# Oracle® Communications

# Cloud Native Core, Network Slice Selection Function REST Specification Guide

Release 25.2.200
G40405-01
February 2026

Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide, Release 25.2.200

G40405-01

# 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 25.2.2xx in Oracle Communications Cloud Native Core, Network Slice Selection Function REST Specification Guide.

**Release 25.2.200- G40405-01, February 2026**

- New and redesigned managed objects have been introduced to simplify REST API configured NSSF options, system settings, and profile management. Several outdated managed objects have also been updated or removed to streamline the configuration.
- Performed the following updates in the NSSF REST Specifications section:
  - Added the following APIs:
    * SystemOptions
    * SupportedSlicesMapping
    * PlmnConfig
    * BarredSlicesMapping
  - Updated the Following APIs:
    * NsiProfiles
    * GrSites
    * NSSF Backup and Restore Configurations
  - Removed nrf-client-discovery fromRuntime Log Level Update section as it is deprecated now.
- Added Appendix A - Common Error Responses for Managed Objects .
- Added Appendix B - Important Guidelines for Configuring the Managed Objects

# 1 Introduction

This document provides information about how to configure the services and managed objects (MO) 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 Protocol 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 objects used in Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF).

## 2.1 Primitive Tables

### Plmnid

**Table 2-1 Plmnid**

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

### Snssai

**Table 2-2 Snssai**

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

### PlmnInfo

**Table 2-3 PlmnInfo- Parameters**

| Parameter | Description | Details |
|---|---|---|
| configuredNssai | This is a mandatory parameter.<br>The configuredNSSAI parameter contains the list of all S-NSSAIs configured at the PLMN level, but this alone does not ensure authorization.<br>• To consider an S-NSSAI as part of the TAI-S-NSSAI mapping, it must be explicitly mapped to a TAI.<br>• If an S-NSSAI is supported but not configured at the PLMN level, it will not be considered for selection.<br>Autoconfiguration Scenarios:<br>1. **Autoconfiguration ON:** The AMF dynamically learns the TAI-S-NSSAI mapping.<br>2. **Autoconfiguration OFF:** The operator must manually configure the supported NSSAI per TAI to establish this mapping. | **Data Type:** array<SnssaiInfo><br>See SnssaiInfo |

**Table 2-3 (Cont.) PlmnInfo- Parameters**

| Parameter | Description | Details |
|---|---|---|
| barredSnssaiList | This is an optional parameter.<br>**Slices Configured as Restricted for PLMN**<br>S-NSSAIs restricted for the PLMN: When present, these S-NSSAIs will be restricted within the PLMN. Any S-NSSAI included in this list must not be present in the allowed S-NSSAI list for any TAI in the PLMN.To add an already existing barred slice at the TAI level to the barredSnssaiList, it must first be removed from BarredSlicesMapping. | **Data Type:** array<SnssaiInfo><br>See SnssaiInfo |
| nsiInformationList | This is a mandatory parameter.<br>Default fallback network slice instance information for the PLMN. This applies to all TAI-SNSSAI combinations within the PLMN, unless specifically overridden in allowedSnssai. If nsiInformation is not configured, or in the case of autoconfiguration, this serves as the default fallback. | **Data Type:**<br>array(NsiInformationInfo) |

### SnssaiInfo

**Table 2-4 SnssaiInfo - Parameters**

| Parameter | Description | Details |
|---|---|---|
| sst | This is a mandatory parameter. | **Data Type:** Integer |
| sd | This is an optional parameter. | **Data Type:** string |
| accessType | This is a mandatory parameter. | **Data Type:** enum("3GPP", "NON_3GPP", "BOTH") |
| nsiInformationInfo | This is an optional parameter.<br>For more information see, NsiInformationInfo. | **Data Type:**<br>array(NsiInformationInfo) |

### NsiInformationInfo

**Table 2-5 NsiInformationInfo - Parameters**

| Parameter | Description | Details |
|---|---|---|
| nsiProfileId | This is a mandatory parameter.<br>SliceId of NSI profile | **Data Type:** string |
| salience | This is an optional parameter.<br>An input value must be more than 1. | **Data Type:** Integer |

### PlmnTacList

**Table 2-6 PlmnTacList - Parameters**

| Parameter | Description | Details |
|---|---|---|
| `plmnId` | This is a mandatory parameter.<br>The PLMN ID uniquely identifies the PLMN and should be formatted as MCC-MNC. It must be cross-checked with the PLMN-level information corresponding to the `plmnId`. If the PLMN information is not present, respond with a 400 Bad Request. | **Data Type:** string |
| `tacList` | This is a mandatory parameter.<br>List of TACs in PLMN "plmnId" | **Data Type:** List<Tac> |

## 2.2 Enumerations

### Access Type

**Table 2-7 Access Type**

| Value | Description |
|---|---|
| "3GPP" | Specifies 5G network |
| "NON_3GPP" | Specifies non 5G network |
| "BOTH" | Specifies both |

# 3
# NSSF REST Specifications

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

NSSF can be configured using Helm configurations, REST APIs, and Cloud Native Core (CNC) Console. The NSSF deployment configurations are performed during 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*.

> **Note**
>
> For details on common response codes and key guidelines for configuring managed objects, see:
>
> - Appendix A - Common Error Responses for Managed Objects .
> - Appendix B - Important Guidelines for Configuring the Managed Objects

## 3.1 SystemOptions

Managed object to configure SystemOptions. It consists of feature flags used to enable or disable various features within NSSF.

- This Managed Object is independent of other Managed Objects and is created during a fresh installation of NSSF, with values based on helm parameters.
- The operator can later update these options, but deletion is not supported; only **GET** and **PUT** operations are allowed if SystemOptions is not configured.

> **Note**
>
> - During a fresh installation, no configuration is present for system options. The operator must configure the system options first. Until the system options are configured, it is not possible to configure other settings.
> - All NSSF microservices check the database for SystemOptions upon startup.
> - During an upgrade, the current system options are stored in a backup table, and the new system options for the updated version are computed and stored.

**Table 3-1 SystemOptions - Supported REST APIs**

| URI | Description | Data Type | HTTP Method |
|---|---|---|---|
| /nnssf-configuration/v1/systemoptions | Returns array of all configured SystemOptions. There will be only one active SystemOptions on an NSSF. | SystemOptions | GET |
| /nnssf-configuration/v1/systemoptions | Update the existing SystemOptions object | SystemOptions | PUT |

### SystemOptions - Dependencies

This Managed Object is not dependent on other Managed Objects.

### Request or Response Body Parameters

**Table 3-2 Request or Response Body Parameters**

| Parameter | Description | Details |
|---|---|---|
| `autoAuthorizeNssaiAvailabilityDataEnable` | This is a mandatory parameter.<br>Main object containing other objects required for enhanced computation of allowedNSSAI in NSSF. | **Data Type:** Boolean |
| `enhancedPatchBehaviourEnable` | This is a mandatory parameter.<br>Enables or disables allowing zero Snssai in TAI. When true, NSSF shall allow NssaiAvailabilityPatch to remove all Supported SNSSAIs in TAI, as well as in PUT for NssaiAvailability. | **Data Type:** Boolean |
| `plmnLevelSystemOptionsList` | This is a mandatory parameter.<br>Array of per-PLMN system options. Every PLMN must be covered, otherwise API will return 400 Bad Request. | **Data Type:** Array |

**Table 3-3 Request or Response Body Parameters- plmnLevelSystemOptions**

| Parameter | Description | Details |
|---|---|---|
| `plmnId` | This is a mandatory parameter.<br>PLMN ID in format mcc-mnc. Must uniquely identify the PLMN. Cross-check with PLMN level info to verify all PLMNs are covered.<br>For more details, see Plmnid. | **Data Type:** String |
| `enhancedAllowedNssaiEnable` | This is a mandatory parameter.<br>Enables or disables enhancedAllowedNssai feature for the specified PLMN. | **Data Type:** Boolean |

**Example**

```
{
  "autoAuthorizeNssaiAvailabilityDataEnable": false,
  "enhancedPatchBehaviourEnable": false,
  "plmnLevelSystemOptionsList": [
    {
      "plmnId": "311-480",
      "enhancedAllowedNssaiEnable": false
    }
  ]
}
```

# 3.2 NsiProfiles

The NsiProfiles is a mandatory Managed Object. It consists of network slice identification information, including the NSI ID and NRF-URI. This Managed Object contains the essential details defining a network slice instance, used for NSSF operation and selection.

**NSI Profiles - Supported REST APIs**

**Table 3-4 NSI Profiles - Supported REST APIs**

| URI | Description | Data Type | HTTP Method |
|---|---|---|---|
| /nnssf-configuration/v1/nsiprofiles | Returns an array of all configured NSI Profiles. | array(NsiProfile) | GET |
| /nnssf-configuration/v1/nsiprofiles | Create a network slice instance profile. | NsiProfile | POST |
| /nnssf-configuration/v1/nsiprofiles/{nsiId} | Read a network slice instance profile. Returns NsiProfile based on the given `nsiId`. | NsiProfile | GET |

**Table 3-4 (Cont.) NSI Profiles - Supported REST APIs**

| URI | Description | Data Type | HTTP Method |
|---|---|---|---|
| /nnssf-configuration/v1/nsiprofiles/{nsiId} | Delete a network slice instance profile based on `nsiId`.<br>**Checks:**<br>- NSI Profile must not be linked to any ConfiguredSNSSAI in any PLMN.<br>- NSI Profile must not be linked to any not allowed SupportedSlicesMapping. | NsiProfile | DELETE |
| /nnssf-configuration/v1/nsiprofiles/{nsiId} | Update a network slice instance profile based on `nsiId`.<br>**Note:** nsiId is immutable; all other parameters are mutable. | NsiProfile | PUT |

### NSI Profiles - Dependencies

This Managed Object is not dependent on other Managed Objects. No prerequisites are required for configuring NSI Profiles.

### Request or Response Body Parameters

The **NSI Profiles** Managed Object allows the configuration and management of **Network Slice Instance (NSI) profiles**. The operator can create, modify, or retrieve these profiles by providing the appropriate parameters in the request or by receiving them in the response.

The key parameters used in the request or response bodies enable the configuration of the **Network Slice** by defining specific attributes, such as the **NSI ID** and **NRF URL** (Network Resource Function URL) corresponding to the slice.

**Table 3-5 NSI Profiles - Parameters**

| Parameter | Description | Details |
|---|---|---|
| `nrfUri` | This is a mandatory parameter.<br>Contains the API URI of the NRF NFDiscovery Service to be used for selecting the NFs or services within the selected Network Slice instance. | **Data Type:** URI |
| `nsiId` | This is a mandatory parameter.<br>Contains the identifier of the selected Network Slice Instance<br>**Note**: `nsiId` is stored in UPPER CASE in database. | **Data Type:** NsiId |
| `nrfNfMgtUri` | This is an optional parameter.<br>When present, it contains the API URI of the NRF NFManagement Service. | **Data Type:** URI |
| `nrfAccessTokenUri` | This is a mandatory parameter.<br>When present, it contains the API URI of the NRF Access Token Service. | **Data Type:** URI |

**Examples**

```
{
 "nsiId": "NSI-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"
}
```

> **Note**
>
> `nsiId` is stored in UPPER CASE in database.

# 3.3 PlmnConfig

The PlmnConfig is a mandatory managed object that contains configurations applicable to a specific PLMN, uniquely identified by its PLMN ID. Multiple PLMN is supported. Hence, it is required to configure all supported PLMNs.

**Key Parameters:**

1. `plmnid` – Mandatory
   - **Description:** Identifies the PLMN for which the configuration is being applied.
2. `configuredNSSAI` (Configured Network Slice Selection Assistance Information) – Mandatory
   - **Description:** Lists all S-NSSAIs (Single Network Slice Selection Assistance Information) configured at the PLMN level. This represents the complete set of S-NSSAIs that can be supported in TAIs (Tracking Area Identifiers) within the PLMN.
   - **Usage:**
     - If an NsSelection GET request contains a `requestedNSSAI` not present in `configuredNSSAI`, the NSSF (Network Slice Selection Function) will include the configured NSSAI in the response.
     - S-NSSAI support for a TAI within the PLMN is considered unauthorized unless explicitly listed in `configuredNSSAI`.
3. **`barredSnssaiList` (Restricted Network Slice Selection Assistance Information) – Optional**
   - **Description:** Lists S-NSSAIs that are restricted or barred within the PLMN. Any S-NSSAI in this list must not be included in the `supportedslicesmapping` for any TAI within the PLMN, ensuring that restricted S-NSSAIs are unavailable for use.
   - **Usage:** Ensures that certain S-NSSAIs are completely unavailable for selection within the PLMN.
4. **`nsiInformationList` (Network Slice Instance Information) – Mandatory**
   - **Description:** Defines the default fallback network slice instance information for the PLMN. This information applies to all TAI-S-NSSAI combinations unless overridden by Allowed NSSAI.

- **Usage:** If `nsiInformation` is not explicitly configured or if auto-configuration is enabled, this list serves as the default fallback for slice selection.

### PlmnConfig - Supported REST APIs

**Table 3-6 PlmnConfig - Supported REST APIs**

| URI | Description | Details |
|---|---|---|
| /nnssf-configuration/v1/plmnconfig | Create a PLMN Config | **Data Type:** plmnconfig<br>**HTTP Method:** POST |
| /nnssf-configuration/v1/plmnconfig | Read all PLMN Configs | **Data Type:** plmnconfig<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/plmnconfig/{plmnId} | Update a PLMN Config | **Data Type:** plmnconfig<br>**HTTP Method:** PUT |
| /nnssf-configuration/v1/plmnconfig/{plmnId} | Read a PLMN Config | **Data Type:** plmnconfig<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/plmnconfig/{plmnId} | Delete a PLMN Config | **Data Type:** plmnconfig<br>**HTTP Method:** DELETE |

### PlmnConfig- Dependencies

Required `NsiProfiles` and `SystemOptions` must be configured beforehand in order to configure `PlmnConfig`. Only PLMNs that are configured in `SystemOptions` will be accepted as part of `PlmnConfig`.

### Request or Response Body Parameters

**Table 3-7 PlmnConfig- Parameters**

| Parameter | Description | Details |
|---|---|---|
| `plmnId` | This is a mandatory parameter.<br>Identifies the PLMN for which the configuration is being made. Format: mcc-mnc.<br>For more details, see Plmnid. | **Data Type:** String |
| `plmnInfo` | This is a mandatory parameter.<br>Configuration of PLMN: Contains the Configured NSSAI for the PLMN, the Barred SNSSAI for the PLMN, and the default fallback Network Slice Instance information for the PLMN.<br>For more information, see PlmnInfo. | **Data Type:** PlmnInfo |

### Examples

```
{
    "plmnId": "311-480",
    "plmnInfo": {
        "configuredNssai": [
            {
                "sst": "1",
                "sd": "EABB01",
                "accessType": "3GPP_ACCESS",
                "nsiInformationList": [
```

```
                {
                    "nsiProfileId": "NSI-1",
                    "salience": "4"
                },
                {
                    "nsiProfileId": "NSI-2",
                    "salience": "2"
                }
            ]
        },
        {
            "sst": "2",
            "sd": "EABB02",
            "accessType": "3GPP_ACCESS",
            "nsiInformationList": [
                {
                    "nsiProfileId": "NSI-2",
                    "salience": "3"
                }
            ]
        },
        {
            "sst": "3",
            "sd": "EABB03",
            "accessType": "BOTH"
        },
        {
            "sst": "4",
            "sd": "EABB04",
            "accessType": "NON_3GPP_ACCESS"
        },
        {
            "sst": "5",
            "sd": "EABB05",
            "accessType": "BOTH"
        },
        {
            "sst": "6",
            "sd": "EABB06",
            "accessType": "NON_3GPP_ACCESS"
        },
        {
            "sst": "7",
            "sd": "EABB07",
            "accessType": "BOTH"
        },
        {
            "sst": "8",
            "sd": "EABB08",
            "accessType": "NON_3GPP_ACCESS"
        },
        {
            "sst": "9",
            "sd": "EABB09",
            "accessType": "BOTH"
        },
```

```
            {
                "sst": "10",
                "sd": "EABB10",
                "accessType": "NON_3GPP_ACCESS"
            }
        ],
        "barredSnssaiList": [
            {
                "sst": "5",
                "sd": "EABB05"
            }
        ],

        "nsiInformationList": [
            {
                "nsiProfileId": "NSI-1",
                "salience": "3"
            }
        ]
    }
}
```

# 3.4 SupportedSlicesMapping

The **SupportedSlicesMapping** is an optional managed object that enables configuration and management of supported slices based on TAI (Tracking Area Identity).

This managed object is applicable only if auto-authorization of NssaiAvailability data is not enabled. Operator shall be allowed to configure this managed object even when the feature is enabled, but such configurations will be stored in the database without effect.

### SupportedSlicesMapping - Supported REST APIs

**Table 3-8 SupportedSlicesMapping - Supported REST APIs**

| URI | Description | Data Type | HTTP Method |
|---|---|---|---|
| /nnssf-configuration/v1/supportedslicesmapping | Create a SupportedSlicesMapping. | supportedslicesmapping | POST |
| /nnssf-configuration/v1/supportedslicesmapping | Read all SupportedSlices mappings | supportedslicesmapping | GET |
| /nnssf-configuration/v1/supportedslicesmapping/{name} | Update a SupportedSlices mapping | supportedslicesmapping | PUT |
| /nnssf-configuration/v1/supportedslicesmapping/{name} | Read a Supported Slices mapping | supportedslicesmapping | GET |
| /nnssf-configuration/v1/supportedslicesmapping/{name} | Delete a Supported Slices mapping | supportedslicesmapping | DELETE |

### SupportedSlicesMapping - Dependencies

We need `NsiProfile`, `SystemOptions`, and `PlmnConfig` to configure Supported Slices Mapping.

### Request or Response Body Parameters

**Table 3-9 SupportedSlicesMapping - Parameters**

| Parameter | Description | Details |
|---|---|---|
| `name` | This is a mandatory parameter.<br>Unique Identifier of the mapping.<br>**Note**: `name` is stored in UPPER CASE in the database. | **Data Type:** string |
| `supportedSnssaiList` | This is a mandatory parameter.<br>SnssaiInfo Allowed for List of TAIs<br>1. **Validate NSI Profile Existence**: Ensure that each NSI profile specified is present in the NSI profiles table.<br>2. **Validate SNSSAI Uniqueness**: Check that there are no duplicate SNSSAIs within the same entry.<br>3. **Validate SNSSAI Configuration**: Verify that all specified SNSSAIs are part of the `ConfiguredNssai` for the PLMN specified in `plmnTacList`.<br>4. **Validate SNSSAI Barred Status**: Ensure that none of the specified SNSSAIs are included in the `barredSnssaiList` for the PLMN specified in `plmnTacList`.<br>5. **Validate SNSSAI Configuration Conflict**: Check that no SNSSAI is configured in `BarredSlicesMappingConfig` for the same TAI.<br>6. **Validate AccessType**: Validate `AccessType` of each SNSSAI should be the same as that of `AccessType` in corresponding SNSSAI in the `ConfiguredNssai` for the PLMN specified. | **Data Type:** array<SnssaiInfo><br>See SnssaiInfo |

**Table 3-9 (Cont.) SupportedSlicesMapping - Parameters**

| Parameter | Description | Details |
|---|---|---|
| `plmnTacList` | This is a mandatory parameter.<br>List of TACs in a PLMN for Which Supported Slices are Being Configured per PLMN<br>Validations to be Performed<br>1. **Validate PLMN Membership**: Verify that the PLMN obtained from `plmnTacList` is included in the list of supported PLMNs.<br>2. **Validate TAC Uniqueness**: Ensure that the `tacList` contains unique TACs, with no duplication of TACs either within the same entry or across different entries.<br>Examples and Invalid Cases<br>• **Supported PLMNs Example**:<br>– PLMN: (311-480)<br>• **plmnTacList Example 1**:<br>– plmn: 311-480<br>* tac1: (1001)<br>* tac2: (1002)<br>* tac3: (1003)<br>• **Invalid Examples**:<br>– **Duplication of TACs in plmnTacList**:<br>* Entry 1: tac1, tac2 and Entry 2: tac2, tac3 (Invalid, as tac2 is common between the entries)<br>* Entry 1: tac1, tac2, tac1 (Invalid, as tac1 is repeated within the same entry)<br>– **plmnTacList Belonging to an Unsupported PLMN**:<br>* The `plmnTacList` contains a PLMN that is not supported.<br>For more information, see PlmnTacList. | **Data Type:** PlmnTacList |

### Examples

```
{
"name": "SUPPORTED-SNSSAIS-CONFIG-1",
"supportedSnssaiList": [
  {
    "sst": 1,
    "sd": "EABB01",
    "accessType": "3GPP_ACCESS",
    "nsiInformationList": [
      {
        "nsiProfileId": "NSI-1",
        "salience": 2
      }
    ]
  },
  {
    "sst": 1,
```

```
        "sd": "EABB02",
        "accessType": "3GPP_ACCESS",
        "nsiInformationList": [
          {
            "nsiProfileId": "NSI-1",
            "salience": 2
          }
        ]
      }
    ],
    "plmnTacList": {
      "plmnId": "311-480",
      "tacList": [
        {
          "tac": "202400"
        },
        {
          "tac": "202401"
        },
        {
          "tac": "202402"
        }
      ]
    }
}
```


# 3.5 BarredSlicesMapping

This is an optional managed object. The Barred Slices Mapping Config allows for the configuration and management of slice restrictions based on TAI (Tracking Area Identity). The operator is allowed to configure this managed object even when the feature is enabled, but in that case, the configuration will only be stored in the database and will not take effect.

**BarredSlicesMapping - Supported REST APIs**

**Table 3-10 BarredSlicesMapping - Supported REST APIs**

| URI | Description | Details |
|---|---|---|
| /nnssf-configuration/v1/barredslicesmapping | Create a Barred Slices mapping | **Data Type:** barredslicesmapping<br>**HTTP Method:** POST |
| /nnssf-configuration/v1/barredslicesmapping | Read all Barred Slices mappings | **Data Type:** barredslicesmapping<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/barredslicesmapping/{name} | Update a Barred Slices mapping | **Data Type:** barredslicesmapping<br>**HTTP Method:** PUT |
| /nnssf-configuration/v1/barredslicesmapping/{name} | Read a Barred Slices mapping | **Data Type:** barredslicesmapping<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/barredslicesmapping/{name} | Delete a Barred Slices mapping | **Data Type:** barredslicesmapping<br>**HTTP Method:** DELETE |

### BarredSlicesMapping- Dependencies

Required `NsiProfiles`, `SystemOptions`, and `PlmnConfig` must be configured beforehand in order to configure `BarredSlicesMapping`.

### Request or Response Body Parameters

**Table 3-11 BarredSlicesMapping- Parameters**

| Parameter | Description | Details |
|---|---|---|
| `name` | This is a mandatory parameter.<br>Unique Identifier of mapping<br>**Note**: `name` is stored in UPPER CASE in the database. | **Data Type:** string |
| `barredSnssaiList` | This is a mandatory parameter.<br>Slices Configured as Barred for List of TACs<br>Validations to be performed:<br>1. **Validate SNSSAI Uniqueness:** Ensure there are no duplicate SNSSAIs within the same entry.<br>2. **Validate SNSSAI Configuration:** Verify that all specified SNSSAIs are included in the `ConfiguredNssai` for the PLMN specified in `plmnTacList`.<br>3. **Validate SNSSAI Barred Status:** Ensure that none of the specified SNSSAIs are present in the `barredSnssaiList` for the PLMN specified in `plmnTacList`.<br>4. **Validate SNSSAI Configuration Conflict:** Check that no SNSSAI is configured in the `SupportedSlicesMappingConfig` for the same TAI. | **Data Type:** array<Snssai><br>See Snssai |

**Table 3-11 (Cont.) BarredSlicesMapping- Parameters**

| Parameter | Description | Details |
|---|---|---|
| `plmnTacList` | This is a mandatory parameter.<br>List of TACs for Which Barred Slices Are Being Configured per PLMN<br>Validations to be Performed:<br>1. **Validate PLMN Membership:** Verify that the PLMN specified in `plmnTacList` is part of the supported PLMNs.<br>2. **Validate TAC Uniqueness:** Ensure that the `tacList` contains unique TACs, with no duplication either within the same entry or across different entries.<br>Examples and Invalid Cases:<br>• **Supported PLMNs Example:**<br>– PLMN: (311-480)<br>• **plmnTacList Example 1:**<br>– plmn: 311-480<br>* tac1: (1001)<br>* tac2: (1002)<br>* tac3: (1003)<br>• **Invalid Examples:**<br>– **Duplication of TACs in plmnTacList:**<br>* Entry 1: tac1, tac2 and Entry 2: tac2, tac3 (Invalid, as tac2 is common between the entries)<br>* Entry 1: tac1, tac2, tac1 (Invalid, as tac1 is repeated within the same entry)<br>– **plmnTacList Belonging to an Unsupported PLMN:**<br>* The `plmnTacList` contains a PLMN that is not supported.<br>For more information, see PlmnTacList. | **Data Type:** PlmnTacList |

### Examples

```
{
  "name": "BARRED-SNSSAIS-CONFIG-1",
  "barredSnssaiList": [
    {
      "sst": 3,
      "sd": "EABB03"
    },
     {
      "sst": 4,
      "sd": "EABB04"
    }
  ],
  "plmnTacList": {
    "plmnId": "311-480",
    "tacList": [
      {
        "tac": "202509"
```

```
      },
      {
        "tac": "202510"
      }
    ]
  }
}
```

# 3.6 GrSites

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

**GrSites - Supported REST APIs**

**Table 3-12 GrSites - Supported REST APIs**

| URI | Description | Details |
|---|---|---|
| /nnssf-configuration/v1/grsites | It returns array of all configured GrSites. | **Data Type:** Array(GrSite)<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/grsites/{nfInstanceId} | It returns GrSites for corresponding nfInstanceId. | **Data Type:** Array(GrSite)<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/grsites | It creates a GrSite. | **Data Type:** GrSite<br>**HTTP Method:** POST |
| /nnssf-configuration/v1/grsites/{nfInstanceId} | It updates GrSite rank for corresponding nfInstanceId. Changes in other fields will be ignored. | **Data Type:** GrSite<br>**HTTP Method:** PUT |
| /nnssf-configuration/v1/grsites/{nfInstanceId} | It deletes GrSite for corresponding nfInstanceId. | **Data Type:** GrSite<br>**HTTP Method:** DELETE |

**GrSites - 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 georedundancy is enabled in the helm options. For information about enabling georedundancy, see **Georedundancy** section in *Oracle Communications Cloud Native Core, Network Slice Selection Function User Guide*.

**Request or Response Body Parameters**

The GrSites 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.

**Table 3-13 GrSites - Parameters**

| Parameter | Description | Details |
|---|---|---|
| nfInstanceId | This is a mandatory parameter.<br>Contains the instance ID of the site NSSF.<br>Validation - Controlled at application layer<br>uuid<br>Unique | **Data Type:** String |
| siteId | This is a mandatory parameter.<br>Contains the instance ID of the site NSSF.<br>Validation - Controlled at application layer<br>Unique | **Data Type:** String |
| Rank | This is a mandatory parameter.<br>Contains the priority given by the operator to related georedundant sites.<br>Validation - Controlled at application layer<br>Unique | **Data Type:** Integer |

### Examples

**Sample PlmnConfig**

```
{
    "siteId": "01",
    "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
    "rank": 1
}
```

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

**Example Curl Command:**

```
1. curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"1", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1", "siteId": "01"}'
2. curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"2", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b2", "siteId": "02"}'
3. curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"3", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b3", "siteId": "03"}'
```

**Example Response:**

```
1. { "rank": "1", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1",
"siteId": "01", "grSiteStatus":"ACTIVE" }
2. { "rank": "2", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b2",
"siteId": "02", "grSiteStatus":"ACTIVE" }
```

```
3. { "rank": "3", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b3",
"siteId": "03", "grSiteStatus":"ACTIVE" }
```

**Example Curl Command:**

```
curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"1", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1", "siteId": "01"}'

curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"2", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b2", "siteId": "02"}'

curl --http2-prior-knowledge -X POST http://{{host}}:{{port}}/nnssf-
configuration/v1/grsites -H 'Content-Type: application/json' -d '{ "rank":
"3", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b3", "siteId": "03"}'
```

**Example of the Response Body:**

```
{ "rank": "1", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b1",
"siteId": "01", "grSiteStatus":"ACTIVE" }

{ "rank": "2", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b2",
"siteId": "02", "grSiteStatus":"ACTIVE" }

{ "rank": "3", "nfInstanceId": "fe7d992b-0541-4c7d-ab84-c6d70b1b01b3",
"siteId": "03", "grSiteStatus":"ACTIVE" }
```

# 3.7 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: PUT/GET/DELETE
- URL: *http://{apiRoot}/nnssf-configuration/v1/allconfig*
- 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 **PUT**, **GET**, and **DELETE**. Perform the configuration of ConfigurationBackup Managed Object as follows:

**Table 3-14 NSSF Backup and Restore Configurations - Supported REST APIs**

| URI | Description | Details |
|---|---|---|
| /nnssf-configuration/v1/allconfig | Overrides all configuration managed objects.<br>Deletes existing configuration and overrides. | **Data Type:** ConfigBackup<br>**HTTP Method:** PUT |

**Table 3-14 (Cont.) NSSF Backup and Restore Configurations - Supported REST APIs**

| URI | Description | Details |
|---|---|---|
| /nnssf-configuration/v1/allconfig | Responds with get ALL for all configured managed objects. | **Data Type:** ConfigBackup<br>**HTTP Method:** GET |
| /nnssf-configuration/v1/deleteConfiguration | Deletes existing configuration. | **HTTP Method:** DELETE |

### Request or Response Body Parameters

ConfigurationBackup consists of list of following parameters:

**Table 3-15 NSSF Backup and Restore Configurations - Parameters**

| Parameter | Description | Details |
|---|---|---|
| `systemOptionsConfigInfo` | This is a mandatory parameter.<br>This contains all the `systemOptions` configurations.<br>`systemOptions` will not be deleted by the `deleteConfiguration` API; it can only be updated using the PUT method. | **Data Type**: SystemOptionsConfigInfoDto |
| `nsiProfileConfigList` | This is an optional parameter<br>This contains the list of all `nsiProfileConfigs`. | **Data Type:** Array<NsiProfileConfigDto> |
| `plmnConfigList` | This is an optional parameter<br>This contains the list of all plmnConfig.<br>This is dependent on `systemOptionsConfigInfo` and `nsiProfileConfigList`. Proper `systemOptionsConfigInfo` should already exist, or it must be included in the API request body. Additionally, `nsiProfileConfigList` needs to be present in the API request body for `nsiInformationList` in `plmnConfig`. | **Data Type:** Array<PlmnConfigDto> |
| `barredSlicesMappingConfigList` | This is an optional parameter<br>This contains the list of all `barredSlicesMapping`.It is dependent on `plmnConfigList` and `nsiProfileConfigList`. | **Data Type:** Array<BarredSlicesMappingConfigDto> |
| `supportedSlicesMappingConfigList` | This is an optional parameter<br>This contains the list of all `supportedSlicesMappingConfig`.It is dependent on `plmnConfigList` and `nsiProfileConfigList`. | **Data Type:** Array<SupportedSlicesMappingConfigDto> |
| `geoRedundantSitesInfoList` | This is a mandatory parameter<br>This contains the list of all `geoRedundantSitesInfo`.This is a mandatory parameter and must not be empty. | **Data Type:** Array<GeoRedundantSitesInfoDto> |

### Examples

```
{
    "systemOptionsConfigInfo": {
        "autoAuthorizeNssaiAvailabilityDataEnable": true,
        "enhancedPatchBehaviourEnable": true,
        "plmnLevelSystemOptionsList": [
            {
                "plmnId": "311-480",
```

```
                "enhancedAllowedNssaiEnable": true
            },
            {
                "plmnId": "100-101",
                "enhancedAllowedNssaiEnable": true
            }
        ]
    },
    "nsiProfileConfigList": [
        {
            "nsiId": "NSI-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": "NSI-2",
            "nrfUri": "https://nrf.slice1.oracle.com/nnrf-disc/v2",
            "nrfNfMgtUri": "https://nrf.slice1.oracle.com/nnrf-nfm/v2",
            "nrfAccessTokenUri": "https://nrf.slice2.oracle.com/oauth2/token"
        }
    ],
    "plmnConfigList": [
        {
            "plmnId": "311-480",
            "plmnInfo": {
                "configuredNssai": [
                    {
                        "sst": "1",
                        "sd": "EABB01",
                        "accessType": "3GPP_ACCESS",
                        "nsiInformationList": [
                            {
                                "nsiProfileId": "NSI-1",
                                "salience": "4"
                            },
                            {
                                "nsiProfileId": "NSI-2",
                                "salience": "2"
                            }
                        ]
                    },
                    {
                        "sst": "2",
                        "sd": "EABB02",
                        "accessType": "3GPP_ACCESS",
                        "nsiInformationList": [
                            {
                                "nsiProfileId": "NSI-2",
                                "salience": "3"
                            }
                        ]
                    },
                    {
                        "sst": "3",
                        "sd": "EABB03",
```

```
                        "accessType": "BOTH"
                    },
                    {
                        "sst": "4",
                        "sd": "EABB04",
                        "accessType": "NON_3GPP_ACCESS"
                    },
                    {
                        "sst": "5",
                        "sd": "EABB05",
                        "accessType": "BOTH"
                    },
                    {
                        "sst": "6",
                        "sd": "EABB06",
                        "accessType": "NON_3GPP_ACCESS"
                    },
                    {
                        "sst": "7",
                        "sd": "EABB07",
                        "accessType": "BOTH"
                    },
                    {
                        "sst": "8",
                        "sd": "EABB08",
                        "accessType": "NON_3GPP_ACCESS"
                    },
                    {
                        "sst": "9",
                        "sd": "EABB09",
                        "accessType": "BOTH"
                    },
                    {
                        "sst": "10",
                        "sd": "EABB10",
                        "accessType": "NON_3GPP_ACCESS"
                    }
                ],
                "barredSnssaiList": [
                    {
                        "sst": "5",
                        "sd": "EABB05"
                    }
                ],
                "nsiInformationList": [
                    {
                        "nsiProfileId": "NSI-1",
                        "salience": "3"
                    }
                ]
            }
        }
    ],
    "barredSlicesMappingConfigList": [
        {
            "name": "Barred-NSSAIS-CONFIG-1",
```

```
            "barredSnssaiList": [
                {
                    "sst": 3,
                    "sd": "EABB03"
                },
                {
                    "sst": 4,
                    "sd": "EABB04"
                }
            ],
            "plmnTacList": {
                "plmnId": "311-480",
                "tacList": [
                    {
                        "tac": "202509"
                    },
                    {
                        "tac": "202510"
                    }
                ]
            }
        }
    ],
    "supportedSlicesMappingConfigList": [
        {
            "name": "SUPPORTED-SLICES-MAPPING-CONFIG-1",
            "supportedSnssaiList": [
                {
                    "sst": 1,
                    "sd": "EABB01",
                    "accessType": "3GPP_ACCESS",
                    "nsiInformationList": [
                        {
                            "nsiProfileId": "NSI-1",
                            "salience": 2
                        }
                    ]
                }
            ],
            "plmnTacList": {
                "plmnId": "311-480",
                "tacList": [
                    {
                        "tac": "200000"
                    }
                ]
            }
        }
    ],
    "geoRedundantSitesInfoList": [
        {
            "siteId": "01",
            "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a01",
            "rank": 1
        },
        {
```

```
                "siteId": "02",
                "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a02",
                "rank": 2
            },
            {
                "siteId": "03",
                "nfInstanceId": "9faf1bbc-6e4a-4454-a507-aef01a101a03",
                "rank": 3
            }
        ]
}
```

# 3.8 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 3-16 Runtime Log Level Update - Parameters**

| Parameter | Description | Details |
|---|---|---|
| `commonCfgClient.enabled` | This is a mandatory parameter.<br>It is used to enable or disable client. | **Data Type:** Boolean |
| `commonCfgClient.pollingInterval` | This is a mandatory parameter.<br>It is used to set polling interval in Milliseconds. | **Data Type:** Integer |

### 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 3-17 Runtime Log Level Update - Supported REST APIs**

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

### Examples API Calls

**1. GET All Services**

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

**2. GET Log Level for a Microservice**

**cURL**: http://<nsconfig-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://<nsconfig-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 3-18 App-Info**

| Parameter Name | Description | Details |
|---|---|---|
| `appLogLevel` | This is a mandatory parameter<br>Application log level | **Data Type:** String<br>**Range:** TRACE, DEBUG, INFO, WARN, ERROR, 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 3-19 Perf-Info**

| Parameter Name | Description | Details |
|---|---|---|
| appLogLevel | This is a mandatory parameter<br>Application log level | **Data Type:** String<br>**Range:** TRACE, DEBUG, INFO, WARN, ERROR, 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 3-20 Ingress Gateway**

| Parameter Name | Description | Details |
|---|---|---|
| `appLogLevel` | This is a mandatory parameter<br>Application-wide log level | **Data Type:** String<br>**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF |
| `logDiscarding.enabled` | This is a mandatory parameter<br>Enable/disable log discarding | **Data Type:** Boolean<br>**Range:** true, false |
| `featureToThresholdMapping` | This is a mandatory parameter<br>Feature thresholds list | **Data Type:** List of Objects<br>**Range:** Each item contains feature and thresholdFactor |
| `feature` | This is a mandatory parameter<br>Feature name | **Data Type:** String<br>**Range:**<br>• RATE_LIMITING<br>• OVERLOAD_CONTROL<br>• ROUTE_LEVEL_RATE_LIMITING<br>• EGRESS_RATE_LIMITING |
| `thresholdFactor` | This is a mandatory parameter<br>Threshold factor for feature | **Data Type:** Integer<br>**Range:** 0–100 |
| `packageLogLevel` | This is a mandatory parameter<br>Specific package log settings | **Data Type:** List of Objects<br>**Range**: Each item contains packageName and logLevelForPackage |
| `packageName` | This is a mandatory parameter<br>Name of the package | **Data Type**: String<br>**Range**: Any valid Java package name |
| `logLevelForPackage` | This is a mandatory parameter<br>Log level assigned to the package | **Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF |
| `logSubscriberInfo` | This is an optional parameter<br>Subscriber info logging setting | **Data Type:** String<br>**Range:** ENABLED, DISABLED |
| `additionalErrorLogging` | This is an optional parameter<br>Additional error logging setting | **Data Type:** String<br>**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 3-21 Egress Gateway**

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

**Table 3-21 (Cont.) Egress Gateway**

| Parameter Name | Description | Details |
|---|---|---|
| packageLogLevel | This is a mandatory parameter<br>Specific package log settings | **Data Type:** List of Objects<br>**Range**: Each item contains packageName and logLevelForPackage |
| packageName | This is a mandatory parameter<br>Name of the package | **Data Type**: String<br>**Range**: Any valid Java package name |
| logLevelForPackage | This is a mandatory parameter<br>Log level assigned to the package | **Data Type:**String**Range:** TRACE, DEBUG, INFO, WARN, ERROR, OFF |
| logSubscriberInfo | This is an optional parameter<br>Subscriber info logging setting | **Data Type:** String<br>**Range:** ENABLED, DISABLED |
| additionalErrorLogging | This is an optional parameter<br>Additional error logging setting | **Data Type:** String<br>**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 3-22 Alternate Route**

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

**Table 3-22 (Cont.) Alternate Route**

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

### 6. 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 3-23 NRF Client - Management**

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

### 7. 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 3-24 NsAvailability**

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

**8. 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 3-25 NsAuditor**

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

### 9. 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 3-26 NsConfig**

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

### 10. 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 3-27 NsSelection**

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

### 11. 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 3-28 NsSubscription**

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

# 3.9 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 3-29 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)<br>**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)<br>**HTTP Method:** PUT |

### Request or Response Body Parameters

**Table 3-30 configmapApplicationConfig.profile.appProfiles Parameters**

| Parameter | Description | Details |
|---|---|---|
| `nfInstanceId` | This is a mandatory parameter.<br>Unique identifier for the network function instance. | **Data Type:** String<br>**Range:** UUID |
| `nfType` | This is a mandatory parameter.<br>Type of the network function (for example, NSSF). | **Data Type:** String<br>**Range:** Predefined types (for example, NSSF) |

**Table 3-30 (Cont.) configmapApplicationConfig.profile.appProfiles Parameters**

| Parameter | Description | Details |
|---|---|---|
| `nfStatus` | This is a mandatory parameter.<br>Status of the network function. | **Data Type:** String<br>**Range:** REGISTERED, UNREGISTERED, etc. |
| `heartBeatTimer` | This is a conditional parameter.<br>Time in seconds expected between two consecutive heartbeat messages from an NF Instance to the NRF.<br>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.<br>It shall be included in responses from NRF to registration requests (PUT) or in NF profile updates (PUT or PATCH).<br>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<br>**Range:** 1-3600 seconds |
| `fqdn` | This is a conditional parameter.<br>Fully Qualified Domain Name of the network function. | **Data Type:** String<br>**Range:** Valid domain |
| `priority` | This is an optional parameter.<br>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.<br>The NRF may overwrite the received priority value when exposing an NFProfile with the NRF Discovery service. | **Data Type:** Integer<br>**Range:** 0-65535 |
| `capacity` | This is an optional parameter.<br>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<br>**Range:** 0-65535 |
| `load` | This is an optional parameter.<br>Dynamic load information, ranged from 0 to 100, indicates the current load percentage of the NF. | **Data Type:** Integer<br>**Range:** 0-100 |
| `plmnList.mcc` | This is a conditional parameter.<br>Mobile Country Code (MCC) of the PLMN. | **Data Type:** String<br>**Range:** 3-digit numeric code |
| `plmnList.mnc` | This is a conditional parameter.<br>Mobile Network Code (MNC) of the PLMN. | **Data Type:** String<br>**Range:** 2- or 3-digit numeric code |

**Table 3-30 (Cont.) configmapApplicationConfig.profile.appProfiles Parameters**

| Parameter | Description | Details |
|---|---|---|
| nfSetIdList | This is a conditional parameter.<br>NF Set ID defined in clause 28.12 of 3GPP TS 23.003 [12].<br>At most one NF Set ID shall be indicated per PLMN of the NF. | **Data Type:** array(NfSetId)<br>**Range:** Valid String |
| locality | This is an optional parameter.<br>Operator defined information about the location of the NF instance (for example, geographic location, data center). | **Data Type:** String<br>**Range:** Valid location String |
| nfServices.serviceInstanceId | This is a mandatory parameter.<br>Unique ID of the service instance within a given NF Instance. | **Data Type:** String<br>**Range:** UUID |
| nfServices.serviceName | This is a mandatory parameter.<br>Name of the service provided by the NF. | **Data Type:** String<br>**Range:** Predefined service names |
| nfServices.versions.expiry | This is an optional parameter.<br>Expiry date and time of the NF service. | **Data Type:** DateTime<br>**Range:** ISO 8601 timestamp or null |
| nfServices.versions.apiFullVersion | This is a mandatory parameter.<br>Full version number of the API as specified in clause 4.3.1 of 3GPP 29.501. | **Data Type:** String<br>**Range:** x.y.z (for example, 1.0.0) |
| nfServices.versions.apiVersionInUri | This is a mandatory parameter.<br>Version of the service instance to be used in the URI for accessing the API (for example, "v1"). | **Data Type:** String<br>**Range:** Version String (for example, v1) |
| nfServices.scheme | This is a mandatory parameter.<br>Communication scheme (for example, HTTP, HTTPS). | **Data Type:** String(UriScheme)<br>**Range:** http, https |
| nfServices.nfServiceStatus | This is a mandatory parameter.<br>Status of the NF Service Instance | **Data Type:** String<br>**Range:** REGISTERED, UNREGISTERED |
| nfServices.fqdn | This is an optional parameter.<br>Fully Qualified Domain Name of the service endpoint. | **Data Type:** String<br>**Range:** Valid domain String |
| nfServices.interPlmnFqdn | This is a conditional parameter.<br>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.<br>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<br>**Range**: Valid domain String or null |
| nfServices.ipEndPoints.ipv4Address | This is an optional parameter.<br>IPv4 address of the service endpoint. | **Data Type:** String<br>**Range:** Valid IPv4 address |

**Table 3-30 (Cont.) configmapApplicationConfig.profile.appProfiles Parameters**

| Parameter | Description | Details |
|---|---|---|
| `nfServices.ipEndPoints.transport` | This is an optional parameter.<br>Transport protocol used by the service (for example, TCP). | **Data Type:** String<br>**Range:** TCP, UDP |
| `nfServices.ipEndPoints.port` | This is an optional parameter.<br>Port number for the service endpoint. | **Data Type:** Integer<br>**Range:** 1-65535 |
| `nfServices.allowedNfTypes` | This is an optional parameter.<br>Type of the NFs allowed to access the service instance.<br>The absence of this attribute indicates that any NF type is allowed to access the service instance.<br>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<br>**Range:** Valid NF types |
| `nfServices.priority` | This is an optional parameter.<br>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.<br>The NRF may overwrite the received priority value when exposing an NFProfile to the NRF Discovery service. | **Data Type:** Integer<br>**Range:** 0-65535 |
| `nfServices.capacity` | This is an optional parameter.<br>Static capacity information in the range of 0-65535, expressed as a weight relative to other services of the same type. | **Data Type:** Integer<br>**Range:** 0-65535 |
| `nfServices.load` | This is an optional parameter.<br>Dynamic load information, ranging from 0 to 100, indicates the current load percentage of the NF Service. | **Data Type:** Integer<br>**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
      }
    ]
  }
]
```

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}*<br>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.<br>**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" }`<br><br>or<br><br>`{ "instanceIdList": [ { "instanceId": "4bc0c762-0212-416a-bd94-b7f1fb348bd4", "certName": "4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt", "kSecretName": "oauthsecret", "certAlgorithm": "ES256" } ], "oauthValidationMode": "INSTANCEID_ONLY" }`<br><br>**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 fallback mode where it checks for `keyId` in token, If token is present then validation mode is `KID_ONLY`, else it will fallback 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.<br>This parameter specifies whether the feature is enabled or disabled. | **Data Type**: Boolean<br>**Range**: true or false<br>**Example Value**: false |
| `errorCodeSeriesId` | This is an optional parameter.<br>Specifies the error list IDs | **Data Type**: String<br>**Range**: NA<br>**Example Value**: NA |
| `configuration.nfType` | This is a mandatory parameter.<br>Specifies the type of network function | **Data Type**: String<br>**Range**: NA<br>**Example Value**: NSSF |

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

| Parameter | Description | Details |
|---|---|---|
| configuration.nfInstanceId | This is a mandatory parameter.<br>This parameter represents the UUID of the NSSF deployment that is used to generate the Server Header. | **Data Type**: String<br>**Range**: NA<br>**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 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.<br>Error name. | **Data Type:** string |
| `errorCause` | This is an optional parameter.<br>`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 |

| Field Name | Description | Details |
|---|---|---|
| errorCode | This is a mandatory parameter.<br>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.<br>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.<br>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.<br>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 |
| retry-after | This is an optional parameter.<br>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"}]'
```

## 5.1.3 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, `ocDiscardPolicies` 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.<br>Name of the discarded policy.<br>**Note**: name must be the value configured in `policyName` under `ocpolicymapping`. | **Data Type:** string |
| `policies.action` | This is an optional parameter.<br>Defines the action to be taken on selected requests rejection based on error code. | **Data Type:** string<br>**Example Value:** `RejectWithErrorCode` |
| `policies.errorCodeProfile` | This is an optional parameter.<br>The error code profiles. | **Data Type:** string |
| `policies.level` | This is an optional parameter.<br>Defines the overload level. | **Data Type:** string |
| `policies.value` | This is a mandatory parameter.<br>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.<br>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"}]}]'
```


## 5.1.4 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
- PATCH: 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 5-2 Request or Response Body Parameters**

| Field Name | Description | Details |
|---|---|---|
| `enabled` | This is a mandatory parameter.<br>To enable or disable the Overload Control feature. Set values to `true` or `false`, respectively. | **Data Type:** boolean |

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

| Field Name | Description | Details |
|---|---|---|
| `mappings` | This is an optional parameter.<br>**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.<br>The discard policy entry determines a mapping between the service and discards policy name per service.<br>**Note**: If a value for `mappings.svcName` is provided, then `mappings.policyName` is mandatory. | **Data Type:** string |
| `mappings.svcName` | This is a conditional parameter.<br>The service entry to determine a mapping between service and discard policy name per `service.svcName` must be added in the following format:<br><deployment-name>-<servicename><br>**Note**: servicename is fixed and cannot be changed.<br>**Note**: If a value for `mappings.policyName` is provided, then `mappings.svcName` is mandatory. | **Data Type:** string |
| `samplingPeriod` | This is a mandatory parameter.<br>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}'
```

## 5.1.5 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 5-3 Request or Response Body Parameters**

| Parameter | Description | Details |
|---|---|---|
| `id` | This is a mandatory parameter.<br>Indicates the error code identifier. | **Data Type**: string |
| `errorCodeSeries` | This is a mandatory parameter.<br>List the error codes for a specific service.<br>**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 |
| `exceptionList` | This is an optional parameter.<br>Lists the configurable exception or error for an error scenario in Ingress Gateway. | **Data Type**: string<br>**Range:**<br>• `ConnectionTimeout`<br>• `RequestTimeout`<br>• `UnknownHostException`<br>• `ConnectException`<br>• `NotFoundException` |

**Table 5-3 (Cont.) Request or Response Body Parameters**

| Parameter | Description | Details |
|---|---|---|
| errorSet | This is a mandatory parameter.<br>Possible values for "errorSet" attribute: 5xx, 4xx, 3xx | **Data Type**: string |
| errorCodes | This is a mandatory parameter.<br>Possible values include all error codes in the respective HttpSeries value assigned for "errorSet".<br>**Note**: Use single value of "-1" if all error codes in that HttpSeries are to be considered.<br>For example:<br>• If the series is 4XX, the values should be between 400-499.<br>• If the series is 5XX, the values should be between 500-599.<br>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}/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
```

## 5.1.6 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 5-4 Request or Response Body Parameters**

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

## 5.1.7 Perf-Info REST APIs

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

### 5.1.7.1 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 5-5 Request or Response Body Parameters**

| Parameter | Description | Details |
|---|---|---|
| `svcName` | This is a mandatory parameter.<br>Name of the backend service. | **Data Type:** string |
| `metricsThresholdList.metricsName` | This is a mandatory parameter.<br>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.<br>The `metricsThresholdList.levelThresholdList.level` is considered to be breached if the `metricsThresholdList.metricsName` value has crossed this onset value. | **Data Type:** integer |
| `metricsThresholdList.levelThresholdList.level` | This is a mandatory parameter.<br>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.<br>The overload condition is considered as cleared this level, if the `metricsThresholdList.metricsName` is below the abatement value. | **Data Type:** integer |

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

| Parameter | Description | Details |
|---|---|---|
| `metricsThresholdList.levelThresholdList` | This is a mandatory parameter.<br>List of threshold values. | **Data Type:** array |
| `metricsThresholdList` | This is a mandatory parameter.<br>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'
```

## 5.1.8 Configuration To Check If Overload Control is Enabled

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

### 5.1.8.1 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`

### 5.1.8.2 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
    }
  ]
}
```

### 5.1.8.3 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
    }
  ]
}
```

## 5.1.8.4 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
}
```

## 5.1.8.5 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-
efABCDEF-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%221%22%2C
%0A%09%09%09%22sd%22%3A%22EABB01%22%0A%09%09%7D%2C%0A%09%22roamingIndication%2
2%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 5-6 PeerConfiguration**

| Parameter | Data Type | Constraints | Description |
|---|---|---|---|
| `id` | string | Unique value in peer configuration. | Identifier for Peer |
| `host` | string | NA | Host details of the Peer. In case of SLF, it is SCP.<br>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. |

**Table 5-6 (Cont.) PeerConfiguration**

| Parameter | Data Type | Constraints | Description |
|---|---|---|---|
| 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 5-7 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 5-8 Peer Identifier Configuration**

| Parameter | Data Type | Constraints | Description |
|---|---|---|---|
| `priority` | integer | Priority should be unique | Priority of peer to be used in a peer set. |
| `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 5-9 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 5-10 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. |

**Table 5-10 (Cont.) Error Action Sets**

| Parameter | Data Type | Constraints | Description |
|---|---|---|---|
| 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 5-11 RoutesConfiguration**

| Parameter | Data Type | Constraints | Description |
|---|---|---|---|
| `id` | string | Unique value of Route | Route configuration identifier.<br>**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" }`<br>Keep `enabled` value to `true` for SBI routing to work.<br>`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 .<br>`false`: Means no change occurred to the scheme. |
| `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).<br>`false`: Perform as per provided scheme. |

**Table 5-11 (Cont.) RoutesConfiguration**

| Parameter | Data Type | Constraints | Description |
|---|---|---|---|
| predicates | Predicate Structure. See Description | NA | Path predicate details for matching SLF path mapped to this SBIRoute rule.<br>**Sample value**:<br>"predicates": [{ "args": {"pattern": "/nudr-group-id-map/v1/nf-group-ids"}, "name": "Path" }]<br>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.<br>**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 5-12 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.<br>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.<br>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 5-13 Request or Response Body Parameters**

| Parameter | Description | Details |
|---|---|---|
| `enabled` | This is a mandatory parameter.<br>This parameter specifies whether the feature is enabled or disabled. | **Data Type**: Boolean<br>**Range**: true or false<br>**Example Value**: false |
| `nfType` | This is a mandatory parameter.<br>This parameter holds the `nfType` that is used to generate the User-Agent Header. | **Data Type**: String<br>**Range**: NA<br>**Example Value**: NSSF |
| `nfInstanceId` | This is a mandatory parameter.<br>This parameter represents the UUID of the NSSF deployment that is used to generate the User-Agent Header. | **Data Type**: String<br>**Range**: NA<br>**Example Value**: Valid NSSF Instance ID |
| `addFqdnToHeader` | This is a mandatory parameter.<br>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<br>**Range**: true or false<br>**Example Value**: false |
| `nfFqdn` | This is an optional parameter.<br>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<br>**Range**: NA<br>**Example Value**: "nssf.oracle.com" |
| `overwriteHeader` | This is a mandatory parameter.<br>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`.<br>**Note:**<br>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<br>**Range**: true or false<br>**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 Appendix A - Common Error Responses for Managed Objects

### GET Method Responses

| HTTP Method | Response Code | Description | Notes |
|---|---|---|---|
| GET | 200 OK | Success response | - |
| GET | 404 Not Found | Incorrect API Path | The requested API path is incorrect. |
| GET | 403 Forbidden | PLMN Not Supported | • Requested PLMN is not supported.<br>• The system does not support the requested PLMN configuration. |
| GET | 500 Internal Server Error | • Database Connectivity Issue<br>• Any database Exception<br>• Unknown Issue (Uncaught Exception) | • Database connectivity issues.<br>• System failed to access or query the database. |

### POST Method Error Scenarios

| Method | Response Code | Description | Notes |
|---|---|---|---|
| POST | 200 OK | Success Response | - |
| POST | 400 Bad Request | • Mandatory parameter missing<br>• Mandatory parameter validation issue (incorrect format/values)<br>• Optional parameter validation issue (incorrect format/values) | - |

| Method | Response Code | Description | Notes |
|---|---|---|---|
| POST | 409 Conflict | • A dependent managed object that is required to complete the operation has not been configured or does not exist<br>• A dependent managed object that must be deleted first still exists<br>• A conflicting managed object entry already exists in the system, preventing the current MO from being configured<br>• An SNSSAI cannot be referenced (as supported or barred) per TAI unless it is already configured under the corresponding PLMN configuration | - |
| POST | 415 Unsupported Media Type | Invalid Content-Type | - |
| POST | 500 Internal Server Error | • Database Connectivity Issue<br>• Any database Exception<br>• Unknown Issue (Uncaught Exception) | • Database connection issues.<br>• The system failed to access or query the database. |

### PUT Method Error Scenarios

| HTTP Method | Response Code | Description | Notes |
|---|---|---|---|
| PUT | 200 OK | Success Response | - |
| PUT | 400 Bad Request | • Mandatory parameter missing<br>• Mandatory parameter validation issue (incorrect format/values)<br>• Optional parameter validation issue (incorrect format/values) | - |

| HTTP Method | Response Code | Description | Notes |
|---|---|---|---|
| PUT | 409 Conflict | • A dependent managed object that is required to complete the operation has not been configured or does not exist<br>• A dependent managed object that must be deleted first still exists<br>• A conflicting managed object entry already exists in the system, preventing the current MO from being configured<br>• An SNSSAI cannot be referenced (as supported or barred) per TAI unless it is already configured under the corresponding PLMN configuration | • A dependent managed object that is required to complete the operation has not been configured or does not exist.<br>• A dependent managed object that must be deleted first still exists.<br>• A conflicting managed object entry already exists in the system, preventing the current MO from being configured.<br>• An SNSSAI cannot be referenced (as supported or barred) per TAI unless it is already configured under the corresponding PLMN configuration. |
| PUT | 415 Unsupported Media Type | Invalid Content-Type | • The server does not support the media type.<br>• Only `application/json` content type is supported. |
| PUT | 404 Not Found | Resource not found | - |
| PUT | 500 Internal Server Error | • Database Connectivity Issue<br>• Any database Exception<br>• Unknown Issue (Uncaught Exception) | • Database connection issues.<br>• The system failed to access or query the database. |

### DELETE Method Error Scenarios

| HTTP Method | Response Code | Description | Notes |
|---|---|---|---|
| DELETE | 204 No Content | Successful Managed Object Deletion | • The requested resource was deleted successfully.<br>• No content is returned in the response body. |
| DELETE | 404 Not Found | Resource not found | • Correct API path, but the HTTP method (DELETE) is not supported.<br>• Invalid method for the requested endpoint. |
| DELETE | 500 Internal Server Error | • Database Connectivity Issue<br>• Any database Exception<br>• Unknown Issue (Uncaught Exception) | • Database connection issues.<br>• The system failed to access or query the database. |

7

# Appendix B - Important Guidelines for Configuring the Managed Objects

Given below are some important guidelines to following while configuring the managed objects (MOs) explained in this document:

1. **Use REST APIs or CNCC GUI:** Direct database manipulations are not allowed. Users must use the REST APIs or the CNCC GUI to configure the managed objects.
2. **PLMN-specific configuration:** For the managed objects PlmnConfig, BarredSlicesMapping, and SupportedSlicesMapping, data must be configured per PLMN. Each request must contain only a single PLMN-related information. To create another PLMN-related configuration, a separate request must be sent for that particular managed object.
3. **Dependencies between managed objects:** The following dependencies must be observed:
   a. To configure a PlmnConfig managed object, ensure that:
      - SystemOptions with the particular PLMN specified in PlmnInfo is configured.
      - Required NsiProfiles specified in PlmnInfo are created.
   b. To configure a SupportedSlicesMapping managed object, ensure that:
      - PlmnConfig with the particular PLMN specified in PlmnTacList is configured.
      - Required NsiProfiles specified in `nsiInformationList` of `supportedSnssaiList` are created.
   c. To configure a BarredSlicesMapping managed object, ensure that a PlmnConfig with the particular PLMN specified in PlmnTacList is configured.
4. **Order of configuring the managed objects:** The following order must be followed when configuring the managed objects:
   a. Configure SystemOptions and NsiProfiles.
   b. Configure PlmnConfig.
   c. Configure SupportedSlicesMapping and/or BarredSlicesMapping.
5. **Order of deleting the managed objects:** The following order must be followed when deleting the managed objects:
   a. Delete SupportedSlicesMapping and/or BarredSlicesMapping.
   b. Delete PlmnConfig.
   c. Delete NsiProfiles and/or SystemOptions.