which reroute/retry has to be triggered. To enable retry or reroute for all the HTTP status belonging to a `statusSeries`, configure this as -1.

Example:

```
curl -v -X PUT "http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/
sbiroutingerrorcriteriasets" -H  "Content-Type: application/json" -d
```

```
sample body.json:-
[
  {
    "id": "criteria_0",
    "method": [
      "GET",
      "POST",
      "PUT",
      "DELETE",
      "PATCH"
    ],
    "exceptions": [
      "java.util.concurrent.TimeoutException",
      "java.net.UnknownHostException"
    ]
  },
  {
    "id": "criteria_1",
    "method": [
      "GET",
      "POST",
      "PUT",
      "DELETE",
      "PATCH"
    ],
    "response": {
      "statuses": [
        {
          "statusSeries": "4xx",
          "status": [
            400,
            404
          ]
        },
        {
          "statusSeries": "5xx",
          "status": [
            500,
            503
          ]
        }
      ]
    }
  }
]
```

5.2.4 Error Action Sets

API: *nssf/nf-common-component/v1/egw/sbiroutingerroractionsets*

Resource: array (sbiroutingerroractionsets)

Table 5-11 Error Action Sets

Parameter	Data Type	Constraints	Description
id	string	Unique value of Route	Unique id for sbiRoutingErrorActionSet.
action	string	reroute, retry	Action that needs to be taken when specific criteria set is matched. Currently we support only 2 values. reroute or retry
attempts	integer	NA	Maximum no of retries to either same or different peer in case of error or failures from backend.
enabled	boolean	true,false	This flag enables the peer blocklist feature using the server headers received in the response.
duration	integer	seconds	The duration for which the peer will be blocked and no traffic will be routed to that peer for this period.

Example:

```
curl -v -X PUT "http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/
sbiroutingerroractionsets" -H  "Content-Type: application/json" -d

sample body.json:-
[
  {
    "id": "action_0",
    "action": "reroute",
    "attempts": 3,
    "blacklist": {
      "enabled": false,
      "duration": 60000
    }
  },
  {
    "id": "action_1",
    "action": "reroute",
    "attempts": 3,
    "blacklist": {
      "enabled": false,
      "duration": 60000
    }
  }
]
```

5.2.5 Routes Configuration

This URI can be used to add or update list of routes. The ID of each route must match the route ID present in Helm chart only if `routeConfigMode` is configured as HELM and `sbiRoutingConfigMode` is configured as REST. The configuration under `sbiRoutingConfiguration` corresponds to the `SBIRouting` specific configuration.

API:

nssf/nf-common-component/v1/egw/routesconfiguration

Resource:: array (RoutesConfiguration)

Table 5-12 RoutesConfiguration

Parameter	Data Type	Constraints	Description
`id`	string	Unique value of Route	Route configuration identifier. **CAUTION**: Default Route with id '`default_route`' is configured automatically. Include this route in the message body while adding new routes using PUT operation. Otherwise it will impact the traffic.
`uri`	string	NA	Provide any dummy URL, or leave the existing URL with existing value.
`order`	integer	NA	Provide the order of the execution of this route.
`sbiRoutingConfiguration`	SbiRoutingConfiguration details. See Description for child values	NA	`"sbiRoutingConfiguration": { "enabled": true, "peerSetIdentifier": "set0" }` Keep `enabled` value to `true` for SBI routing to work. `peerSetIdentifier` is the value configured during `PeerSetConfiguration`. So, this is mapping of route to Peer Set configuration
`httpRuriOnly`	boolean (true,false)	Don't change the value of this Parameter from true. NRF	`true`: Means scheme of URI is changed to http . `false`: Means no change occurred to the scheme.

Table 5-12 (Cont.) RoutesConfiguration

Parameter	Data Type	Constraints	Description
`httpsTargetOnly`	boolean (true,false)	Don't change the value of this Parameter from true. This is required to make SBI routing work for NRF	`true`: Select SBI instances for https list only (if 3gpp sbi target root header is http). `false`: Perform as per provided scheme.
`predicates`	Predicate Structure. See Description	NA	Path predicate details for matching SLF path mapped to this `SBIRoute` rule. **Sample value**: `"predicates": [{ "args": {"pattern": "/nudr-group-id-map/v1/nf-group-ids"}, "name": "Path" }]` All requests with the path indicated in the pattern will be routed to the peers bases on priority. This happens only when `sbiRoutingConfiguration` is enabled. **Note**: Do not change path pattern.
`filters`	Filter Structure. See Description	NA	Filters can be created for various purposes. Use all of the filters as mentioned in example.

Dependency: `Peerconfiguration` and `Peersetconfiguration` must be configured with either set empty list or populated with values before Routes Configuration. These attributes are used for routing only if `sbiRoutingConfiguration` is enabled for a particular route.

Example:

```
curl -v -X PUT "http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/
routesconfiguration" -H  "Content-Type: application/json" -d @header.json

sample header.json:-
[
  {
    "id": "egress_scp_proxy1",
    "uri": "http://localhost:32068/",
    "order": 0,
    "metadata": {
      "httpsTargetOnly": false,
      "httpRuriOnly": false,
      "sbiRoutingEnabled": true
```

```
    },
    "predicates": [
      {
        "args": {
          "pattern": "/notification/amf2/"
        },
        "name": "Path"
      }
    ],
    "filters": [
      {
        "name": "SbiRouting",
        "args": {
          "peerSetIdentifier": "set0",
          "customPeerSelectorEnabled": true,
          "errorHandling": [
            {
              "errorCriteriaSet": "criteria_1",
              "actionSet": "action_1",
              "priority": 1
            },
            {
              "errorCriteriaSet": "criteria_0",
              "actionSet": "action_0",
              "priority": 2
            }
          ]
        }
      }
    ]
  },
  {
    "id": "default_route",
    "uri": "egress://request.uri",
    "order": 100,
    "filters": [
      {
        "name": "DefaultRouteRetry"
      }
    ],
    "predicates": [
      {
        "args": {
          "pattern": "/**"
        },
        "name": "Path"
      }
    ]
  }
]
```

5.2.6 Peer Monitoring Configuration

This URI can be used to enable, disable, and configure peer monitoring.

API :

nssf/nf-common-component/v1/{serviceName}/peermonitoringconfiguration

Method supported: PUT, PATCH, GET

Resource: array (peermonitoringconfiguration)

Table 5-13 peermonitoringconfiguration

Attribute	Data Type	Constraints	Description
`enabled`	boolean	true or false	It is used to enable or disable the peer monitoring feature.
`timeout`	long	300 milliseconds to 10000 milliseconds	Attribute to configure the duration of time after which calls to the SCP health API is timed out.
`frequency`	long	300 milliseconds to 10000 milliseconds	Indicates the frequency or recurring interval at which Egress Gateway initiates health check calls toward SCP.
`failureThreshold`	integer	NA	Indicates the number of failure responses after which a healthy SCP can be marked as unhealthy. Health API call to given SCP shall fail consecutively to these many attempts before it is marked as "Unavailable" from "Available".
`successThreshold`	integer	NA	It indicates the number of successful responses after which an unhealthy SCP can be marked as healthy. Health API call to given SCP shall succeed consecutively to these many attempts before it is marked as "Available" from "Unavailable".

Dependency: It is mandatory to configure `peerconfiguration` with `healthApiPath` if `peermonitoringconfiguration` is enabled. Enable `sbiRoutingConfiguration` in `RoutesConfig` before enabling Peer Monitoring Configuration.

Example:

```
curl -v -X PUT "http://{{host}}:{{port}}/nssf/nf-common-component/v1/
{serviceName}/{instanceId}/peermonitoringconfiguration" -H "Content-Type:
application/json" -d @header.json

sample header.json:-
 {
      "enabled":true,
      "timeout":1000,
      "frequency":2000,
      "failureThreshold":3,
      "successThreshold":3
   }
```

5.2.7 Configurations to Enable or Disable User-Agent Header

To enable the feature, REST API needs to be invoked and the `enabled` flag needs to be updated to `true`.

URI:

/{nfType}/nf-common-component/v1/{serviceName}/useragentheader

Method: GET, PUT, PATCH

Content-Type: application/json

Request or Response Body Parameters

Table 5-14 Request or Response Body Parameters

Parameter	Description	Details
`enabled`	This is a mandatory parameter. This parameter specifies whether the feature is enabled or disabled.	**Data Type**: Boolean **Range**: true or false **Example Value**: false
`nfType`	This is a mandatory parameter. This parameter holds the `nfType` that is used to generate the User-Agent Header.	**Data Type**: String **Range**: NA **Example Value**: NSSF
`nfInstanceId`	This is a mandatory parameter. This parameter represents the UUID of the NSSF deployment that is used to generate the User-Agent Header.	**Data Type**: String **Range**: NA **Example Value**: Valid NSSF Instance ID
`addFqdnToHeader`	This is a mandatory parameter. This parameter specifies if the User-Agent uses the FQDN information under the module to append it when generating the User-Agent Header. The Example Value is set to `false` meaning that the FQDN information is not encoded into the User-Agent Header during its generation.	**Data Type**: Boolean **Range**: true or false **Example Value**: false
`nfFqdn`	This is an optional parameter. This is an optional parameter and can be present or not, if operators want to include the FQDN string configured under this section then the parameter `userAgentHeader.addFqdnToHeader` needs to be enabled.	**Data Type**: String **Range**: NA **Example Value**: "nssf.oracle.com"

Table 5-14 (Cont.) Request or Response Body Parameters

Parameter	Description	Details
`overwriteHeader`	This is a mandatory parameter. This parameter is used to govern if we want to include the User-Agent Header generated at NSSF Egress Gateway or forward the User-Agent received from service request. Set its value as `true`. **Note:** When User-Agent Header is enabled but the header information is missing, then it is picked from the OAuthClient module. If the User-Agent Header is present in the request towards AMF or NRF, then the value present in the header is overwritten or forwarded based on the `overwriteHeader` flag. If the flag is set to `true`, then the header is overwritten.	**Data Type**: Boolean **Range**: true or false **Example Value**: false

Example

Example of Request or Response Body to Enable User-Agent Header:

```
{
   "enabled": true,
   "nfType": "NSSF",
   "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
"nfFqdn": "nssf.oracle.com",
"addFqdnToHeader": true,
 "overwriteHeader": true
}
```

> **Note**
>
> - In the mentioned configuration, when sending notifications to AMF, the User-Agent Header will be appended by the NSSF with the value "`NSSF-9faf1bbc-6e4a-4454-a507-aef01a101a01 nssf.oracle.com`."
> - The `nfInstanceId` and `nfFqdn` values in the above example are samples. Ensure that you update the values of the `nfInstanceId` and `nfFqdn` parameters accordingly.

Sample cURL to enable User-Agent Header:

```
curl --http2-prior-knowledge -X PUT \
  http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/useragentheader \
  -H "Content-Type: application/json" \
  -d '{
    "enabled": true,
    "nfType": "NSSF",
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
    "nfFqdn": "nssf.oracle.com",
```

```
    "addFqdnToHeader": true,
    "overwriteHeader": true
  }' \
  -v
```

Example of Request or Response Body to Disable User-Agent Header:

```
{
   "enabled": false,
   "nfType": "NSSF",
   "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
"nfFqdn": "nssf.oracle.com",
"addFqdnToHeader": true,
 "overwriteHeader": true
}
```

Sample cURL to disable User-Agent Header:

```
curl --http2-prior-knowledge -X PUT \
  http://{{host}}:{{port}}/nssf/nf-common-component/v1/egw/useragentheader \
  -H "Content-Type: application/json" \
  -d '{
    "enabled": false,
    "nfType": "NSSF",
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
    "nfFqdn": "nssf.oracle.com",
    "addFqdnToHeader": true,
    "overwriteHeader": true
  }' \
  -v
```

6
HTTP Response Codes

The following are HTTP Response Codes:

Table 6-1 HTTP Response Codes

Service	Service Operation	HTTP Request Method	HTTP Response Code	Condition
Nnssf_NsSelection	**Initial Registration**	**Get**	400 (Bad Request)	• All semantic, syntax errors leads to this response code. • This response code indicates that the request is not valid according to protocol such as invalid json and patch items.
			401 (Unauthorized)	Missing Authentication
			405 (Method Not Allowed)	Method not implemented for URI
			500 (Internal Error)	• Database operation error • Memory not available
			403 (Forbidden)	• Authentication failure • If Allowed SNSSAI is not matching • For a particular TAI if there is no SNSSAI in the allowed list • In case of PDU establishment • Requested SNSSAI is not allowed
			200 (OK)	Success Case
Nnssf_NsAvailability	**(UPDATE)**	**PUT**	200 (OK)	Success Case
			400 (Bad Request)	• All semantic, syntax errors leads to this response code. • This response code indicates that the request is not valid according to protocol such as invalid json and patch items.
			401 (Unauthorized)	Missing authentication
			403 (Forbidden)	When all SNSSAIs for all TAIs are not allowed in PLMN
			405 (Method Not Allowed)	Method not implemented for URI
			500 (Internal Error)	Database operation error Memory not available
Nnssf_NsAvailability	**(UPDATE)**	**PATCH**	200 (OK)	Success Case
			400 (Bad Request)	• All semantic, syntax errors leads to this response code. • This response code indicates that the request is not valid according to protocol such as invalid json and patch items.

Table 6-1 (Cont.) HTTP Response Codes

Service	Service Operation	HTTP Request Method	HTTP Response Code	Condition
			401 (Unauthorized)	Missing Authentication
			403 (Forbidden)	When all SNSSAIs for all TAIs are rejected in PLMN
			405 (Method Not Allowed)	Method not implemented for URI
			404 (Not found)	AMF Availability data not found
			500 (Internal Error)	• Database operation error • Memory not available
Nnssf_NsAvailability	**(Subscribe)**	**POST**	201 (Created)	Success Case
			401 (Unauthorized)	Missing Authentication
			400 (Bad Request)	• All semantic, syntax errors leads to this response code. • This response code indicates that the request is not valid according to protocol such as invalid json and patch items. • Expiry duration smaller than Min Expiry duation.
			500 (Internal Error)	500 (Internal Error)
Nnssf_NsAvailability	**(Subscribe)**	**DELETE**	204 (No Content)	Success Case
			401 (Unauthorized)	Missing Authentication
			404 (Not found)	Subscription-ID not found
			500 (Internal Error)	Database operation error