3GPP TS 29.510 V15.1.0 (2018-09)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Core Network and Terminals;

5G System;

Network Function Repository Services;

Stage 3

(Release 15)



The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Report is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

3GPP, 5GS

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2018, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword 7

Introduction 7

1 Scope 8

2 References 8

3 Definitions and abbreviations 9

3.1 Definitions 9

3.2 Abbreviations 9

4 Overview 9

5 Services Offered by the NRF 10

5.1 Introduction 10

5.2 Nnrf\_NFManagement Service 10

5.2.1 Service Description 10

5.2.2 Service Operations 10

5.2.2.1 Introduction 10

5.2.2.2 NFRegister 11

5.2.2.2.1 General 11

5.2.2.2.2 NF (other than NRF) registration to NRF 11

5.2.2.2.3 NRF registration to another NRF 12

5.2.2.3 NFUpdate 12

5.2.2.3.1 General 12

5.2.2.3.2 NF Heart-Beat 14

5.2.2.4 NFDeregister 14

5.2.2.4.1 General 14

5.2.2.5 NFStatusSubscribe 15

5.2.2.5.1 General 15

5.2.2.5.2 Subscription to NF Instances in the same PLMN 15

5.2.2.5.3 Subscription to NF Instances in a different PLMN 16

5.2.2.6 NFStatusNotify 17

5.2.2.6.1 General 17

5.2.2.6.2 Notification from NRF in the same PLMN 17

5.2.2.6.3 Notification from NRF in a different PLMN 17

5.2.2.7 NFStatusUnSubscribe 18

5.2.2.7.1 General 18

5.2.2.7.2 Subscription removal in the same PLMN 18

5.2.2.7.3 Subscription removal in a different PLMN 18

5.3 Nnrf\_NFDiscovery Service 19

5.3.1 Service Description 19

5.3.2 Service Operations 19

5.3.2.1 Introduction 19

5.3.2.2 NFDiscover 20

5.3.2.2.1 General 20

5.3.2.2.2 Service Discovery in the same PLMN 20

5.3.2.2.3 Service Discovery in a different PLMN 20

5.3.2.2.4 Service Discovery with intermediate redirecting NRF 21

5.3.2.2.5 Service Discovery with intermediate forwarding NRF 22

5.4 OAuth2 Authorization Service 23

5.4.1 Service Description 23

5.4.2 Service Operations 23

5.4.2.1 Introduction 23

5.4.2.2 Access Token Request 23

5.4.2.2.1 General 23

6 API Definitions 24

6.1 Nnrf\_NFManagement Service API 24

6.1.1 API URI 24

6.1.2 Usage of HTTP 24

6.1.2.1 General 24

6.1.2.2 HTTP Standard Headers 24

6.1.2.2.1 General 24

6.1.2.2.2 Content type 24

6.1.2.3 HTTP custom headers 25

6.1.2.3.1 General 25

6.1.3 Resources 25

6.1.3.1 Overview 25

6.1.3.2 Resource: nf-instances (Store) 26

6.1.3.2.1 Description 26

6.1.3.2.2 Resource Definition 26

6.1.3.2.3 Resource Standard Methods 26

6.1.3.2.3.1 GET 26

6.1.3.2.4 Resource Custom Operations 27

6.1.3.3 Resource: nf-instance (Document) 27

6.1.3.3.1 Description 27

6.1.3.3.2 Resource Definition 27

6.1.3.3.3 Resource Standard Methods 27

6.1.3.3.3.1 GET 27

6.1.3.3.3.2 PUT 28

6.1.3.3.3.3 PATCH 29

6.1.3.3.3.4 DELETE 30

6.1.3.4 Resource: subscriptions (Collection) 30

6.1.3.4.1 Description 30

6.1.3.4.2 Resource Definition 30

6.1.3.4.3 Resource Standard Methods 30

6.1.3.4.3.1 POST 30

6.1.3.5 Resource: subscription (Document) 31

6.1.3.5.1 Description 31

6.1.3.5.2 Resource Definition 31

6.1.3.5.3 Resource Standard Methods 31

6.1.3.5.3.1 DELETE 31

6.1.4 Custom Operations without associated resources 32

6.1.5 Notifications 32

6.1.5.1 General 32

6.1.5.2 NF Instance Status Notification 32

6.1.5.2.1 Description 32

6.1.5.2.2 Notification Definition 32

6.1.6 Data Model 33

6.1.6.1 General 33

6.1.6.2 Structured data types 35

6.1.6.2.1 Introduction 35

6.1.6.2.2 Type: NFProfile 36

6.1.6.2.3 Type: NFService 38

6.1.6.2.4 Type: DefaultNotificationSubscription 39

6.1.6.2.5 Type: IpEndPoint 39

6.1.6.2.6 Type: UdrInfo 39

6.1.6.2.7 Type: UdmInfo 40

6.1.6.2.8 Type: AusfInfo 40

6.1.6.2.9 Type: SupiRange 40

6.1.6.2.10 Type: IdentityRange 41

6.1.6.2.11 Type: AmfInfo 41

6.1.6.2.12 Type: SmfInfo 42

6.1.6.2.13 Type: UpfInfo 42

6.1.6.2.14 Type: SnssaiUpfInfoItem 42

6.1.6.2.15 Type: DnnUpfInfoItem 42

6.1.6.2.16 Type: SubscriptionData 43

6.1.6.2.17 Type: NotificationData 44

6.1.6.2.18 Type: NFRegistrationData 44

6.1.6.2.19 Type: NFServiceVersion 44

6.1.6.2.20 Type: PcfInfo 44

6.1.6.2.21 Type: BsfInfo 45

6.1.6.2.22 Type: Ipv4AddressRange 45

6.1.6.2.23 Type: Ipv6PrefixRange 45

6.1.6.2.24 Type: InterfaceUpfInfoItem 45

6.1.6.2.25 Type: UriList 45

6.1.6.2.26 Type: N2InterfaceAmfInfo 46

6.1.6.2.27 Type: TaiRange 46

6.1.6.2.28 Type: TacRange 46

6.1.6.3 Simple data types and enumerations 47

6.1.6.3.1 Introduction 47

6.1.6.3.2 Simple data types 47

6.1.6.3.3 Enumeration: NFType 47

6.1.6.3.4 Enumeration: NotificationType 47

6.1.6.3.5 Enumeration: TransportProtocol 48

6.1.6.3.6 Enumeration: NotificationEventType 48

6.1.6.3.7 Enumeration: NFStatus 48

6.1.6.3.8 Enumeration: DataSetId 48

6.1.6.3.9 Enumeration: UPInterfaceType 48

6.1.6.3.10 Relation Types 48

6.1.6.3.10.1 General 48

6.1.6.3.11 Enumeration: ServiceName 49

6.1.6.3.12 Enumeration: NFServiceStatus 49

6.1.7 Error Handling 50

6.1.7.1 General 50

6.1.7.2 Protocol Errors 50

6.1.7.3 Application Errors 50

6.1.8 Security 50

6.2 Nnrf\_NFDiscovery Service API 50

6.2.1 API URI 50

6.2.2 Usage of HTTP 50

6.2.2.1 General 50

6.2.2.2 HTTP Standard Headers 51

6.2.2.2.1 General 51

6.2.2.2.2 Content type 51

6.2.2.2.3 Cache-Control 51

6.2.2.2.4 ETag 51

6.2.2.2.5 If-None-Match 51

6.2.2.3 HTTP custom headers 51

6.2.2.3.1 General 51

6.2.3 Resources 51

6.2.3.1 Overview 51

6.2.3.2 Resource: nf-instances (Store) 52

6.2.3.2.1 Description 52

6.2.3.2.2 Resource Definition 52

6.2.3.2.3 Resource Standard Methods 52

6.2.3.2.3.1 GET 52

6.2.3.2.4 Resource Custom Operations 55

6.2.4 Custom Operations without associated resources 55

6.2.5 Notifications 55

6.2.6 Data Model 55

6.2.6.1 General 55

6.2.6.2 Structured data types 56

6.2.6.2.1 Introduction 56

6.2.6.2.2 Type: SearchResult 56

6.2.6.2.3 Type: NFProfile 57

6.2.6.2.4 Type: NFService 58

6.2.6.3 Simple data types and enumerations 58

6.2.6.3.1 Introduction 58

6.2.6.3.2 Simple data types 59

6.2.7 Error Handling 59

6.2.7.1 General 59

6.2.7.2 Protocol Errors 59

6.2.7.3 Application Errors 59

6.2.8 Security 59

6.3 OAuth2.0 Authorization Service API 59

6.3.1 General 59

6.3.2 API URI 59

6.3.3 Usage of HTTP 60

6.3.3.1 General 60

6.3.3.2 HTTP standard headers 60

6.3.3.2.1 General 60

6.3.3.2.2 Content type 60

6.3.3.3 HTTP custom headers 60

6.3.3.3.1 General 60

6.3.4 Custom Operation without Associated Resources 60

6.3.4.1 Overview 60

6.3.4.2 Operation: Access Token Request 61

6.3.4.2.1 Description 61

6.3.4.2.2 Operation Definition 61

6.3.5 Data Model 61

6.3.5.1 General 61

6.3.5.2 Structured data types 62

6.3.5.2.1 Introduction 62

6.3.5.2.2 Type: AccessTokenReq 62

6.3.5.2.3 Type: AccessTokenRsp 62

6.3.5.2.4 Type: AccessTokenClaims 63

6.3.5.3 Simple data types and enumerations 63

6.3.5.3.1 Introduction 63

6.3.5.3.2 Simple data types 63

6.3.5.3.3 Enumeration: GrantType 63

Annex A (normative): OpenAPI specification 63

A.1 General 63

A.2 Nnrf\_NFManagement API 63

A.3 Nnrf\_NFDiscovery API 76

A.4 NRF OAuth2 Authorization 80

Annex B (informative): Change history 83

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# Introduction

This clause is optional. If it exists, it is always the second unnumbered clause.

# 1 Scope

The present document specifies the stage 3 protocol and data model for the Nnrf Service Based Interface. It provides stage 3 protocol definitions and message flows, and specifies the API for each service offered by the NRF.

The 5G System stage 2 architecture and procedures are specified in 3GPP TS 23.501 [2] and 3GPP TS 23.502 [3].

The Technical Realization of the Service Based Architecture and the Principles and Guidelines for Services Definition are specified in 3GPP TS 29.500 [4] and 3GPP TS 29.501 [5].

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[3] 3GPP TS 23.502: "Procedures for the 5G System; Stage 2".

[4] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

[5] 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition; Stage 3".

[6] 3GPP TS 29.518: "5G System; Access and Mobility Management Services; Stage 3".

[7] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces; Stage 3".

[8] ECMA-262: "ECMAScript® Language Specification", <https://www.ecma-international.org/ecma-262/5.1/>.

[9] IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[10] OpenAPI Initiative, "OpenAPI 3.0.0 Specification", <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md>.

[11] IETF RFC 7807: "Problem Details for HTTP APIs".

[12] 3GPP TS 23.003: "Numbering, Addressing and Identification".

[13] IETF RFC 6902: "JavaScript Object Notation (JSON) Patch".

[14] IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".

[15] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

[16] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[17] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[18] IETF RFC 4122: "A Universally Unique IDentifier (UUID) URN Namespace".

[19] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

[20] IETF RFC 7234: "Hypertext Transfer Protocol (HTTP/1.1): Caching".

[21] 3GPP TS 29.244: "Interface between the Control Plane and the User Plane Nodes; Stage 3".

[22] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

[23] IETF RFC 2782: "A DNS RR for specifying the location of services (DNS SRV)".

[24] IETF RFC 7515: "JSON Web Signature (JWS)".

[25] IETF RFC 7519: "JSON Web Token (JWT)".

[26] W3C HTML 4.01 Specification, <https://www.w3.org/TR/2018/SPSD-html401-20180327/>.

[27] 3GPP TS 23.527: "5G System; Restoration Procedures; Stage 2".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

5GC 5G Core Network

CHF Charging Function

NF Network Function

NRF NF Repository Function

# 4 Overview

The Network Function (NF) Repository Function (NRF) is the network entity in the 5G Core Network (5GC) supporting the following functionality:

- Maintains the NF profile of available NF instances and their supported services;

- Allows other NF instances to subscribe to, and get notified about, the registration in NRF of new NF instances of a given type;

- Supports service discovery function. It receives NF Discovery Requests from NF instances, and provides the information of the available NF instances fulfilling certain criteria (e.g., supporting a given service).

Figures 4-1 shows the reference architecture for the 5GC, with focus on the NRF:



Figure 4-1: 5G System architecture

For the sake of clarity, the NRF is never depicted in reference point representation figures, given that the NRF interacts with every other NF in the 5GC. As an exception, in the roaming case, the reference point between the vNRF and the hNRF is named as N27. The reference point name of N27 is used only for representation purposes, but its functionality is included in the services offered by the Nnrf Service-Based Interface.

# 5 Services Offered by the NRF

## 5.1 Introduction

The NRF offers to other NFs the following services:

- Nnrf\_NFManagement

- Nnrf\_NFDiscovery

- OAuth2 Authorization

## 5.2 Nnrf\_NFManagement Service

### 5.2.1 Service Description

The Nnrf\_NFManagement service allows a Network Function Instance in the serving PLMN to register, update or deregister its profile in the NRF.

It also allows an NF to subscribe to be notified of newly registered NF Instances along with their NF services.

The NF profile consists of general parameters of the NF Instance, and also the parameters of the different services exposed by the NF Instance.

### 5.2.2 Service Operations

#### 5.2.2.1 Introduction

The services operations defined for the Nnrf\_NFManagement service are as follows:

- NFRegister: It allows an NF Instance to register its NF profile in the NRF; it includes the registration of the general parameters of the NF Instance, together with the list of services exposed by the NF Instance. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFUpdate: It allows an NF Instance to replace, or update partially, the parameters of its NF profile (including the parameters of the associated services) in the NRF; it also allows to add or delete individual services offered by the NF Instance. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFDeregister: It allows an NF Instance to deregister its NF profile in the NRF, including the services offered by the NF Instance. This service operation is not allowed to be invoked from an NRF in a different PLMN.

- NFStatusSubscribe: It allows an NF Instance to subscribe to changes on the status of NF Instances registered in NRF. This service operation can be invoked by an NF Instance in a different PLMN (via the local NRF in that PLMN).

- NFStatusNotify: It allows the NRF to notify subscribed NF Instances of changes on the status of NF Instances. This service operation can be invoked directly between the NRF and an NF Instance in a different PLMN (without involvement of the local NRF in that PLMN).

- NFStatusUnsubscribe: It allows an NF Instance to unsubscribe to changes on the status of NF Instances registered in NRF. This service operation can be invoked by an NF Instance in a different PLMN (via the local NRF in that PLMN).

NOTE: The "change of status" of the NFStatus service operations can imply a request to be notified of newly registered NF Instances in NRF, or to be notified of profile changes of a specific NF Instance, or to be notified of the deregistration of an NF Instance.

The NFStatusSubscribe / NFstatusNotify / NFStatusUnsubscribe operations can be invoked by an NF Service Consumer (i.e., "source NF") requesting to be notified about events (registration, deregistration, profile change) related to an NF instance (i.e., "target NF") located in the same PLMN, or in a different PLMN.

In the description of these operations in subclauses 5.2.2.5, 5.2.2.6 and 5.2.2.7, when the NF instances are located in the same PLMN, both source NF and target NF are said to be located in the "Serving PLMN" but, in the general case, the functionality is not restricted to the PLMN that is serving a given UE, and it shall be applicable as well to any scenario in which source NF and target NFs belong to the same PLMN.

When source NF and target NF are located in different PLMNs, the source NF is said to be in the "Serving PLMN", and the target NF (and the NRF where such NF is registered) is said to be in the "Home PLMN", similarly to the scenarios described in 3GPP TS 23.502 [2], but the functionality shall be equally applicable to any scenario between any pair of PLMNs (e.g. with the source NF in the Home PLMN and the target NF in the Serving PLMN).

#### 5.2.2.2 NFRegister

##### 5.2.2.2.1 General

This service operation is used:

- to register an NF in the NRF by providing the NF profile of the requesting NF to the NRF, and the NRF marks the requesting NF as available to be discovered by other NFs;

- to register services associated to an existing NF Instance;

- to register NRF information in another NRF, and this information is used for forwarding or redirecting service discovery request.

##### 5.2.2.2.2 NF (other than NRF) registration to NRF



Figure 5.2.2.2.1-1: NF Instance Registration

1. The NF Service Consumer shall send a PUT request to the resource URI representing the NF Instance. The URI is determined by the NF Instance. The variable {nfInstanceID} represents an identifier, provided by the NF Service Consumer, that shall be globally unique inside the PLMN of the NRF where the NF is being registered. The format of the NF Instance ID shall be a Universally Unique Identifier (UUID) version 4, as described in IETF RFC 4122 [18].

EXAMPLE: UUID version 4: "4947a69a-f61b-4bc1-b9da-47c9c5d14b64"

The payload body of the PUT request shall contain a representation of the NF Instance to be created.

2. On success, "201 Created" shall be returned, the payload body of the PUT response shall contain the representation of the created resource and the "Location" header shall contain the URI of the created resource. Additionally, the NRF returns a "heart-beat timer" containing the number of seconds expected between two consecutive heart-beat messages from an NF Instance to the NRF (see subclause 5.2.2.3.2).

If the registration of the NF instance fails at the NRF due to errors in the encoding of the NFProfile JSON object, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the registration of the NF instance fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

The NRF shall allow the registration of a Network Function instance with any of the NF types described in subclause 6.1.6.3.3, and it shall also allow registration of Network Function instances with custom NF types (e.g., NF type values not defined by 3GPP, or NF type values not defined by this API version).

NOTE: When registering a custom NF in NRF, it is recommended to use a NF type name that prevents collisions with other custom NF type names, or with NF types defined in the future by 3GPP. E.g., prefixing the custom NF type name with the string "CUSTOM\_".

During the registration of a Network Function instance with a custom NF type, the NF instance may provide NF-specific data (in the "customInfo" attribute), that shall be stored by the NRF as part of the NF profile of the NF instance.

##### 5.2.2.2.3 NRF registration to another NRF

The procedure specified in subclause 5.2.2.2.2 applies. Additionally:

a) the registering NRF shall set the nfType to "NRF" in the nfProfile;

b) the registering NRF shall set the nfService to contain "nnrf-disc" and "nnrf-nfm" in the nfProfile;

c) the registering NRF may include udrInfo, udmInfo, ausfInfo, amfInfo, smfInfo, upfInfo, pcfInfo and bsfInfo in the nfProfile, this means the registering NRF is able to provide service for discovery of NFs subject to that information;

d) if the NRF receives an NF registration with the nfType set to "NRF", the NRF shall use the information contained in the nfProfile to target the registering NRF when forwarding or redirecting NF service discovery request.

Editor's Note: The details of the nfProfile used for NRF registration to another NRF is FFS.

#### 5.2.2.3 NFUpdate

##### 5.2.2.3.1 General

This service operation updates the profile of a Network Function previously registered in the NRF by providing the updated NF profile of the requesting NF to the NRF. The update operation may apply to the whole profile of the NF (complete replacement of the existing profile by a new profile), or it may apply only to a subset of the parameters of the profile (including adding/deleting/replacing services to the NF profile).

To perform a complete replacement of the NF Profile of a given NF Instance, the NF Service Consumer shall issue an HTTP PUT request, as shown in Figure 5.2.2.3.1-1:



Figure 5.2.2.3.1-1: NF Profile Complete Replacement

1. The NF Service Consumer shall send a PUT request to the resource URI representing the NF Instance. The payload body of the PUT request shall contain a representation of the NF Instance to be completely replaced in the NRF.

2a. On success, "200 OK" shall be returned, the payload body of the PUT response shall contain the representation of the replaced resource.

2b. If the update of the NF instance fails at the NRF due to errors in the encoding of the NFProfile JSON object, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the update of the NF instance fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

To perform a partial update of the NF Profile of a given NF Instance, the NF Service Consumer shall issue an HTTP PATCH request, as shown in Figure 5.2.2.3.1-2. This partial update shall be used to add/delete/replace individual parameters of the NF Instance, and also to add/delete/replace any of the services (and their parameters) offered by the NF Instance.



Figure 5.2.2.3.1-2: NF Profile Partial Update

1. The NF Service Consumer shall send a PATCH request to the resource URI representing the NF Instance. The payload body of the PATCH request shall contain the list of operations (add/delete/replace) to be applied to the NF Profile of the NF Instance; these operations may be directed to individual parameters of the NF Profile or to the list of services (and their parameters) offered by the NF Instances. In order to leave the NF Profile in a consistent state, all the operations specified by the PATCH request body shall be executed atomically.

2a. On success, "200 OK" shall be returned, the payload body of the PATCH response shall contain the representation of the replaced resource.

2b. If the NF Instance, identified by the "nfInstanceID", is not found in the list of registered NF Instances in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

##### 5.2.2.3.2 NF Heart-Beat

Each NF that has previously registered in NRF shall contact the NRF periodically (heart-beat), by invoking the NFUpdate service operation, in order to show that the NF is still operative.

The time interval at which the NRF shall be contacted is deployment-specific, and it is returned by the NRF to the NF Service Consumer as a result of a successful registration.

When the NRF detects that a given NF has not updated its profile for a configurable amount of time (longer than the heart-beat interval), the NRF changes the status of the NF to SUSPENDED and considers that the NF and its services can no longer be discovered by other NFs via the NFDiscovery service. The NRF notifies NFs subscribed to receiving notifications of changes of the NF Profile that the NF status has been changed to SUSPENDED.



Figure 5.2.2.3.2-1: NF Heart-Beat

1. The NF Service Consumer shall send a PATCH request to the resource URI representing the NF Instance. The payload body of the PATCH request shall contain a "replace" operation on the "nfStatus" attribute of the NF Profile of the NF Instance, and set it to the value "REGISTERED".

In addition, the NF Service Consumer may also provide the load information of the NF, and/or the load information of the NF associated NF services.

2a. On success, "204 No Content" shall be returned.

2b. If the NF Instance, identified by the "nfInstanceID", is not found in the list of registered NF Instances in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

EXAMPLE:

PATCH .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64

Content-Type: application/json-patch+json

[

{ "op": "replace", "path": "/nfStatus", "value": "REGISTERED" },

{ "op": "replace", "path": "/load", "value": 50 }

]

HTTP/2 204 No Content

Content-Location: .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64

#### 5.2.2.4 NFDeregister

##### 5.2.2.4.1 General

This service operation removes the profile of a Network Function previously registered in the NRF.

It is executed by deleting a given resource identified by a "NF Instance ID". The operation is invoked by issuing a DELETE request on the URI representing the specific NF Instance.



Figure 5.2.2.4.1-1: NF Instance Deregistration

1. The NF Service Consumer shall send a DELETE request to the resource URI representing the NF Instance. The request body shall be empty.

2a. On success, "204 No Content" shall be returned. The response body shall be empty.

2b. If the NF Instance, identified by the "nfInstanceID", is not found in the list of registered NF Instances in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

#### 5.2.2.5 NFStatusSubscribe

##### 5.2.2.5.1 General

This service operation is used to:

- create a subscription so an NF Service Consumer can request to be notified when NF Instances of a given set, following certain filter criteria are registered/deregistered in NRF or when their profile is modified;

- create a subscription to a specific NF Instance so an NF Service Consumer can request to be notified when the profile of such NF Instance is modified or when the NF Instance is deregistered from NRF.

##### 5.2.2.5.2 Subscription to NF Instances in the same PLMN

The subscription to notifications on NF Instances is executed creating a new individual resource under the collection resource "subscriptions". The operation is invoked by issuing a POST request on the URI representing the "subscriptions" resource.



Figure 5.2.2.5.2-1: Subscription to NF Instances in the same PLMN

1. The NF Service Consumer shall send a POST request to the resource URI representing the "subscriptions" collection resource. The request body shall include the data indicating the type of notifications that the NF Service Consumer is interested in receiving; it also contains a callback URI, where the NF Service Consumer shall be prepared to receive the actual notification from the NRF (see NFStatusNotify operation in 5.2.2.6) and it may contain a validity time, suggested by the NF Service Consumer, representing the time span during which the subscription is desired to be kept active.

2a. On success, "201 Created" shall be returned. The response shall contain the validity time, as determined by the NRF, after which the subscription becomes invalid. Once the subscription expires, if the NF Service Consumer wants to keep receiving status notifications, it shall create a new subscription in the NRF.

2b. If the creation of the subscription fails at the NRF due to errors in the SubscriptionData JSON object in the request body, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the creation of the subscription fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

##### 5.2.2.5.3 Subscription to NF Instances in a different PLMN

The subscription to notifications on NF Instances in a different PLMN is done by creating a resource under the collection resource "subscriptions", in the NRF of the Home PLMN.

For that, step 1 in subclause 5.2.2.5.2 is executed (send a POST request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF in the SubscriptionData parameter in the request body.

Then, steps 1-2 in Figure 5.2.2.5.3-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the presence of the PLMN ID in the SubscriptionData parameter is not required. The NRF in the Home PLMN returns a subscriptionID identifying the created subscription.

Finally, step 2 in subclause 5.2.2.5.2 is executed; a new subscriptionID shall be generated by the NRF in the Serving PLMN as indicated in step 2 of Figure 5.2.2.5.3-1, and shall be sent to the NF Service Consumer in the Serving PLMN.



Figure 5.2.2.5.3-1: Subscription to NF Instances in a different PLMN

1. The NRF in Serving PLMN shall send a POST request to the resource URI in the NRF in Home PLMN representing the "subscriptions" collection resource. The request body shall include the SubscriptionData as received by the NRF in Serving PLMN from the NF Service Consumer in the Serving PLMN (see 5.2.2.5.2), containing the data about the type of notifications that the NF Service Consumer is interested in receiving and the callback URI where the NF Service Consumer shall be prepared to receive the notifications from the NRF (see NFStatusNotify operation in 5.2.2.6).

2a. On success, "201 Created" shall be returned. The NRF in Serving PLMN should not keep state for this created subscription and shall send to the NF Service Consumer in Serving PLMN (step 2 in 5.2.2.5.2) a subscriptionID that shall consist on the following structure: <MCC>+<MNC>+"-"+<OriginalSubscriptionID>

EXAMPLE: If the NRF in a Home PLMN (where MCC = 123, and MNC=456) creates a subscription with value "subs987654", the subscriptionID that the NRF in Serving PLMN would send to the NF Service Consumer in Serving PLMN is: "123456-subs987654"

2b. If the creation of the subscription fails at the NRF due to errors in the SubscriptionData JSON object in the request body, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the creation of the subscription fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

#### 5.2.2.6 NFStatusNotify

##### 5.2.2.6.1 General

This service operation notifies each NF Service Consumer that was previously subscribed to receiving notifications of registration/deregistration of NF Instances, or notifications of changes of the NF profile of a given NF Instance. The notification is sent to a callback URI that each NF Service Consumer provided during the subscription (see NFStatusSubscribe operation in 5.2.2.5).

##### 5.2.2.6.2 Notification from NRF in the same PLMN

The operation is invoked by issuing a POST request to each callback URI of the different subscribed NF Instances.



Figure 5.2.2.6.2-1: Notification from NRF in the same PLMN

1. The NRF shall send a POST request to the callback URI.

For notifications of newly registered NF Instances, the request body shall include the data associated to the newly registered NF, and its services, according to the criteria indicated by the NF Service Consumer during the subscription operation. These data shall contain the NFInstanceID of the NF Instance, an indication of the event being notified ("registration"), and the new profile data (including, among others, the services offered by the NF Instance).

For notifications of changes of the profile of a NF Instance, the request body shall include the NFInstancceID of the NF Instance whose profile was changed, an indication of the event being notified ("profile change"), and the new profile data.

For notifications of deregistration of the NF Instance from NRF, the request body shall include the NFInstanceID of the deregistered NF Instance, and an indication of the event being notified ("deregistration").

2a. On success, "204 No content" shall be returned by the NF Service Consumer.

2b. If the NF Service Consumer does not consider the "nfStatusNotificationURI" as a valid notification URI (e.g., because the URI does not belong to any of the existing subscriptions created by the NF Service Consumer in the NRF), the NF Service Consumer shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

##### 5.2.2.6.3 Notification from NRF in a different PLMN

The operation is invoked by issuing a POST request to each callback URI of the different subscribed NF Instances.



Figure 5.2.2.6.3-1: Notification from NRF in a different PLMN

Steps 1 and 2 are identical to steps 1 and 2 in Figure 5.2.2.6.2-1.

It should be noted that the POST request shall be sent directly from the NRF in Home PLMN to the NF Service Consumer in Serving PLMN, without involvement of the NRF in Serving PLMN.

#### 5.2.2.7 NFStatusUnSubscribe

##### 5.2.2.7.1 General

This service operation removes an existing subscription to notifications.

##### 5.2.2.7.2 Subscription removal in the same PLMN

It is executed by deleting a given resource identified by a "subscriptionID". The operation is invoked by issuing a DELETE request on the URI representing the specific subscription.



Figure 5.2.2.7.2-1: Subscription removal in the same PLMN

1. The NF Service Consumer shall send a DELETE request to the resource URI representing the individual subscription. The request body shall be empty.

2a. On success, "204 No Content" shall be returned. The response body shall be empty.

2b. If the subscription, identified by the "subscriptionID", is not found in the list of active subscriptions in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

##### 5.2.2.7.3 Subscription removal in a different PLMN

The subscription removal in a different PLMN is done by deleting a resource identified by a "subscriptionID", in the NRF of the Home PLMN.

For that, step 1 in subclause 5.2.2.7.2 is executed (send a DELETE request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF (MCC/MNC values) as a leading prefix of the susbcriptionID (see subclause 5.2.2.5.3).

Then, steps 1-2 in Figure 5.2.2.7.3-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the subscriptionID sent to the NRF in the Home PLMN shall not contain the identity of the PLMN (i.e., it shall be the same subscriptionID value as originally generated by the NRF in the Home PLMN). The NRF in the Home PLMN returns a status code with the result of the operation.

Finally, step 2 in subclause 5.2.2.7.2 is executed; a status code is returned to the NF Service Consumer in Serving PLMN in accordance to the result received from NRF in Home PLMN.



Figure 5.2.2.7.3-1: Subscription removal in a different PLMN

1. The NF Service Consumer shall send a DELETE request to the resource URI representing the individual subscription. The request body shall be empty.

2a. On success, "204 No Content" shall be returned. The response body shall be empty.

2b. If the subscription, identified by the "subscriptionID", is not found in the list of active subscriptions in the NRF's database, the NRF shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

## 5.3 Nnrf\_NFDiscovery Service

### 5.3.1 Service Description

The Nnrf\_NFDiscovery service allows a Network Function Instance to discover services offered by other Network Function Instances, by querying the local NRF.

It also allows an NRF in a PLMN to re-issue a discovery request towards an NRF in another PLMN (e.g., the HPLMN of a certain UE).

### 5.3.2 Service Operations

#### 5.3.2.1 Introduction

The service operations defined for the Nnrf\_NFDiscovery service are as follows:

- NFDiscover: It provides to the NF service consumer the IP address(es) or FQDN of the NF Instance(s) or NF Service(s) matching certain input criteria.

The NFDiscover operation can be invoked by an NF Service Consumer (i.e., "source NF") requesting to discover NF instances (i.e., "target NFs") located in the same PLMN, or in a different PLMN.

In the description of these operations in subclause 5.3.2.2, when the NF instances are located in the same PLMN, both source NF and target NFs are said to be located in the "Serving PLMN" but, in the general case, the functionality is not restricted to the PLMN that is serving a given UE, and it shall be applicable as well to any scenario in which source NF and target NFs belong to the same PLMN.

When source NF and target NFs are located in different PLMNs, the source NF is said to be in the "Serving PLMN", and the target NFs (and the NRF where they are registered) are said to be in the "Home PLMN", similarly to the scenarios described in 3GPP TS 23.502 [2], but the functionality shall be equally applicable to any scenario between any pair of PLMNs (e.g. with the source NF in the Home PLMN and the target NF in the Serving PLMN).

#### 5.3.2.2 NFDiscover

##### 5.3.2.2.1 General

This service operation discovers the set of NF Instances (and their associated NF Service Instances), represented by their NF Profile, that are currently registered in NRF and satisfy a number of input query parameters.

##### 5.3.2.2.2 Service Discovery in the same PLMN

This service operation is executed by querying the "nf-instances" resource. The request is sent to an NRF in the same PLMN of the NF Service Consumer.



Figure 5.3.2.2.2-1: Service Discovery Request in the same PLMN

1. The NF Service Consumer shall send an HTTP GET request to the resource URI "nf-instances" collection resource. The input filter criteria for the discovery request shall be included in query parameters.

2a. On success, "200 OK" shall be returned. The response body shall contain a validity period, during which the search result can be cached by the NF Service Consumer, and an array of NF profile objects, that satisfy the search filter criteria (e.g., all NF Instances offering a certain NF Service name).

2b. If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.

If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

The NF profile objects returned in a successful result shall contain generic data of each NF instance, applicable to any NF type, and it may also contain NF-specific data, for those NF instances belonging to a specific type (e.g., the attribute "udrInfo" is typically present in the NF profile when the type of the NF instance takes the value "UDR"). In addition, the attribute "customInfo", may be present in the NF profile for those NF instances with custom NF types.

For those NF instances, the "customInfo" attribute shall be returned by NRF, if available, as part of the NF profiles returned in the discovery response.

##### 5.3.2.2.3 Service Discovery in a different PLMN

The service discovery in a different PLMN is done by querying the "nf-instances" resource in the NRF of the Home PLMN.

For that, step 1 in subclause 5.3.2.2.2 is executed (send a GET request to the NRF in the Serving PLMN); this request shall include the identity of the PLMN of the home NRF in a query parameter of the URI.

Then, steps 1-2 in Figure 5.3.2.2.3-1 are executed, between the NRF in the Serving PLMN and the NRF in the Home PLMN. In this step, the presence of the PLMN ID of the Home NRF in the query parameter of the URI is not required. The NRF in the Home PLMN returns a status code with the result of the operation.

Finally, step 2 in subclause 5.3.2.2.2 is executed; a status code is returned to the NF Service Consumer in Serving PLMN in accordance to the result received from NRF in Home PLMN.



Figure 5.3.2.2.3-1: Service Discovery in a different PLMN

Steps 1 and 2 are similar to steps 1 and 2 in Figure 5.3.2.2.2-1, by replacing the originator of the service invocation with the NRF in Serving PLMN, and the recipient of the service invocation with the NRF in the Home PLMN.

##### 5.3.2.2.4 Service Discovery with intermediate redirecting NRF

When multiple NRFs are deployed in one PLMN, one NRF may query the "nf-instances" resource in a different NRF so as to fulfil the service discovery request from a NF service consumer. The query between these two NRFs is redirected by a third NRF.



Figure 5.3.2.2.4-1: Service Discovery with intermediate redirecting NRF

1. NRF-1 receives a service discovery request but does not have the information to fulfil the request. Then NRF-1 sends the service discovery request to a pre-configured NRF-2.

2a. Upon receiving a service discovery request, based on the information contained in the service discovery request (e.g. the "supi" query parameter in the URI) and locally stored information NRF-2 shall identify the next hop NRF (see subclause 5.2.2.2.3), and redirect the service discovery request by returning HTTP 307 Temporary Redirect response. The locally stored information in NRF-2 may:

a) be preconfigured; or

b) registered by other NRFs (see subclause 5.2.2.2.3).

The 307 Temporary Redirect response shall contain a Location header field, the host part of the URI in the Location header field represents NRF-3.

2b. if NRF-2 does not have enough information to redirect the service discovery request, then it responds with 404 Not Found, and the rest of the steps are omitted.

3. Upon receiving 307 Temporary Redirect response, NRF-1 sends the service discovery request to NRF-3 by using the URI contained in the Location header field of the 307 Temporary Redirect response.

4a. Upon success, NRF-3 returns the search result.

4b. If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.

If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

##### 5.3.2.2.5 Service Discovery with intermediate forwarding NRF

When multiple NRFs are deployed in one PLMN, one NRF may query the "nf-instances" resource in a different NRF so as to fulfil the service discovery request from a NF service consumer. The query between these two NRFs is forwarded by a third NRF.



Figure 5.3.2.2.5-1: Service Discovery with intermediate forwarding NRF

1. NRF-1 receives a service discovery request but does not have the information to fulfil the request. Then NRF-1 sends the service discovery request to a pre-configured NRF-2.

2a. Upon receiving a service discovery request, based on the information contained in the service discovery request (e.g. the "supi" query parameter in the URI) and locally stored information, NRF-2 shall identify the next hop NRF (see subclause 5.2.2.2.3), and forward the service discovery request to that NRF (i.e. NRF-3 in this example), by replacing the originator of the service invocation with NRF-2, and the recipient of the service invocation with NRF-3. The locally stored information in NRF-2 may:

a) be preconfigured; or

b) registered by other NRFs (see subclause 5.2.2.2.3).

2b. if NRF-2 does not have enough information to forward the service discovery request, then it responds with 404 Not Found, and the rest of the steps are omitted.

3a. Upon success, NRF-3 returns the search result.

3b. If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.

If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF shall return "400 Bad Request" status code with the ProblemDetails IE providing details of the error.

If the discovery request fails at the NRF due to NRF internal errors, the NRF shall return "500 Internal Server Error" status code with the ProblemDetails IE providing details of the error.

4a. NRF-2 forwards the success response to NRF-1.

4b. NRF-2 forwards the error response to NRF-1.

NOTE: It is not assumed that there can only be two NRF hierarchies, i.e. the NRF-3 can go on to forward the service discovery request to another NRF.

## 5.4 OAuth2 Authorization Service

### 5.4.1 Service Description

The NRF offers an OAuth2 authorization service (see IETF RFC 6749 [16]), following the "Client Credentials" authorization grant, as specified in 3GPP TS 33.501 [15]. It exposes a "Token Endpoint" where the Access Token Request service can be requested by NF Service Consumers.

### 5.4.2 Service Operations

#### 5.4.2.1 Introduction

The services operations defined for the OAuth2 Authorization service are as follows:

- Access Token Request

#### 5.4.2.2 Access Token Request

##### 5.4.2.2.1 General

This service operation is used by an NF Service Consumer to request an OAuth2 access token from the authorization server (NRF).



Figure 5.4.2.2.1-1: Access Token Request

1. The NF Service Consumer shall send a POST request to the "Token Endpoint", as described in IETF RFC 6749 [16], clause 3.2. The "Token Endpoint" URI shall be:

{nrfApiRoot}/oauth2/token

where {nrfApiRoot} represents the concatenation of the "scheme" and "authority" components of the NRF, as defined in IETF RFC 3986 [17].  
  
The body of the HTTP POST request shall indicate that the required OAuth2 grant must be of type "Client Credentials". The "scope" parameter shall include the name of the NF Service that the NF Service Consumer is trying to access (i.e., the expected NF service name) and the following information shall be carried as additional parameters:

- NF Instance Id of the the NF Service Consumer requesting the OAuth2.0 access token;

- The NF type of the NF Service Producer; and

- NF Service Consumer type

Editor's Note: Whether NF instance Id of the NF service producer needs to be included in the request is FFS.  
  
The NF Service Consumer shall use HTTP Basic authentication towards this endpoint, using the "nfInstanceId" of the NF Service Consumer as "username", and using the registered credential between NF Service Consumer and NRF, as "password".

2. On success, "200 OK" shall be returned, the payload body of the POST response shall contain, among other parameters, the requested access token, the token type and the expiration time for the token. The access token shall be a JSON Web Token (JWT) as specified in IETF RFC 7519 [25]. The access token returned by the NRF shall include the claims encoded as a JSON object as specified in subclause 6.3.5.2.4 and then digitally signed using JWS as specified in IETF RFC 7515 [24] and in subclause 13.4.1 of 3GPP TS 33.501 [15].

The digitally signed access token shall be converted to the JWS Compact Serialization encoding as a string as specified in clause 7.1 of IETF RFC 7515 [24].

If the access token request fails at the NRF, the NRF shall return "400 Bad Request" status code, including a JSON object in the response payload, that includes details about the specific error that occurred.

# 6 API Definitions

## 6.1 Nnrf\_NFManagement Service API

### 6.1.1 API URI

URIs of this API shall have the following root:

{apiRoot}/{apiName}/{apiVersion}/

where the "apiName" shall be set to "nnrf-nfm" and the "apiVersion" shall be set to "v1" for the current version of this specification.

### 6.1.2 Usage of HTTP

#### 6.1.2.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in subclause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies for the Nnrf\_NFManagement service shall comply with the OpenAPI [10] specification contained in Annex A.

#### 6.1.2.2 HTTP Standard Headers

##### 6.1.2.2.1 General

##### 6.1.2.2.2 Content type

The following content types shall be supported:

- JSON, as defined in IETF RFC 8259 [22], shall be used as content type of the HTTP bodies specified in the present specification as indicated in subclause 5.4 of 3GPP TS 29.500 [4].

- The Problem Details JSON Object (IETF RFC 7807 [11]). The use of the Problem Details JSON object in a HTTP response body shall be signalled by the content type "application/problem+json".

- JSON Patch (IETF RFC 6902 [13]). The use of the JSON Patch format in a HTTP request body shall be signalled by the content type "application/json-patch+json".

- The 3GPP hypermedia format as defined in 3GPP TS 29.501 [5]. The use of the 3GPP hypermedia format in a HTTP response body shall be signalled by the content type "application/3gppHal+json".

#### 6.1.2.3 HTTP custom headers

##### 6.1.2.3.1 General

In this release of this specification, no custom headers specific to the Nnrf\_NFManagement service are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see subclause 5.2.3 of 3GPP TS 29.500 [4].

### 6.1.3 Resources

#### 6.1.3.1 Overview

The structure of the Resource URIs of the NFManagement service is shown in figure 6.1.3.1-1.



Figure 6.1.3.1-1: Resource URI structure of the NFManagement API

Table 6.1.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.1.3.1-1: Resources and methods overview

|  |  |  |  |
| --- | --- | --- | --- |
| Resource name | Resource URI | HTTP method or custom operation | Description |
| nf-instances  (Store) | {apiRoot}/nnrf-nfm/v1/nf-instances | GET | Read a collection of NF Instances. |
| nf-instance  (Document) | {apiRoot}/nnrf-nfm/v1/nf-instances/{nfInstanceID} | GET | Read the profile of a given NF Instance. |
| PUT | Register in NRF a new NF Instance, or replace the profile of an existing NF Instance, by providing an NF profile. |
| PATCH | Modify the NF profile of an existing NF Instance. |
| DELETE | Deregister from NRF a given NF Instance. |
| subscriptions  (Collection) | {apiRoot}/nnrf-nfm/v1/subscriptions | POST | Creates a new subscription in NRF to newly registered NF Instances. |
| subscription  (Document) | {apiRoot}/nnrf-nfm/v1/subscriptions/{subscriptionID} | DELETE | Deletes an existing subscription from NRF. |
| Notification Callback | {nfStatusNotificationUri} | POST | Notify about newly created NF Instances, or about changes of the profile of a given NF Instance. |

#### 6.1.3.2 Resource: nf-instances (Store)

##### 6.1.3.2.1 Description

This resource represents a collection of the different NF instances registered in the NRF.

This resource is modelled as the Store resource archetype (see subclause C.3 of 3GPP TS 29.501 [5]).

##### 6.1.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/nf-instances**

This resource shall support the resource URI variables defined in table 6.1.3.2.2-1.

Table 6.1.3.2.2-1: Resource URI variables for this resource

|  |  |
| --- | --- |
| Name | Definition |
| apiRoot | See subclause 6.1.1 |

##### 6.1.3.2.3 Resource Standard Methods

###### 6.1.3.2.3.1 GET

This method retrieves a list of all NF instances currently registered in the NRF. This method shall support the URI query parameters specified in table 6.1.3.2.3.1-1.

Table 6.1.3.2.3.1-1: URI query parameters supported by the GET method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| nf-type | NFType | O | 0..1 | The type of NF to restrict the list of returned NF Instances. |
| limit | integer | O | 0..1 | Maximum number of items to be returned in this query. |

This method shall support the request data structures specified in table 6.1.3.2.3.1-2 and the response data structures and response codes specified in table 6.1.3.2.3.1-3.

Table 6.1.3.2.3.1-2: Data structures supported by the GET Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 6.1.3.2.3.1-3: Data structures supported by the GET Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| UriList | M | 1 | 200 OK | The response body contains a "\_links" object containing the URI of each registered NF in the NRF, or an empty object if there are no NFs to return in the query result (e.g., because there are no registered NFs in the NRF, or because there are no matching NFs of the type specified in the "nf-type" query parameter, currently registered in the NRF). |

##### 6.1.3.2.4 Resource Custom Operations

There are no resource custom operations for the Nnrf\_NFManagement service in this release of the specification.

#### 6.1.3.3 Resource: nf-instance (Document)

##### 6.1.3.3.1 Description

This resource represents a single NF instance.

##### 6.1.3.3.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/nf-instances/{nfInstanceID}**

This resource shall support the resource URI variables defined in table 6.1.3.3.2-1.

Table 6.1.3.3.2-1: Resource URI variables for this resource

|  |  |
| --- | --- |
| Name | Definition |
| apiRoot | See subclause 6.1.1 |
| nfInstanceID | Represents a specific NF Instance |

##### 6.1.3.3.3 Resource Standard Methods

###### 6.1.3.3.3.1 GET

This method retrieves the NF Profile of a given NF instance.

This method shall support the URI query parameters specified in table 6.1.3.3.3.1-1.

Table 6.1.3.3.3.1-1: URI query parameters supported by the GET method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.3.3.1-2 and the response data structures and response codes specified in table 6.1.3.3.3.1-3.

Table 6.1.3.3.3.1-2: Data structures supported by the GET Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 6.1.3.3.3.1-3: Data structures supported by the GET Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| NFProfile | M | 1 | 200 OK | The response body contains the profile of a given NF Instance. |
| ProblemDetails | M | 1 | 403 Forbidden | The NF Service Consumer is not allowed to query the instances of this NF type. |

###### 6.1.3.3.3.2 PUT

This method registers a new NF instance in the NRF, or replaces completely an existing NF instance.

This method shall support the URI query parameters specified in table 6.1.3.3.3.2-1.

Table 6.1.3.3.3.2-1: URI query parameters supported by the PUT method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.3.3.2-2 and the response data structures and response codes specified in table 6.1.3.3.3.2-3.

Table 6.1.3.3.3.2-2: Data structures supported by the PUT Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| NFProfile | M | 1 | Profile of the NF Instance to be registered, or completely replaced, in NRF. |

Table 6.1.3.3.3.2-3: Data structures supported by the PUT Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| NFProfile | M | 1 | 200 OK | This case represents the successful replacement of an existing NF Instance profile.  Upon success, a response body is returned containing the replaced profile of the NF Instance. |
| NFRegistrationData | M | 1 | 201 Created | This case represents the successful registration of a new NF Instance.  Upon success, a response body is returned containing a heart-beat timer and the newly created NF Instance profile; also, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created NF Instance. |
| ProblemDetails | M | 1 | 400 Bad Request | This case represents the failure registration of a new NF Instance, because of input parameter error. |
| ProblemDetails | M | 1 | 500 Internal Server Error | This case represents the failure in the registration of a new NF Instance, because of a server internal error. |

###### 6.1.3.3.3.3 PATCH

This method updates partially the profile of a given NF instance.

This method shall support the URI query parameters specified in table 6.1.3.3.3.3-1.

Table 6.1.3.3.3.3-1: URI query parameters supported by the PATCH method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.3.3.3-2 and the response data structures and response codes specified in table 6.1.3.3.3.3-3.

Table 6.1.3.3.3.3-2: Data structures supported by the PATCH Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| PatchDocument | M | 1 | It contains the list of changes to be made to the profile of the NF Instance, according to the JSON PATCH format specified in IETF RFC 6902 [13]. |

Table 6.1.3.3.3.3-3: Data structures supported by the PATCH Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| NFProfile | M | 1 | 200 OK | Upon success, a response body is returned containing the updated profile of the NF Instance. |
| n/a |  |  | 204 No Content | Successful response sent when there is no need to provide a full updated profile of the NF Instance (e.g., in the Heart-Beat operation response described in subclause 5.2.2.3.2). |

###### 6.1.3.3.3.4 DELETE

This method deregisters an existing NF instance from the NRF.

This method shall support the URI query parameters specified in table 6.1.3.3.3.4-1.

Table 6.1.3.3.3.4-1: URI query parameters supported by the DELETE method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.3.3.4-2 and the response data structures and response codes specified in table 6.1.3.3.3.4-3.

Table 6.1.3.3.3.4-2: Data structures supported by the DELETE Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 6.1.3.3.3.4-3: Data structures supported by the DELETE Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| n/a |  |  | 204 No Content |  |

#### 6.1.3.4 Resource: subscriptions (Collection)

##### 6.1.3.4.1 Description

This resource represents a collection of subscriptions of NF Instances to newly registered NF Instances.

##### 6.1.3.4.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/subscriptions**

This resource shall support the resource URI variables defined in table 6.1.3.4.2-1.

Table 6.1.3.4.2-1: Resource URI variables for this resource

|  |  |
| --- | --- |
| Name | Definition |
| apiRoot | See subclause 6.1.1 |

##### 6.1.3.4.3 Resource Standard Methods

###### 6.1.3.4.3.1 POST

This method creates a new subscription. This method shall support the URI query parameters specified in table 6.1.3.4.3.1-1.

Table 6.1.3.4.3.1-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.4.3.1-2 and the response data structures and response codes specified in table 6.1.3.4.3.1-3.

Table 6.1.3.4.3.1-2: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| SubscriptionData | M | 1 | The request body contains the input parameters for the subscription. These parameters include, e.g.:  - Target NF type  - Target Service Name  - Callback URI of the Requester NF |

Table 6.1.3.4.3.1-3: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| SubscriptionData | M | 1 | 201 Created | This case represents the successful creation of a subscription.  Upon success, the HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource. |

#### 6.1.3.5 Resource: subscription (Document)

##### 6.1.3.5.1 Description

This resource represents an individual subscription of a given NF Instance to newly registered NF Instances.

##### 6.1.3.5.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-nfm/v1/subscriptions/{subscriptionID}**

This resource shall support the resource URI variables defined in table 6.1.3.5.2-1.

Table 6.1.3.5.2-1: Resource URI variables for this resource

|  |  |
| --- | --- |
| Name | Definition |
| apiRoot | See subclause 6.1.1 |
| subscriptionID | Represents a specific subscription |

##### 6.1.3.5.3 Resource Standard Methods

###### 6.1.3.5.3.1 DELETE

This method terminates an existing subscription. This method shall support the URI query parameters specified in table 6.1.3.5.3.1-1.

Table 6.1.3.5.3.1-1: URI query parameters supported by the DELETE method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.5.3.1-2 and the response data structures and response codes specified in table 6.1.3.5.3.1-3.

Table 6.1.3.5.3.1-2: Data structures supported by the DELETE Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 6.1.3.5.3.1-3: Data structures supported by the DELETE Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| n/a |  |  | 204 No Content |  |

### 6.1.4 Custom Operations without associated resources

There are no custom operations defined without any associated resources for the Nnrf\_NFManagement service in this release of the specification.

### 6.1.5 Notifications

#### 6.1.5.1 General

This subclause specifies the notifications provided by the Nnrf\_NFManagement service.

The delivery of notifications shall be supported as specified in subclause 6.2 of 3GPP TS 29.500 [4] for Server-initiated communication.

Table 6.1.5.1-1: Notifications overview

|  |  |  |  |
| --- | --- | --- | --- |
| Notification | Resource URI | HTTP method or custom operation | Description  (service operation) |
| NF Instance Status Notification | {nfStatusNotificationUri}  (NF Service Consumer provided callback reference) | POST | Notify about registrations / deregistrations or profile changes of NF Instances |

#### 6.1.5.2 NF Instance Status Notification

##### 6.1.5.2.1 Description

The NF Service Consumer provides a callback URI for getting notified about NF Instances status events, the NRF shall notify the NF Service Consumer, when the conditions specified in the subscription are met.

##### 6.1.5.2.2 Notification Definition

The POST method shall be used for NF Instance Status notification and the URI shall be the callback reference provided by the NF Service Consumer during the subscription to this notification.

Resource URI: **{nfStatusNotificationUri}**

Support of URI query parameters is specified in table 6.1.5.2.2-1.

Table 6.1.5.2.2-1: URI query parameters supported by the POST method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

Support of request data structures is specified in table 6.1.5.2.2-2, and support of response data structures and response codes is specified in table 6.1.5.2-3.

Table 6.1.5.2.2-2: Data structures supported by the POST Request Body

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| NotificationData | M | 1 | Representation of the NF Instance status notification. |

Table 6.1.5.2.2-3: Data structures supported by the POST Response Body

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| N/A |  |  | 204 No Content | This case represents a successful notification of the NF Instance status event. |

### 6.1.6 Data Model

#### 6.1.6.1 General

This subclause specifies the application data model supported by the API.

Table 6.1.6.1-1 specifies the data types defined for the Nnrf service based interface protocol.

Table 6.1.6.1-1: Nnrf\_NFManagement specific Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Section defined | Description |
| NFProfile | 6.1.6.2.2 |  |
| NFService | 6.1.6.2.3 |  |
| DefaultNotificationSubscription | 6.1.6.2.4 | Data structure for specifying the notifications the NF service subscribes by default along with callback URI. |
| IpEndPoint | 6.1.6.2.5 |  |
| UdrInfo | 6.1.6.2.6 |  |
| UdmInfo | 6.1.6.2.7 |  |
| AusfInfo | 6.1.6.2.8 |  |
| SupiRange | 6.1.6.2.9 |  |
| IdentityRange | 6.1.6.2.10 |  |
| AmfInfo | 6.1.6.2.11 |  |
| SmfInfo | 6.1.6.2.12 |  |
| UpfInfo | 6.1.6.2.13 | Information related to UPF |
| SnssaiUpfInfoItem | 6.1.6.2.14 |  |
| DnnUpfInfoItem | 6.1.6.2.15 |  |
| SubscriptionData | 6.1.6.2.16 |  |
| NotificationData | 6.1.6.2.17 |  |
| NFRegistrationData | 6.1.6.2.18 |  |
| NFServiceVersion | 6.1.6.2.19 | Contains the version details of an NF service. |
| PcfInfo | 6.1.6.2.20 |  |
| BsfInfo | 6.1.6.2.21 |  |
| Ipv4AddressRange | 6.1.6.2.22 |  |
| Ipv6PrefixRange | 6.1.6.2.23 |  |
| InterfaceUpfInfoItem | 6.1.6.2.24 |  |
| UriList | 6.1.6.2.25 |  |
| N2InterfaceAmfInfo | 6.1.6.2.26 | AMF N2 interface information |
| TaiRange | 6.1.6.2.27 |  |
| TacRange | 6.1.6.2.28 |  |
| Fqdn | 6.1.6.3.2 |  |
| NFType | 6.1.6.3.3 |  |
| NotificationType | 6.1.6.3.4 |  |
| TransportProtocol | 6.1.6.3.5 |  |
| NotificationEventType | 6.1.6.3.6 |  |
| NFStatus | 6.1.6.3.7 |  |
| DataSetId | 6.1.6.3.8 |  |
| UPInterfaceType | 6.1.6.3.9 |  |
| ServiceName | 6.1.6.3.11 |  |
| NFServiceStatus | 6.1.6.3.12 |  |

Table 6.1.6.1-2 specifies data types re-used by the Nnrf service based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nnrf service based interface.

Table 6.1.6.1-2: Nnrf\_NFManagement re-used Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Reference | Comments |
| N1MessageClass | 3GPP TS 29.518 [6] | The N1 message type |
| N2InformationClass | 3GPP TS 29.518 [6] | The N2 information type |
| IPv4Addr | 3GPP TS 29.571 [7] |  |
| IPv6Addr | 3GPP TS 29.571 [7] |  |
| IPv6Prefix | 3GPP TS 29.571 [7] |  |
| Uri | 3GPP TS 29.571 [7] |  |
| Dnn | 3GPP TS 29.571 [7] |  |
| SupportedFeatures | 3GPP TS 29.571 [7] |  |
| Snssai | 3GPP TS 29.571 [7] |  |
| PlmnId | 3GPP TS 29.571 [7] |  |
| Guami | 3GPP TS 29.571 [7] |  |
| Tai | 3GPP TS 29.571 [7] |  |
| NfInstanceId | 3GPP TS 29.571 [7] |  |
| LinksValueSchema | 3GPP TS 29.571 [7] | 3GPP Hypermedia link |
| UriScheme | 3GPP TS 29.571 [7] |  |
| AmfName | 3GPP TS 29.571 [7] |  |
| DateTime | 3GPP TS 29.571 [7] |  |

#### 6.1.6.2 Structured data types

##### 6.1.6.2.1 Introduction

This subclause defines the structures to be used in resource representations.

##### 6.1.6.2.2 Type: NFProfile

Table 6.1.6.2.2-1: Definition of type NFProfile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| nfInstanceID | NfInstanceId | M | 1 | Unique identity of the NF Instance. |
| nfType | NFType | M | 1 | Type of Network Function |
| nfStatus | NFStatus | M | 1 | Status of the NF Instance (NOTE 5) |
| plmn | PlmnId | O | 0..1 | PLMN of the Network Function |
| sNssais | array(Snssai) | O | 0..N | S-NSSAIs of the Network Function |
| nsiList | array(string) | O | 0..N | NSI identities of the Network Function |
| fqdn | Fqdn | C | 0..1 | FQDN of the Network Function (NOTE 1, NOTE 2). For AMF the FQDN registered with the NRF shall be that of the AMF Name (see 3GPP TS 23.003 [12] subclause 28.3.2.5). |
| interPlmnFqdn | Fqdn | C | 0..1 | If the NF 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 TS 23.003 [12] shall be registered with the NRF. |
| ipv4Addresses | array(Ipv4Addr) | C | 0..N | IPv4 address(es) of the Network Function (NOTE 1, NOTE 2) |
| ipv6Addresses | array(Ipv6Addr) | C | 0..N | IPv6 address(es) of the Network Function (NOTE 1, NOTE 2) |
| priority | integer | O | 0..1 | 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. (See NOTE 4).  The NRF may overwrite the received priority value when exposing an NFProfile with the Nnrf\_NFDiscovery service. |
| capacity | integer | O | 0..1 | 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. (See NOTE 4). |
| load | integer | O | 0..1 | Dynamic load information, ranged from 0 to 100, indicates the current load percentage of the NF. |
| locality | string | O | 0..1 | Operator defined information about the location of the NF instance (e.g. geographic location, data center) (NOTE 3) |
| udrInfo | UdrInfo | O | 0..1 | Specific data for the UDR (ranges of SUPI, group ID …) |
| udmInfo | UdmInfo | O | 0..1 | Specific data for the UDM (ranges of SUPI, group ID…) |
| ausfInfo | AusfInfo | O | 0..1 | Specific data for the AUSF (ranges of SUPI, group ID…) |
| amfInfo | AmfInfo | O | 0..1 | Specific data for the AMF (AMF Set ID, …) |
| smfInfo | smfInfo | O | 0..1 | Specific data for the SMF (DNN's, …) |
| upfInfo | UpfInfo | O | 0..1 | Specific data for the UPF (S-NSSAI, DNN, SMF serving area, interface…) |
| pcfInfo | PcfInfo | O | 0..1 | Specific data for the PCF |
| bsfInfo | BsfInfo | O | 0..1 | Specific data for the BSF |
| customInfo | object | O | 0..1 | Specific data for custom Network Functions |
| recoveryTime | DateTime | O | 0..1 | Timestamp when the NF was (re)started (NOTE 5, NOTE 6) |
| nfServices | array(NFService) | O | 0..N | List of NF Service Instances |
| NOTE 1: At least one of the addressing parameters (fqdn, ipv4address or ipv6adress) shall be included in the NF Profile.  NOTE 2: If the type of Network Function is UPF, the addressing information is for the UPF N4 interface.  NOTE 3: A requester NF may use this information to select a NF instance (e.g. a NF instance preferably located in the same data center).  NOTE 4: The capacity and priority parameters, if present, are used for NF selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].  NOTE 5: The NRF shall notify NFs subscribed to receiving notifications of changes of the NF profile, if the NF recoveryTime or the nfStatus is changed. See subclause 6.2 of 3GPP TS 23.527 [27].  NOTE 6: A requester NF may consider that all the resources created in the NF before the NF recovery time have been lost. This may be used to detect a restart of a NF and to trigger appropriate actions, e.g. release local resources. See subclause 6.2 of 3GPP TS 23.527 [27]. | | | | |

##### 6.1.6.2.3 Type: NFService

Table 6.1.6.2.3-1: Definition of type NFService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| serviceInstanceID | string | M | 1 | Unique ID of the service instance within a given NF Instance |
| serviceName | ServiceName | M | 1 | Name of the service instance (e.g. "nudm-sdm") |
| versions | array(NFServiceVersion) | M | 1..N | The API versions supported by the NF Service and if available, the corresponding retirement date of the NF Service.  The different array elements shall have distinct unique values for "apiVersionInUri", and consequently, the values of "apiFullVersion" shall have a unique first digit version number. |
| scheme | UriScheme | M | 1 | URI scheme (e.g. "http", "https") |
| nfServiceStatus | NFServiceStatus | M | 0..1 | Status of the NF Service Instance (see NOTE 3) |
| fqdn | Fqdn | O | 0..1 | FQDN of the NF where the service is hosted (see NOTE 1) |
| interPlmnFqdn | Fqdn | O | 0..1 | 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 TS 23.003 [12] may be registered with the NRF (see NOTE 1). |
| ipEndPoints | array(IpEndPoint) | O | 0..N | IP address(es) and port information of the Network Function (including IPv4 and/or IPv6 address) where the service is listening for incoming service requests (see NOTE 1) |
| apiPrefix | string | O | 0..1 | Optional path segment(s) used to construct the {apiRoot} variable of the different API URIs, as described in 3GPP TS 29.501 [5], subclause 4.4.1 |
| defaultNotificationSubscriptions | array(DefaultNotificationSubscription) | O | 0..N | Notification endpoints for different notification types. |
| allowedPlmns | array(PlmnId) | O | 0..N | PLMNs allowed to access the service instance |
| allowedNfTypes | array(NFType) | O | 0..N | Type of the NFs allowed to access the service instance |
| allowedNfDomains | array(string) | O | 0..N | Pattern (regular expression according to the ECMA-262 dialect [8]) representing the NF domain names allowed to access the service instance. |
| allowedNssais | array(Snssai) | O | 0..N | S-NSSAI of the allowed slices to access the service instance |
| priority | integer | O | 0..1 | 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. (See NOTE 2).  The NRF may overwrite the received priority value when exposing an NFProfile with the Nnrf\_NFDiscovery service. |
| capacity | integer | O | 0..1 | Static capacity information in the range of 0-65535, expressed as a weight relative to other services of the same type. (See NOTE 2). |
| load | integer | O | 0..1 | Dynamic load information, ranged from 0 to 100, indicates the current load percentage of the NF Service. |
| recoveryTime | DateTime | O | 0..1 | Timestamp when the NF service was (re)started (NOTE 3, NOTE 4) |
| supportedFeatures | SupportedFeatures | O | 0..1 | Supported Features of the NF Service instance |
| NOTE 1: If the fqdn, interPlmnFqdn and ipEndpoint attributes are not present, the FQDN and IP address related attributes from the NF Profile shall be used to construct the API URIs of this service.  NOTE 2: The capacity and priority parameters, if present, are used for NF selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].  NOTE 3: The NRF shall notify NFs subscribed to receiving notifications of changes of the NF profile, if the recoveryTime or the nfServiceStatus is changed. See subclause 6.2 of 3GPP TS 23.527 [27].  NOTE 4: A requester NF subscribed to NF status changes may consider that all the resources created in the NF service before the NF service recovery time have been lost. This may be used to detect a restart of a NF service and to trigger appropriate actions, e.g. release local resources. See subclause 6.2 of 3GPP TS 23.527 [27]. | | | | |

##### 6.1.6.2.4 Type: DefaultNotificationSubscription

Table 6.1.6.2.4-1: Definition of type DefaultNotificationSubscription

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| notificationType | NotificationType | M | 1 | Type of notification for which the corresponding callback URI is provided. |
| callbackUri | Uri | M | 1 | The callback URI. |
| n1MessageClass | N1MessageClass | C | 0..1 | If the notification type is N1\_MESSAGES, this IE shall be present and shall identify the class of N1 messages to be notified. |
| n2InformationClass | N2InformationClass | C | 0..1 | If the notification type is N2\_INFORMATION, this IE shall be present and shall identify the class of N2 information to be notified. |

##### 6.1.6.2.5 Type: IpEndPoint

Table 6.1.6.2.5-1: Definition of type IpEndPoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| ipv4Address | Ipv4Addr | C | 0..1 | IPv4 address (NOTE 1) |
| ipv6Address | Ipv6Addr | C | 0..1 | IPv6 address (NOTE 1) |
| transport | TransportProtocol | O | 0..1 | Transport protocol |
| port | integer | O | 0..1 | Port number (NOTE 2) |
| NOTE 1: At most one occurrence of either ipv4Address or ipv6Address shall be included in this data structure.  NOTE 2: If the port number is absent from the ipEndPoints attribute, the NF service consumer shall use the default HTTP port number, i.e. TCP port 80 for "http" URIs or TCP port 443 for "https" URIs as specified in IETF RFC 7540 [9] when invoking the service. | | | | |

##### 6.1.6.2.6 Type: UdrInfo

Table 6.1.6.2.6-1: Definition of type UdrInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| groupId | string | O | 0..1 | Identity of the UDR group that is served by the UDR instance |
| supiRanges | array(SupiRange) | O | 0..N | List of ranges of SUPI's whose profile data is available in the UDR instance (NOTE 1) |
| gpsiRanges | array(IdentityRange) | O | 0..N | List of ranges of GPSIs whose profile data is available in the UDR instance (NOTE 1) |
| externalGroupIdentifiersRanges | array(IdentityRange) | O | 0..N | List of ranges of external groups whose profile data is available in the UDR instance (NOTE 1) |
| supportedDataSets | array(DataSetId) | O | 0..N | List of supported data sets in the UDR instance |
| NOTE 1: If none of these parameters is provided, the UDR can serve any external group and any SUPI or GPSI. | | | | |

##### 6.1.6.2.7 Type: UdmInfo

Table 6.1.6.2.7-1: Definition of type UdmInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| groupId | String | O | 0..1 | Identity of the UDM group that is served by the UDM instance |
| supiRanges | array(SupiRange) | O | 0..N | List of ranges of SUPIs whose profile data is available in the UDM instance (NOTE 1) |
| gpsiRanges | array(IdentityRange) | O | 0..N | List of ranges of GPSIs whose profile data is available in the UDM instance (NOTE 1) |
| externalGroupIdentifiersRanges | array(IdentityRange) | O | 0..N | List of ranges of external groups whose profile data is available in the UDM instance (NOTE 1) |
| routingIndicators | array(string) | O | 0..N | List of Routing Indicator information that allows to route network signalling with SUCI (see 3GPP TS 23.003 [12]) to the UDM instance. |
| NOTE 1: If none of these parameters is provided, the UDM can serve any external group and any SUPI or GPSI. | | | | |

##### 6.1.6.2.8 Type: AusfInfo

Table 6.1.6.2.8-1: Definition of type AusfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| groupId | string | O | 0..1 | Identity of the AUSF group |
| supiRanges | array(SupiRange) | O | 0..N | List of ranges of SUPIs that can be served by the AUSF instance. If not provided, the AUSF can serve any SUPI. |
| routingIndicators | array(string) | O | 0..N | List of Routing Indicator information that allows to route network signalling with SUCI (see 3GPP TS 23.003 [12]) to the AUSF instance. |

##### 6.1.6.2.9 Type: SupiRange

Table 6.1.6.2.9-1: Definition of type SupiRange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| start | string | O | 0..1 | First value identifying the start of a SUPI range, to be used when the range of SUPI's can be represented as a numeric range (e.g., IMSI ranges). This string shall consist only of digits.  Pattern: "^\d+$" |
| end | string | O | 0..1 | Last value identifying the end of a SUPI range, to be used when the range of SUPI's can be represented as a numeric range (e.g. IMSI ranges). This string shall consist only of digits.  Pattern: "^\d+$" |
| pattern | string | O | 0..1 | Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of SUPI's belonging to this range. A SUPI value is considered part of the range if and only if the SUPI string fully matches the regular expression. |
| NOTE: Either the start and end attributes, or the pattern attribute, shall be present. | | | | |

EXAMPLE 1: IMSI range. From: 123 45 6789040000 To: 123 45 6789059999 (i.e., 20,000 IMSI numbers)  
JSON: { "start": "123456789040000", "end": "123456789059999" }

EXAMPLE 2: IMSI range. From: 123 45 6789040000 To: 123 45 6789049999 (i.e., 10,000 IMSI numbers)  
JSON: { "pattern": "^imsi-12345678904\d{4}$" }, or  
JSON: { "start": "123456789040000", "end": "123456789049999" }

EXAMPLE 3: NAI range. "[smartmeter-*{factoryID}*@company.com](mailto:smartmeter-%7bfactoryID%7d@company.com)" where "*{factoryID}*" can be any string.  
JSON: { "pattern": "[^nai-smartmeter-.+@company\.com$](mailto:%5enai-smartmeter-.+@gascompany\.com$)" }

##### 6.1.6.2.10 Type: IdentityRange

Table 6.1.6.2.10-1: Definition of type IdentityRange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| start | string | O | 0..1 | First value identifying the start of an identity range, to be used when the range of identities can be represented as a numeric range (e.g., MSISDN ranges). This string shall consist only of digits.  Pattern: "^\d+$" |
| end | string | O | 0..1 | Last value identifying the end of an identity range, to be used when the range of identities can be represented as a numeric range (e.g. MSISDN ranges). This string shall consist only of digits.  Pattern: "^\d+$" |
| pattern | String | O | 0..1 | Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of identities belonging to this range. An identity value is considered part of the range if and only if the identity string fully matches the regular expression. To be used when identity is External Identifier or External Group Identifier or MSISDN. |
| NOTE: Either the start and end attributes, or the pattern attribute, shall be present. | | | | |

##### 6.1.6.2.11 Type: AmfInfo

Table 6.1.6.2.11-1: Definition of type AmfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| amfRegionId | string | M | 1 | AMF region identifier |
| amfSetId | string | M | 1 | AMF set identifier. |
| guamiList | array(Guami) | M | 1..N | List of supported GUAMIs |
| taiList | array(Tai) | O | 0..N | The list of TAIs the AMF can serve. It may contain the non-3GPP access TAI. The absence of this attribute and the taiRangeList attribute indicate that the AMF can be selected for any TAI in the serving network. |
| taiRangeList | array(TaiRange) | O | 0..N | The range of TAIs the AMF can serve. The absence of this attribute and the taiList attribute indicate that the AMF can be selected for any TAI in the serving network. |
| backupInfoAmfFailure | array(Guami) | O | 0..N | List of GUAMIs for which the AMF acts as a backup for AMF failure |
| backupInfoAmfRemoval | array(Guami) | O | 0..N | List of GUAMIs for which the AMF acts as a backup for planned AMF removal |
| n2InterfaceAmfInfo | N2InterfaceAmfInfo | O | 0..1 | N2 interface information of the AMF. This information needs not be sent in NF Discovery responses. It may be used by the NRF to update the DNS for AMF discovery by the 5G Access Network. The procedures for updating the DNS are out of scope of this specification. |

##### 6.1.6.2.12 Type: SmfInfo

Table 6.1.6.2.12-1: Definition of type SmfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| dnnList | array(Dnn) | M | 1..N | DNNs supported by the SMF |
| taiList | array(Tai) | O | 0..N | The list of TAIs the SMF can serve. It may contain the non-3GPP access TAI. The absence of this attribute and the taiRangeList attribute indicate that the SMF can be selected for any TAI in the serving network. |
| taiRangeList | array(TaiRange) | O | 1..N | The range of TAIs the SMF can serve. It may contain the non-3GPP access' TAI. The absence of this attribute and the taiList attribute indicate that the SMF can be selected for any TAI in the serving network. |
| pgwFqdn | Fqdn | O | 0..1 | The FQDN of the PGW if the SMF is a combined SMF/PGW-C. |

##### 6.1.6.2.13 Type: UpfInfo

Table 6.1.6.2.13-1: Definition of type UpfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| sNssaiUpfInfoList | array(SnssaiUpfInfoItem) | M | 1..N | List of parameters supported by the UPF per S-NSSAI |
| smfServingArea | array(string) | O | 0..N | The SMF service area(s) the UPF can serve |
| interfaceUpfInfoList | array(InterfaceUpfInfoItem) | O | 0..N | List of User Plane interfaces configured on the UPF. When this IE is provided in the NF Discovery response, the NF Service Consumer (e,g SMF) may use this information for UPF selection. |

##### 6.1.6.2.14 Type: SnssaiUpfInfoItem

Table 6.1.6.2.14-1: Definition of type SnssaiUpfInfoItem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| sNssai | Snssai | M | 1 | Supported S-NSSAI |
| dnnUpfInfoList | array(DnnUpfInfoItem) | M | 1..N | List of parameters supported by the UPF per DNN |

##### 6.1.6.2.15 Type: DnnUpfInfoItem

Table 6.1.6.2.15-1: Definition of type DnnUpfInfoItem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| dnn | Dnn | M | 1 | Supported DNN |

##### 6.1.6.2.16 Type: SubscriptionData

Table 6.1.6.2.16-1: Definition of type SubscriptionData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| nfStatusNotificationUri | Uri | M | 1 | Callback URI where the NF Service Consumer will receive the notifications from NRF. |
| subscriptionId | string | C | 0..1 | Subscription ID for the newly created resource. This parameter shall only be included by NRF in the response to the subscription creation request.  Read-Only: true |
| validityTime | DateTime | C | 0..1 | Time instant after which the subscription becomes invalid. This parameter may be sent by the client, as a hint to the server, but it shall be always sent back by the server (regardless of the presence of the attribute in the request) in the response to the subscription creation request. |
| reqNotifEvents | array(NotificationEventType) | O | 0..N | If present, this attribute shall contain the list of event types that the NF Service Consumer is interested in receiving.  If this attribute is not present, it means that notifications for all event types are requested. |
| nfInstanceId | NfInstanceId | C | 0..1 | If present, this attribute contains the NF Instance ID of a specific NF Instance, whose status is requested to be monitored.  If this attribute is present, "nfType", "serviceName", "amfRegionId", "amfSetId" and "guamiList" shall be absent. |
| nfType | NFType | C | 0..1 | If present, this attribute contains the NF type of the NFs whose status is requested to be monitored.  If this attribute is present, "nfInstanceId" shall be absent. |
| serviceName | ServiceName | C | 0..1 | If present, this attribute contains the service name of those NFs that offer such service, whose status is requested to be monitored.  If this attribute is present, "nfInstanceId" shall be absent. |
| amfRegionId | string | C | 0..1 | If present, this attribute contains the AMF region identifier of the NFs whose status is requested to be monitored.  If this attribute is present, "nfInstanceId" shall be absent. |
| amfSetId | string | C | 0..1 | If present, this attribute contains the AMF set identifier of those NFs having a certain "AMF Set ID" attribute in their profile, whose status is requested to be monitored.  If this attribute is present, "nfInstanceId" shall be absent. |
| guamiList | array(Guami) | C | 0..N | If present, this attribute contains the GUAMI(s) of the AMFs whose status is requested to be monitored.  If this attribute is present, "nfInstanceId" shall be absent. |
| plmnId | PlmnId | O | 0..1 | If present, this attribute contains the target PLMN ID of the NF Instance(s) whose status is requested to be monitored. |
| NOTE 1: At least one of the subscription conditions (nfInstanceID, nfType, serviceName, amfRegionId, amfSetId or guamiList) shall be present in the SubscriptionData object.  NOTE 2: If both the amfRegionId and amfSetId attributes are present in the SubscriptionData, this indicates a subscription for notifications satisfying both attributes (i.e. notifications for NFs from that amfRegionId and amfSetId). | | | | |

##### 6.1.6.2.17 Type: NotificationData

Table 6.1.6.2.17-1: Definition of type NotificationData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| event | NotificationEventType | M | 1 | Notification type. It shall take the values "NF\_REGISTERED", "NF\_DEREGISTERED" OR "NF\_PROFILE\_CHANGED". |
| nfInstanceUri | Uri | M | 1 | Uri of the NF Instance (see subclause 6.1.3.3.2) associated to the notification event. |
| nfProfile | NFProfile | C | 0..1 | New NF Profile or Updated NF Profile; it shall be present when the notification type is "NF\_REGISTERED" or "NF\_PROFILE\_CHANGED". |

##### 6.1.6.2.18 Type: NFRegistrationData

Table 6.1.6.2.18-1: Definition of type NFRegistrationData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| heartBeatTimer | integer | M | 1 | Time in seconds expected between 2 consecutive heart-beat messages from an NF Instance to the NRF |
| nfProfile | NFProfile | M | 1 | Profile of the registered NF Instance |

##### 6.1.6.2.19 Type: NFServiceVersion

Table 6.1.6.2.19-1: Definition of type NFServiceVersion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| apiVersionInUri | string | M | 1 | Version of the service instance to be used in the URI for accessing the API (e.g. "v1"). |
| apiFullVersion | string | M | 1 | Full version number of the API as specified in subclause 4.3.1 of 3GPP TS 29.501 [5]. |
| expiry | DateTime | O | 0..1 | Expiry date and time of the NF service. This represents the planned retirement date as specified in subclause 4.3.1.5 of 3GPP TS 29.501 [5]. |

##### 6.1.6.2.20 Type: PcfInfo

Table 6.1.6.2.20-1: Definition of type PcfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| dnnList | array(Dnn) | O | 1..N | DNNs supported by the PCF |
| supiRangeList | array(SupiRange) | O | 1..N | List of ranges of SUPIs that can be served by the PCF instance. If not provided, the PCF can serve any SUPI. |

##### 6.1.6.2.21 Type: BsfInfo

Table 6.1.6.2.21-1: Definition of type BsfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| ipv4AddressRanges | array(Ipv4AddressRange) | O | 0..N | List of ranges of IPv4 addresses handled by BSF |
| ipv6PrefixRanges | array(Ipv6PrefixRange) | O | 0..N | List of ranges of IPv6 prefixes handled by BSF |

##### 6.1.6.2.22 Type: Ipv4AddressRange

Table 6.1.6.2.22-1: Definition of type IPv4AddressRange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| start | Ipv4Addr | M | 1 | First value identifying the start of an IPv4 address range |
| end | Ipv4Addr | M | 1 | Last value identifying the end of an IPv4 address range |

##### 6.1.6.2.23 Type: Ipv6PrefixRange

Table 6.1.6.2.23-1: Definition of type IPv6PrefixRange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| start | Ipv6Prefix | M | 1 | First value identifying the start of an IPv6 prefix range |
| end | Ipv6Prefix | M | 1 | Last value identifying the end of an IPv6 prefix range |

##### 6.1.6.2.24 Type: InterfaceUpfInfoItem

Table 6.1.6.2.24-1: Definition of type InterfaceUpfInfoItem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| interfaceType | UPInterfaceType | M | 1 | User Plane interface type |
| ipv4EndpointAddress | array(Ipv4Addr) | C | 0..N | Available endpoint IPv4 address(es) of the User Plane interface (NOTE 1) |
| ipv6EndpointAddress | array(Ipv6Addr) | C | 0..N | Available endpoint IPv6 address(es) of the User Plane interface (NOTE 1) |
| endpointFqdn | Fqdn | C | 0..1 | FQDN of available endpoint of the User Plane interface (NOTE 1) |
| networkInstance | string | O | 0..1 | Network Instance (See 3GPP TS 29.244 [21]) associated to the User Plane interface |
| NOTE 1: At least one of the addressing parameters (ipv4address, ipv6adress or endpointFqdn) shall be included in the InterfaceUpfInfoItem. | | | | |

##### 6.1.6.2.25 Type: UriList

Table 6.1.6.2.25-1: Definition of type UriList

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| \_links | map(LinksValueSchema) | O | 1..N | See sub-clause 4.9.4 of 3GPP TS 29.501 [5] for the description of the members. |

##### 6.1.6.2.26 Type: N2InterfaceAmfInfo

Table 6.1.6.2.26-1: Definition of type N2InterfaceAmfInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| ipv4EndpointAddress | array(Ipv4Addr) | C | 0..N | Available AMF endpoint IPv4 address(es) for N2 (see NOTE 1) |
| ipv6EndpointAddress | array(Ipv6Addr) | C | 0..N | Available AMF endpoint IPv6 address(es) for N2 (see NOTE 1) |
| amfName | AmfName | O | 0..1 | AMF Name |
| NOTE 1: At least one of the addressing parameters (ipv4address or ipv6adress) shall be included. | | | | |

##### 6.1.6.2.27 Type: TaiRange

Table 6.1.6.2.27-1: Definition of type TaiRange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| plmnId | PlmnId | M | 1 | PLMN ID related to the TacRange. |
| tacRangeList | array(TacRange) | M | 1..N | The range of the TACs |

##### 6.1.6.2.28 Type: TacRange

Table 6.1.6.2.28-1: Definition of type TacRange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| start | string | O | 0..1 | First value identifying the start of a TAC range, to be used when the range of TAC's can be represented as a hexadecimal range (e.g., TAC ranges). 3-octet string identifying a tracking area code, each character in the string shall take a value of "0" to "9" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC shall appear last in the string.  Pattern: "^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6}$)" |
| end | string | O | 0..1 | Last value identifying the end of a TAC range, to be used when the range of TAC's can be represented as a hexadecimal range (e.g. TAC ranges). 3-octet string identifying a tracking area code, each character in the string shall take a value of "0" to "9" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC shall appear last in the string.  Pattern: "^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$" |
| pattern | string | O | 0..1 | Pattern (regular expression according to the ECMA-262 dialect [8]) representing the set of TAC's belonging to this range. A TAC value is considered part of the range if and only if the TAC string fully matches the regular expression. |
| NOTE: Either the start and end attributes, or the pattern attribute, shall be present. | | | | |

EXAMPLE 1: TAC range. From: 543000 To: 5433E7 (i.e., 1000 TAC numbers)  
JSON: { "start": "543000", "end": "5433E7" }

EXAMPLE 2: TAC range. From: 54E000 To: 54EFFF (i.e., 4096 TAC numbers)  
JSON: { "pattern": "^tac-54E[0-9a-fA-F]{3}$" }, or  
JSON: { "start": "54E000", "end": "54EFFF" }

#### 6.1.6.3 Simple data types and enumerations

##### 6.1.6.3.1 Introduction

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

##### 6.1.6.3.2 Simple data types

The simple data types defined in table 6.1.6.3.2-1 shall be supported.

Table 6.1.6.3.2-1: Simple data types

|  |  |  |
| --- | --- | --- |
| Type Name | Type Definition | Description |
| Fqdn | string | FQDN (Fully Qualified Domain Name) |

##### 6.1.6.3.3 Enumeration: NFType

The enumeration NFType represents the different types of Network Functions that can be found in the 5GC.

Table 6.1.6.3.3-1: Enumeration NFType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "NRF" | Network Function: NRF |
| "UDM" | Network Function: UDM |
| "AMF" | Network Function: AMF |
| "SMF" | Network Function: SMF |
| "AUSF" | Network Function: AUSF |
| "NEF" | Network Function: NEF |
| "PCF" | Network Function: PCF |
| "SMSF" | Network Function: SMSF |
| "NSSF" | Network Function: NSSF |
| "UDR" | Network Function: UDR |
| "LMF" | Network Function: LMF |
| "GMLC" | Network Function: GMLC |
| "5G\_EIR" | Network Function: 5G-EIR |
| "SEPP" | Network Function: SEPP |
| "UPF" | Network Function: UPF |
| "N3IWF" | Network Function: N3IWF |
| "AF" | Network Function: AF |
| "UDSF" | Network Function: UDSF |
| "BSF" | Network Function: BSF |
| "CHF" | Network Function: CHF |
| "NWDAF" | Network Function: NWDAF |

##### 6.1.6.3.4 Enumeration: NotificationType

Table 6.1.6.3.4-1: Enumeration NotificationType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "N1\_MESSAGES" | Notification of N1 messages |
| "N2\_INFORMATION" | Notification of N2 information |
| "LOCATION\_NOTIFICATION" | Notification of Location Information by AMF towards NF Service Consumers (e.g GMLC) |
| "DATA\_REMOVAL\_NOTIFICATION" | Notification of Data Removal by UDR (e.g., removal of UE registration data upon subscription withdrawal) |
| "DATA\_CHANGE\_NOTIFICATION" | Notification of Data Changes by UDR |

##### 6.1.6.3.5 Enumeration: TransportProtocol

Table 6.1.6.3.5-1: Enumeration TransportProtocol

|  |  |
| --- | --- |
| Enumeration value | Description |
| "TCP" | Transport protocol: TCP |

##### 6.1.6.3.6 Enumeration: NotificationEventType

Table 6.1.6.3.6-1: Enumeration NotificationEventType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "NF\_REGISTERED" | The NF Instance has been registered in NRF |
| "NF\_DEREGISTERED" | The NF Instance has been deregistered from NRF |
| "NF\_PROFILE\_CHANGED" | The profile of the NF Instance has been modified |

##### 6.1.6.3.7 Enumeration: NFStatus

Table 6.1.6.3.7-1: Enumeration NFStatus

|  |  |
| --- | --- |
| Enumeration value | Description |
| "REGISTERED" | The NF Instance is registered in NRF and can be discovered by other NFs |
| "SUSPENDED" | The NF Instance is registered in NRF but it is not operative and cannot be discovered by other NFs |

##### 6.1.6.3.8 Enumeration: DataSetId

The enumeration DataSetId represents the different types of data sets supported by an UDR instance.

Table 6.1.6.3.8-1: Enumeration DataSetId

|  |  |
| --- | --- |
| Enumeration value | Description |
| "SUBSCRIPTION" | Data set: Subscription data |
| "POLICY" | Data set: Policy data |
| "EXPOSURE" | Data set: Structured data for exposure |
| "APPLICATION" | Data set: Application data |

##### 6.1.6.3.9 Enumeration: UPInterfaceType

Table 6.1.6.3.9-1: Enumeration UPInterfaceType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "N3" | User Plane Interface: N3 |
| "N6" | User Plane Interface: N6 |
| "N9" | User Plane Interface: N9 |

##### 6.1.6.3.10 Relation Types

###### 6.1.6.3.10.1 General

This clause describes the possible relation types defined within NRF API. See sub-clause 4.7.5.2 of 3GPP TS 29.501 [5] for the description of the relation types.

Table 6.1.6.3.10.1-1: supported registered relation types

|  |
| --- |
| Relation Name |
| self |
| item |

##### 6.1.6.3.11 Enumeration: ServiceName

Table 6.1.6.3.11-1: Enumeration ServiceName

|  |  |
| --- | --- |
| Enumeration value | Description |
| "nnrf-nfm" | Nnrf\_NFManagement Service offered by the NRF |
| "nnrf-disc" | Nnrf\_NFDiscovery Service offered by the NRF |
| "nudm-sdm" | Nudm\_SubscriberDataManagement Service offered by the UDM |
| "nudm-uecm" | Nudm\_UEContextManagement Service offered by the UDM |
| "nudm-ueau" | Nudm\_UEAuthentication Service offered by the UDM |
| "nudm-ee" | Nudm\_EventExposure Service offered by the UDM |
| "nudm-pp" | Nudm\_ParameterProvision Service offered by the UDM |
| "namf-comm" | Namf\_Communication Service offered by the AMF |
| "namf-evts" | Namf\_EventExposure Service offered by the AMF |
| "namf-mt" | Namf\_MT Service offered by the AMF |
| "namf-location" | Namf\_Location Service offered by the AMF |
| "nsmf-pdusession" | Nsmf\_PDUSession Service offered by the SMF |
| "nsmf-event-exposure" | Nsmf\_EventExposure Service offered by the SMF |
| "nausf-auth" | Nausf\_UEAuthentication Service offered by the AUSF |
| "nausf-sorprotection" | Nausf\_SoRProtection Service offered by the AUSF |
| "nnef-pfdmanagement" | Nnef\_PFDManagement offered by the NEF |
| "npcf-am-policy-control" | Npcf\_AMPolicyControl Service offered by the PCF |
| "npcf-smpolicycontrol" | Npcf\_SMPolicyControl Service offered by the PCF |
| "npcf-policyauthorization" | Npcf\_PolicyAuthorization Service offered by the PCF |
| "npcf-bdtpolicycontrol" | Npcf\_BDTPolicyControl Service offered by the PCF |
| "nsmsf-sms" | Nsmsf\_SMService Service offered by the SMSF |
| "nnssf-nsselection" | Nnssf\_NSSelection Service offered by the NSSF |
| "nnssf-nssaiavailability" | Nnssf\_NSSAIAvailability Service offered by the NSSF |
| "nudr-dr" | Nudr\_DataRepository Service offered by the UDR |
| "nlmf-loc" | Nlmf\_Location Service offered by the LMF |
| "n5g-eir-eic" | N5g-eir\_ EquipmentIdentityCheck Service offered by the 5G-EIR |
| "nbsf-management" | Nbsf\_Management Service offered by the BSF |
| "nchf-spendinglimitcontrol" | Nchf\_SpendingLimitControl Service offered by the CHF |
| "nnwdaf-eventssubscription" | Nnwdaf\_EventsSubscription Service offered by the NWDAF |
| "nnwdaf-analyticsinfo" | Nnwdaf\_AnalyticsInfo Service offered by the NWDAF |
| NOTE: The services defined in this table are those defined by 3GPP NFs in 5GC; however, in order to support custom services offered by standard and custom NFs, the NRF shall also accept the registration of NF Services with other service names. | |

##### 6.1.6.3.12 Enumeration: NFServiceStatus

Table 6.1.6.3.12-1: Enumeration NFServiceStatus

|  |  |
| --- | --- |
| Enumeration value | Description |
| "REGISTERED" | The NF Service Instance is registered in NRF and can be discovered by other NFs |
| "SUSPENDED" | The NF Service Instance is registered in NRF but it is not operative and cannot be discovered by other NFs |

### 6.1.7 Error Handling

#### 6.1.7.1 General

HTTP error handling shall be supported as specified in subclause 5.2.4 of 3GPP TS 29.500 [4].

#### 6.1.7.2 Protocol Errors

Protocol errors handling shall be supported as specified in subclause 5.2.7 of 3GPP TS 29.500 [4].

#### 6.1.7.3 Application Errors

The application errors defined for the Nnmf\_NFManagement service are listed in Table 6.1.7.3-1.

Table 6.1.7.3-1: Application errors

|  |  |  |
| --- | --- | --- |
| Application Error | HTTP status code | Description |
|  |  |  |

### 6.1.8 Security

As indicated in 3GPP TS 33.501 [15], the access to the Nnrf\_NFManagement API shall be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [16]), using the "Client Credentials" authorization grant, where the NRF plays the role of the authorization server.

An NF Service Consumer, prior to consuming services offered by the Nnrf\_NFManagement API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in subclause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF where the Nnrf\_NFManagement service is invoked by the NF Service Producer.

The Nnrf\_NFManagement API does not define any scopes for OAuth2 authorization.

## 6.2 Nnrf\_NFDiscovery Service API

### 6.2.1 API URI

URIs of this API shall have the following root:

{apiRoot}/{apiName}/{apiVersion}/

where the "apiName" shall be set to "nnrf-disc" and the "apiVersion" shall be set to "v1" for the current version of this specification.

### 6.2.2 Usage of HTTP

#### 6.2.2.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in subclause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies for the Nnrf\_NFDiscovery service shall comply with the OpenAPI [10] specification contained in Annex A.

#### 6.2.2.2 HTTP Standard Headers

##### 6.2.2.2.1 General

The mandatory standard HTTP headers as specified in subclause 5.2.2.2 of 3GPP TS 29.500 [4] shall be supported.

##### 6.2.2.2.2 Content type

The following content types shall be supported:

- The JSON format (IETF RFC 8259 [x]). The use of the JSON format shall be signalled by the content type "application/json". See also subclause 5.4 of 3GPP TS 29.500 [4].

- The Problem Details JSON Object (IETF RFC 7807 [11]). The use of the Problem Details JSON object in a HTTP response body shall be signalled by the content type "application/problem+json".

##### 6.2.2.2.3 Cache-Control

A "Cache-Control" header should be included in HTTP responses, as described in IETF RFC 7234 [20], section 5.2. It shall contain a "max-age" value, indicating the amount of time in seconds after which the received response is considered stale; this value shall be the same as the content of the "validityPeriod" element described in subclause 6.2.6.2.2.

##### 6.2.2.2.4 ETag

An "ETag" (entity-tag) header should be included in HTTP responses, as described in IETF RFC 7232 [19], section 2.3. It shall contain a server-generated strong validator, that allows further matching of this value (included in subsequent client requests) with a given resource representation stored in the server or in a cache.

##### 6.2.2.2.5 If-None-Match

An NF Service Consumer should issue conditional GET request towards NRF, by including an If-None-Match header in HTTP requests, as described in IETF RFC 7232 [19], section 3.2, containing one or several entity tags received in previous responses for the same resource.

#### 6.2.2.3 HTTP custom headers

##### 6.2.2.3.1 General

In this release of this specification, no custom headers specific to the Nnrf\_NFDiscovery service are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see subclause 5.2.3 of 3GPP TS 29.500 [4].

### 6.2.3 Resources

#### 6.2.3.1 Overview

The structure of the Resource URIs of the NFDiscovery service is shown in figure 6.2.3.1-1.



Figure 6.2.3.1-1: Resource URI structure of the NFDiscovery API

Table 6.2.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.2.3.1-1: Resources and methods overview

|  |  |  |  |
| --- | --- | --- | --- |
| Resource name | Resource URI | HTTP method or custom operation | Description |
| nf-instances  (Store) | {apiRoot}/nnrf-disc/v1/nf-instances | GET | Retrieve a collection of NF Instances according to certain filter criteria. |

#### 6.2.3.2 Resource: nf-instances (Store)

##### 6.2.3.2.1 Description

This resource represents a collection of the different NF instances registered in the NRF.

This resource is modelled as the Store resource archetype (see subclause C.3 of 3GPP TS 29.501 [5]).

##### 6.2.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nnrf-disc/v1/nf-instances**

This resource shall support the resource URI variables defined in table 6.1.3.2.2-1.

Table 6.2.3.2.2-1: Resource URI variables for this resource

|  |  |
| --- | --- |
| Name | Definition |
| apiRoot | See subclause 6.1.1 |

##### 6.2.3.2.3 Resource Standard Methods

###### 6.2.3.2.3.1 GET

This operation retrieves a list of NF Instances, and their offered services, currently registered in the NRF, satisfying a number of filter criteria, such as those NF Instances offering a certain service name, or those NF Instances of a given NF type (e.g., AMF).

Table 6.2.3.2.3.1-1: URI query parameters supported by the GET method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| target-nf-type | NFType | M | 1 | This IE shall contain the NF type of the NF Service Producer being discovered. |
| requester-nf-type | NFType | M | 1 | This IE shall contain the NF type of the NF Service Consumer that is invoking the Nnrf\_NFDiscovery service. |
| service-names | array(string) | O | 1..N | If included, this IE shall contain an array of service names for which the NRF is queried to provide the list of NF profiles. The NRF shall return the NF profiles that have at least one NF service matching the NF service names in this list. The NF service names returned by the NRF shall be an intersection of the NF service names requested and the NF service names registered in the NF profile.  If not included, the NRF shall return all the NF service names registered in the NF profile. |
| requester-nf-instance-fqdn | Fqdn | O | 0..1 | If included, this IE shall contain the FQDN of the NF Service Consumer that is invoking the Nnrf\_NFDiscovery service.  The NRF shall use this to return only those NF profiles that include at least one NF service containing an entry in the "allowedNfDomains" list (see subclause 6.1.6.2.3) that matches the domain of the requester NF. |
| target-plmn | PlmnId | C | 0..1 | This IE shall be included when NF services in a different PLMN need to be discovered. When included, this IE shall contain the PLMN ID of the different PLMN. |
| requester-plmn | PlmnId | C | 0..1 | This IE shall be included when NF services in a different PLMN need to be discovered. When included, this IE shall contain the PLMN ID of the requester NF. |
| target-nf-instance-id | NfInstanceId | O | 0..1 | Identity of the NF instance being discovered. |
| target-nf-fqdn | Fqdn | O | 0..1 | FQDN of the target NF instance being discovered. |
| hnrf-uri | Uri | C | 0..1 | If included, this IE shall contain the API URI of the NFDiscovery Service (see subclause 6.2.1) of the home NRF. It shall be included if the NF Service Consumer has previously received such API URI to be used for service discovery (e.g., from the NSSF in the home PLMN). |
| snssais | array(Snssai) | O | 0..N | If included, this IE shall contain the list of S-NSSAI that are served by the services being discovered. The NRF shall use this to identify the NF services that have registered their support for these S-NSSAIs. The NRF shall return the NF profiles that have at least one S-NSSAI matching the S-NSSAIs in this list. The S-NSSAIs included in the NF services returned by the NRF shall be an intersection of the S-NSSAIs requested and the S-NSSAIs registered in the NF profile. |
| nsi-list | array(string) | O | 0..N | If included, this IE shall contain the list of NSI IDs that are served by the services being discovered. |
| dnn | Dnn | O | 0..1 | If included, this IE shall contain the DNN for which NF services serving that DNN is discovered. DNN may be included if the target NF type is "SMF" or "UPF". |
| smf-serving-area | string | O | 0..1 | If included, this IE shall contain the serving area of the SMF. It may be included if the target NF type is "UPF". |
| tai | Tai | O | 0..1 | Tracking Area Identity. |
| amf-region-id | string | O | 0..1 | AMF Region Identity. |
| amf-set-id | string | O | 0..1 | AMF Set Identity. |
| guami | Guami | O | 0..1 | Guami used to search for an appropriate AMF.  (NOTE 1) |
| supi | Supi | O | 0..1 | If included, this IE shall contain the SUPI of the requester UE to search for an appropriate NF. SUPI may be included if the target NF type is e.g. "PCF", "AUSF", "UDM" or "UDR". |
| ue-ipv4-address | Ipv4Addr | O | 0..1 | The IPv4 address of the UE for which a BSF needs to be discovered. |
| ue-ipv6-prefix | Ipv6Prefix | O | 0..1 | The IPv6 prefix of the UE for which a BSF needs to be discovered. |
| pgw-ind | boolean | O | 0..1 | When present, this IE indicates whether a combined SMF/PGW-C or a standalone SMF needs to be discovered.  true: A combined SMF/PGW-C is requested to be discovered; false: A standalone SMF is requested to be discovered. (See NOTE 2) |
| pgw | Fqdn | O | 0..1 | If included, this IE shall contain the PGW FQDN which is received by the AMF from the MME to find the combined SMF/PGW. |
| gpsi | Gpsi | O | 0..1 | If included, this IE shall contain the GPSI of the requester UE to search for an appropriate NF. GPSI may be included if the target NF type is "UDM" or "UDR". |
| external-group-identity | GroupId | O | 0..1 | If included, this IE shall contain the external group identifier of the requester UE to search for an appropriate NF. This may be included if the target NF type is "UDM" or "UDR". |
| data-set | DataSetId | O | 0..1 | Indicates the data set to be supported by the NF to be discovered. May be included if the target NF type is "UDR". |
| routing-indicator | string | O | 0..1 | Routing Indicator information that allows to route network signalling with SUCI (see 3GPP TS 23.003 [12]) to an AUSF and UDM instance capable to serve the subscriber. May be included if the target NF type is "AUSF" or "UDM". |
| group-id-list | array(string) | O | 1..N | Identity of the group(s) of the NFs of the target NF type to be discovered. May be included if the target NF type is "UDR", "UDM" or "AUSF". |
| supported-features | SupportedFeatures | O | 0..1 | List of features required to be supported by the target Network Function. |
| NOTE 1: If this parameter is present and no AMF supporting the requested GUAMI is available due to AMF Failure or planned AMF removal, the NRF shall return in the response AMF instances acting as a backup for AMF failure or planned AMF removal respectively for this GUAMI. The NRF can detect if an AMF has failed, using the Heartbeat procedure. The NRF will receive a de-registration request from an AMF performing a planned removal.  NOTE 2: If the combined SMF/PGW-C is requested to be discovered, the NRF shall return in the response the SMF instances registered with the SmfInfo containing pgwFqdn. | | | | |

This method shall support the request data structures specified in table 6.1.3.2.3.1-2 and the response data structures and response codes specified in table 6.1.3.2.3.1-3.

Table 6.2.3.2.3.1-2: Data structures supported by the GET Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 6.2.3.2.3.1-3: Data structures supported by the GET Response Body on this resource

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data type | P | | Cardinality | Response  codes | Description |
| SearchResult | M | | 1 | 200 OK | The response body contains the result of the search over the list of registered NF Instances. |
| n/a |  | |  | 307 Temporary Redirect | The response shall be used when the intermediate NRF redirects the service discovery request. |
| ProblemDetails | M | | 1 | 400 Bad Request | The response body contains the error reason of the request message. |
| ProblemDetails | | M | 1 | 403 Forbidden | This response shall be returned if the NF Service Consumer is not allowed to discover the NF Service(s) being queried. |
| ProblemDetails | M | | 1 | 500 Internal Server Error | The response body contains the error reason of the request message. |

##### 6.2.3.2.4 Resource Custom Operations

There are no resource custom operations for the Nnrf\_NFDiscovery service in this release of the specification.

### 6.2.4 Custom Operations without associated resources

There are no custom operations defined without any associated resources for the Nnrf\_NFDiscovery service in this release of this specification.

### 6.2.5 Notifications

There are no notifications defined for the Nnrf\_NFDiscovery service in this release of the specification.

### 6.2.6 Data Model

#### 6.2.6.1 General

This subclause specifies the application data model supported by the API.

Table 6.2.6.1-1 specifies the data types defined for the Nnrf service based interface protocol.

Table 6.2.6.1-1: Nnrf\_NFDiscovery specific Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Section defined | Description |
| SearchResult | 6.2.6.2.2 |  |
| NFProfile | 6.2.6.2.3 |  |
| NFService | 6.2.6.2.4 |  |

Table 6.2.6.1-2 specifies data types re-used by the Nnrf service based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nnrf service based interface.

Table 6.2.6.1-2: Nnrf\_NFDiscovery re-used Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Reference | Comments |
| Snssai | 3GPP TS 29.571 [7] |  |
| PlmnId | 3GPP TS 29.571 [7] |  |
| Dnn | 3GPP TS 29.571 [7] |  |
| Tai | 3GPP TS 29.571 [7] |  |
| SupportedFeatures | 3GPP TS 29.571 [7] |  |
| NfInstanceId | 3GPP TS 29.571 [7] |  |
| Uri | 3GPP TS 29.571 [7] |  |
| Gpsi | 3GPP TS 29.571 [7] |  |
| GroupId | 3GPP TS 29.571 [7] |  |
| Guami | 3GPP TS 29.571 [7] |  |
| IPv4Addr | 3GPP TS 29.571 [7] |  |
| IPv6Addr | 3GPP TS 29.571 [7] |  |
| UriScheme | 3GPP TS 29.571 [7] |  |
| DefaultNotificationSubscription | 3GPP TS 29.510 | See clause 6.1.6.2.4 |
| IPEndPoint | 3GPP TS 29.510 | See clause 6.1.6.2.5 |
| NFType | 3GPP TS 29.510 | See clause 6.1.6.3.3 |
| UdrInfo | 3GPP TS 29.510 | See clause 6.1.6.2.6 |
| UdmInfo | 3GPP TS 29.510 | See clause 6.1.6.2.7 |
| AusfInfo | 3GPP TS 29.510 | See clause 6.1.6.2.8 |
| SupiRange | 3GPP TS 29.510 | See clause 6.1.6.2.9 |
| AmfInfo | 3GPP TS 29.510 | See clause 6.1.6.2.11 |
| SmfInfo | 3GPP TS 29.510 | See clause 6.1.6.2.12 |
| UpfInfo | 3GPP TS 29.510 | See clause 6.1.6.2.13 |
| PcfInfo | 3GPP TS 29.510 | See clause 6.1.6.2.20 |
| BsfInfo | 3GPP TS 29.510 | See clause 6.1.6.2.21 |
| NFServiceVersion | 3GPP TS 29.510 | See clause 6.1.6.2.19 |
| NFStatus | 3GPP TS 29.510 | See clause 6.1.6.3.7 |
| DataSetId | 3GPP TS 29.510 | See clause 6.1.6.3.8 |
| ServiceName | 3GPP TS 29.510 | See clause 6.1.6.3.11 |
| NFServiceStatus | 3GPP TS 29.510 | See clause 6.1.6.3.12 |

#### 6.2.6.2 Structured data types

##### 6.2.6.2.1 Introduction

This subclause defines the structures to be used in resource representations.

##### 6.2.6.2.2 Type: SearchResult

Table 6.2.6.2.2-1: Definition of type SearchResult

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| validityPeriod | integer | M | 1 | It shall contain the time in seconds during which the discovery result is considered valid and can be cached by the NF Service Consumer. This value shall be the same as the value contained in the "max-age" parameter of the "Cache-Control" header field sent in the HTTP response. |
| nfInstances | array(NFProfile) | 0 | 0..N | If present, it shall contain an array of NF Instance profiles, matching the search criteria indicated by the query parameters of the discovery request. |

##### 6.2.6.2.3 Type: NFProfile

Table 6.2.6.2.3-1: Definition of type NFProfile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| nfInstanceID | NfInstanceId | M | 1 | Unique identity of the NF Instance. |
| nfType | NFType | M | 1 | Type of Network Function |
| nfStatus | NFStatus | M | 1 | Status of the NF Instance |
| plmn | PlmnId | O | 0..1 | PLMN of the Network Function |
| sNssais | array(Snssai) | O | 0..N | S-NSSAIs of the Network Function |
| nsiList | array(string) | O | 0..N | List of NSIs of the Network Function |
| fqdn | Fqdn | C | 0..1 | FQDN of the Network Function (NOTE 1, NOTE 3) |
| ipv4Addresses | array(Ipv4Addr) | C | 0..N | IPv4 address(es) of the Network Function (NOTE 1) |
| ipv6Addresses | array(Ipv6Addr) | C | 0..N | IPv6 address(es) of the Network Function (NOTE 1) |
| capacity | integer | O | 0..1 | 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. (See NOTE 2) |
| load | integer | O | 0..1 | Latest known load information of the NF ranged from 0 to 100 in percentage (See NOTE 4) |
| locality | string | O | 0..1 | Operator defined information about the location of the NF instance (e.g. geographic location, data center) |
| priority | integer | O | 0..1 | 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. (See NOTE 2) |
| udrInfo | UdrInfo | O | 0..1 | Specific data for the UDR (ranges of SUPI, …) |
| udmInfo | UdmInfo | O | 0..1 | Specific data for the UDM |
| ausfInfo | AusfInfo | O | 0..1 | Specific data for the AUSF |
| amfInfo | AmfInfo | O | 0..1 | Specific data for the AMF (AMF Set ID, …) |
| smfInfo | smfInfo | O | 0..1 | Specific data for the SMF (DNN's, …) |
| upfInfo | UpfInfo | O | 0..1 | Specific data for the UPF (S-NSSAI, DNN, SMF serving area, …) |
| pcfInfo | PcfInfo | O | 0..1 | Specific data for the PCF |
| bsfInfo | BsfInfo | O | 0..1 | Specific data for the BSF |
| customInfo | object | O | 0..1 | Specific data for custom Network Functions |
| recoveryTime | DateTime | O | 0..1 | Timestamp when the NF was (re)started |
| nfServices | array(NFService) | O | 0..N | List of NF Service Instances |
| NOTE 1: At least one of the addressing parameters (fqdn, ipv4address or ipv6adress) shall be included in the NF Profile.  NOTE 2: The capacity and priority parameters, if present, are used for NF selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].  NOTE 3: If the requester-plmn in the query parameter is different from the PLMN of the discovered NF, then the fqdn attribute value shall contain the interPlmnFqdn value registered by the NF during NF registration (see subclause 6.1.6.2.2).  NOTE 4: The usage of the load parameter by the NF service consumer is implementation specific, e.g. be used for NF selection and load balancing, together with other parameters. | | | | |

##### 6.2.6.2.4 Type: NFService

Table 6.2.6.2.4-1: Definition of type NFService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| serviceInstanceID | string | M | 1 | Unique ID of the service instance within a given NF Instance |
| serviceName | ServiceName | M | 1 | Name of the service instance (e.g. "udm-sdm") |
| versions | array(NFServiceVersion) | M | 1..N | The API versions supported by the NF Service and if available, the corresponding retirement date of the NF Service.  The different array elements shall have distinct unique values for "apiVersionInUri", and consequently, the values of "apiFullVersion" shall have a unique first digit version number. |
| scheme | UriScheme | M | 1 | URI scheme (e.g. "http", "https") |
| nfServiceStatus | NFServiceStatus | M | 0..1 | Status of the NF Service Instance |
| fqdn | string | O | 0..1 | FQDN of the NF where the service is hosted (see NOTE 1, NOTE 3) |
| ipEndPoints | array(IpEndPoint) | O | 0..N | IP address(es) and port information of the Network Function (including IPv4 and/or IPv6 address) where the service is listening for incoming service requests (see NOTE 1, NOTE 5) |
| apiPrefix | string | O | 0..1 | Optional path segment(s) used to construct the {apiRoot} variable of the different API URIs, as described in 3GPP TS 29.501 [5], subclause 4.4.1 (optional deployment-specific string that starts with a "/" character) |
| defaultNotificationSubscriptions | array(DefaultNotificationSubscription) | O | 0..N | Notification endpoints for different notification types. |
| capacity | integer | O | 0..1 | Static capacity information in the range of 0-65535, expressed as a weight relative to other services of the same type. (See NOTE 2) |
| load | integer | O | 0..1 | Latest known load information of the NF Service, ranged from 0 to 100 in percentage. (See NOTE 4) |
| priority | integer | O | 0..1 | 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. (See NOTE 2) |
| recoveryTime | DateTime | O | 0..1 | Timestamp when the NF service was (re)started |
| supportedFeatures | SupportedFeatures | O | 0..1 | Supported Features of the NF Service instance |
| NOTE 1: If both the fqdn and ipEndpoints attributes are not present, the FQDN and IP address related attributes from the NF Profile shall be used to construct the API URIs of this service.  NOTE 2: The capacity and priority parameters, if present, are used for service selection and load balancing. The priority and capacity attributes shall be used for NF selection in the same way that priority and weight are used for server selection as defined in IETF RFC 2782 [23].  NOTE 3: If the requester-plmn in the query parameter is different from the PLMN of the discovered NF Service, then the fqdn attribute value, if included shall contain the interPlmnFqdn value registered by the NF Service during NF registration (see subclause 6.1.6.2.3).  NOTE 4: The usage of the load parameter by the NF service consumer is implementation specific, e.g. be used for NF service selection and load balancing, together with other parameters.  NOTE 5: If the ipEndPoints attribute is absent in the NF Service and NF Profile, the NF service consumer shall use the fqdn attribute value for DNS query and if the NF service consumer does not receive a port number during the DNS query it shall use the default HTTP port number, i.e. TCP port 80 for "http" URIs or TCP port 443 for "https" URIs as specified in IETF RFC 7540 [9] when invoking the service. | | | | |

#### 6.2.6.3 Simple data types and enumerations

##### 6.2.6.3.1 Introduction

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

##### 6.2.6.3.2 Simple data types

The simple data types defined in table 6.2.6.3.2-1 shall be supported.

Table 6.2.6.3.2-1: Simple data types

|  |  |  |
| --- | --- | --- |
| Type Name | Type Definition | Description |
|  |  |  |

### 6.2.7 Error Handling

#### 6.2.7.1 General

HTTP error handling shall be supported as specified in subclause 5.2.4 of 3GPP TS 29.500 [4].

#### 6.2.7.2 Protocol Errors

Protocol errors handling shall be supported as specified in subclause 5.2.7 of 3GPP TS 29.500 [4].

#### 6.2.7.3 Application Errors

The application errors defined for the Nnmf\_NFDiscovery service are listed in Table 6.2.7.3-1.

Table 6.2.7.3-1: Application errors

|  |  |  |
| --- | --- | --- |
| Application Error | HTTP status code | Description |
|  |  |  |

### 6.2.8 Security

As indicated in 3GPP TS 33.501 [15], the access to the Nnrf\_NFDiscovery API shall be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [16]), using the "Client Credentials" authorization grant, where the NRF plays the role of the authorization server.

An NF Service Consumer, prior to consuming services offered by the Nnrf\_NFDiscovery API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in subclause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF where the Nnrf\_NFDiscovery service is invoked by the NF Service Consumer.

The Nnrf\_NFDiscovery API does not define any scopes for OAuth2 authorization.

## 6.3 OAuth2.0 Authorization Service API

### 6.3.1 General

This API reuses the API endpoints and input / output parameters specified in IETF RFC 6749 [16] as a custom operation without resources. Hence this clause does not follow the 3GPP specific API specification guideline specified in 3GPP TS 29.501 [5].

### 6.3.2 API URI

URIs of this API shall have the following root:

{nrfApiRoot}/oauth2/

where {nrfApiRoot} represents the concatenation of the "scheme" and "authority" components of the NRF, as defined in IETF RFC 3986 [17].

### 6.3.3 Usage of HTTP

#### 6.3.3.1 General

HTTP/2, as defined in IETF RFC 7540 [9], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in subclause 5.3 of 3GPP TS 29.500 [4].

HTTP messages and bodies this API shall comply with the OpenAPI [10] specification contained in Annex A.

#### 6.3.3.2 HTTP standard headers

##### 6.3.3.2.1 General

The HTTP headers as specified in subclause 4.4 of IETF RFC 6749 [16] shall be supported. The "Authorization" header shall be supported for HTTP Basic Authentication mechanism (see subclause 5.4.2.2.1).

##### 6.3.3.2.2 Content type

The following content types shall be supported:

- The x-www-form-urlencoded format (see subclause 17.13.4.1 of W3C HTML 4.01 Specification [26]). The use of the x-www-form-urlencoded format shall be signalled by the content type "application/x-www-form-urlencoded".

- The JSON format (IETF RFC 8259 [22]). The use of the JSON format shall be signalled by the content type "application/json". See also subclause 5.4 of 3GPP TS 29.500 [4].

#### 6.3.3.3 HTTP custom headers

##### 6.3.3.3.1 General

In this release of this specification, no custom headers specific to the OAuth2.0 Authorization Service API are defined. For 3GPP specific HTTP custom headers used across all service-based interfaces, see clause 5.2.3 of 3GPP TS 29.500 [4].

### 6.3.4 Custom Operation without Associated Resources

#### 6.3.4.1 Overview

The /token endpoint as specified in IETF RFC 6749 [16] shall be supported. The "token endpoint" URI shall be:

{apiRoot}/token

where {apiRoot} is defined in subclause 6.3.1.

Table 6.3.4.1-1 provides an overview of the endpoints and applicable HTTP methods.

Table 6.3.4.1-1: Custom operations without associated resources

|  |  |  |
| --- | --- | --- |
| Custom operation URI | Mapped HTTP method | Description |
| {apiRoot}/token | POST | Access token request for obtaining OAuth2.0 access token |

#### 6.3.4.2 Operation: Access Token Request

##### 6.3.4.2.1 Description

This custom operation represents the process for issuing the OAuth2.0 access token.

##### 6.3.4.2.2 Operation Definition

This operation returns an OAuth 2.0 access token based on the input parameters provided. This custom operation shall use the HTTP POST method.

This method shall support the request data structures specified in table 6.3.4.2.2-1 and the response data structures and response codes specified in table 6.3.4.2.2-2. The data structure used for the POST request body shall be using x-www-urlencoded format as specified in subclause 17.13.4.1 of W3C HTML 4.01 Specification [26].

Table 6.3.4.2.2-1: Data structures supported by the POST Request Body on this endpoint

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| AccessTokenReq | M | 1 | This IE shall contain the request information for the access token request. |

Table 6.3.4.2.2-2: Data structures supported by the POST Response Body on this endpoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| AccessTokenRsp | M | 1 | 200 OK | This IE shall contain the access token response information. |
| AccessTokenErr | M | 1 | 400 Bad Request | See IETF RFC 6749 [16] subclause 5.2. |

### 6.3.5 Data Model

#### 6.3.5.1 General

This subclause specifies the application data model supported by the API.

Table 6.3.5.1-1 specifies the data types defined for the OAuth 2.0 Authorization Service API. The AccessTokenReq data structure shall be converted to the content type "application/x-www-urlencoded" when the OAuth 2.0 Access Token Request is invoked.

Table 6.3.5.1-1: OAuth 2.0 Authorization service specific Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Section defined | Description |
| AccessTokenReq | 6.3.5.2.2 | Data type for carrying information related to access token request. |
| AccessTokenRsp | 6.3.5.2.3 | Data type for carrying information related to access token response. |
| AccessTokenClaims | 6.3.5.2.4 | The claims data structure for the access token. |

Table 6.3.5.1-2 specifies data types re-used by the OAuth 2.0 Authorization service from other specifications, including a reference to their respective specifications and when needed, a short description of their use.

Table 6.3.5.1-2: OAuth 2.0 Authorization service re-used Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Reference | Comments |
| NfInstanceId | 3GPP TS 29.571 [7] |  |
| NFType | 3GPP TS 29.510 | See subclause 6.1.6.3.3 |

#### 6.3.5.2 Structured data types

##### 6.3.5.2.1 Introduction

This subclause defines the structures to be used in the APIs.

##### 6.3.5.2.2 Type: AccessTokenReq

Table 6.3.5.2.2-1: Definition of type AccessTokenReq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| grant\_type | GrantType | M | 1 | This IE shall contain the grant type as "client\_credentials" |
| nfInstanceId | NfInstanceId | M | 1 | This IE shall contain the NF instance id of the NF service consumer. |
| nfType | NFType | M | 1 | This IE shall contain the NF type of the NF service consumer. |
| targetNfType | NFType | M | 1 | This IE shall contain the NF type of the NF service producer. |
| scope | string | M | 1 | This IE shall contain the NF service name of the NF service producer. The service name may include a wildcard.  pattern: '^([a-zA-Z0-9\_]\*[\*]{0,1})$'  See NOTE 2. |
| NOTE 1: This data structure shall not be treated as a JSON object. It shall be treated as a key, value pair data structure to be encoded using x-www-urlencoded format as specified in subclause 17.13.4.1 of W3C HTML 4.01 Specification [26].  NOTE 2: Though scope attribute is optional as per IETF RFC 6749 [16], it is mandatory for 3GPP as per 3GPP TS 33.501 [15]. | | | | |

##### 6.3.5.2.3 Type: AccessTokenRsp

Table 6.3.5.2.3-1: Definition of type AccessTokenRsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| access\_token | string | M | 1 | This IE shall contain JWS Compact Serialized representation of the JWS signed JSON object containing AccessTokenClaims (see subclause 6.3.5.2.3). |
| token\_type | string | M | 1 | This IE shall contain the token type (e.g "JWT") |
| expires\_in | string | O | 0..1 | Thie IE when present shall contain the number of seconds after which the access\_token is considered to be expired. |
| scope | string | O | 0..1 | This IE when present shall contain the NF service name of the NF service producer. The service name may include a wildcard.  pattern: '^([a-zA-Z0-9\_]\*[\*]{0,1})$' |

##### 6.3.5.2.4 Type: AccessTokenClaims

Table 6.3.5.2.4-1: Definition of type AccessTokenClaims

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| issuer | NfInstanceId | M | 1 | This IE shall contain NF instance id of the NF service consumer. |
| subject | string | M | 1 | This IE shall contain the NF instance ID of the NF service consumer. |
| audience | array(NfInstanceId) | M | 1..N | This IE shall contain the NF service producer's NF instance ID (if the exact NF instance of the NF service producer is known) or a list of NF service instances of the requested NF type for which the claim is applicable. |
| scope | string | M | 1 | This IE shall contain the name of the NF services for which the access\_token is authorized for use. The service name may include a wildcard.  pattern: '^([a-zA-Z0-9\_]\*[\*]{0,1})$' |
| expiration | integer | M | 1 | This IE shall contain the number of seconds after which the access\_token is considered to be expired. |

#### 6.3.5.3 Simple data types and enumerations

##### 6.3.5.3.1 Introduction

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

##### 6.3.5.3.2 Simple data types

There are no specific simple data types defined in this version of this API. For the re-used data types from other specifications see subclause 6.3.5.1

##### 6.3.5.3.3 Enumeration: GrantType

Table 6.3.5.3.3-1: Enumeration GrantType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "client\_credentials" | Represents the Client Credentials grant type. |

Annex A (normative): OpenAPI specification

## A.1 General

This Annex specifies the formal definition of the Nnrf Service API(s). It consists of OpenAPI 3.0.0 specifications, in YAML format.

## A.2 Nnrf\_NFManagement API

openapi: 3.0.0

info:

version: '1.PreR15.1.0'

title: 'NRF NFManagement Service'

description: 'NRF NFManagement Service'

servers:

- url: '{apiRoot}/nnrf-nfm/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in subclause subclause 4.4 of 3GPP TS 29.501

security:

- {}

- oAuth2ClientCredentials: []

paths:

/nf-instances:

get:

summary: Retrieves a collection of NF Instances

operationId: GetNFInstances

tags:

- NF Instances (Store)

parameters:

- name: nf-type

in: query

description: Type of NF

required: false

schema:

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

- name: limit

in: query

description: How many items to return at one time

required: false

schema:

type: integer

responses:

'200':

description: Expected response to a valid request

content:

application/3gppHal+json:

schema:

type: object

properties:

\_links:

type: object

description: 'List of the URI of NF instances. It has two members whose names are item and self. The item one contains an array of URIs.'

additionalProperties:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/LinksValueSchema'

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

/nf-instances/{nfInstanceID}:

get:

summary: Read the profile of a given NF Instance

operationId: GetNFInstance

tags:

- NF Instance ID (Document)

parameters:

- name: nfInstanceID

in: path

description: Unique ID of the NF Instance

required: true

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

responses:

'200':

description: Expected response to a valid request

content:

application/json:

schema:

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

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

put:

summary: Register a new NF Instance

operationId: RegisterNFInstance

tags:

- NF Instance ID (Document)

parameters:

- name: nfInstanceID

in: path

required: true

description: Unique ID of the NF Instance to register

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

requestBody:

content:

application/json:

schema:

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

required: true

responses:

'200':

description: OK (Profile Replacement)

content:

application/json:

schema:

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

'201':

description: Expected response to a valid request

content:

application/json:

schema:

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

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

patch:

summary: Update NF Instance profile

operationId: UpdateNFInstance

tags:

- NF Instance ID (Document)

parameters:

- name: nfInstanceID

in: path

required: true

description: Unique ID of the NF Instance to update

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

requestBody:

content:

application/json-patch+json:

schema:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PatchItem'

required: true

responses:

'200':

description: Expected response to a valid request

content:

application/json:

schema:

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

'204':

description: Expected response with empty body

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

delete:

summary: Deregisters a given NF Instance

operationId: DeregisterNFInstance

tags:

- NF Instance ID (Document)

parameters:

- name: nfInstanceID

in: path

required: true

description: Unique ID of the NF Instance to deregister

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

responses:

'204':

description: Expected response to a successful deregistration

/subscriptions:

post:

summary: Create a new subscription

operationId: CreateSubscription

tags:

- Subscriptions (Collection)

requestBody:

content:

application/json:

schema:

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

required: true

responses:

'201':

description: Expected response to a valid request

content:

application/json:

schema:

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

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

callbacks:

onNFStatusEvent:

'{$request.body#/nfStatusNotificationUri}':

post:

requestBody:

description: Notification Payload

content:

application/json:

schema:

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

responses:

'204':

description: Expected response to a successful callback processing

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

/subscriptions/{subscriptionID}:

delete:

summary: Deletes a subscription

operationId: RemoveSubscription

tags:

- Subscription ID (Document)

parameters:

- name: subscriptionID

in: path

required: true

description: Unique ID of the subscription to remove

schema:

type: string

pattern: '^([0-9]{5,6}-)?[^-]+$'

responses:

'204':

description: Expected response to a successful subscription removal

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

components:

securitySchemes:

oAuth2ClientCredentials:

type: oauth2

flows:

clientCredentials:

tokenUrl: '/oauth2/token'

scopes: {}

schemas:

NFProfile:

type: object

required:

- nfInstanceId

- nfType

- nfStatus

properties:

nfInstanceId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

nfType:

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

nfStatus:

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

plmn:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

sNssais:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Snssai'

nsiList:

type: array

items:

type: string

fqdn:

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

interPlmnFqdn:

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

ipv4Addresses:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

ipv6Addresses:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Addr'

priority:

type: integer

minimum: 0

maximum: 65535

capacity:

type: integer

minimum: 0

maximum: 65535

load:

type: integer

minimum: 0

maximum: 100

locality:

type: string

udrInfo:

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

udmInfo:

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

ausfInfo:

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

amfInfo:

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

smfInfo:

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

upfInfo:

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

pcfInfo:

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

bsfInfo:

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

customInfo:

type: object

recoveryTime:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

nfServices:

type: array

items:

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

NFService:

type: object

required:

- serviceInstanceId

- serviceName

- versions

- scheme

- nfServiceStatus

properties:

serviceInstanceId:

type: string

serviceName:

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

versions:

type: array

items:

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

scheme:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/UriScheme'

nfServiceStatus:

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

fqdn:

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

interPlmnFqdn:

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

ipEndPoints:

type: array

items:

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

apiPrefix:

type: string

defaultNotificationSubscriptions:

type: array

items:

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

allowedPlmns:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

allowedNfTypes:

type: array

items:

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

allowedNfDomains:

type: array

items:

type: string

allowedNssais:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Snssai'

priority:

type: integer

minimum: 0

maximum: 65535

capacity:

type: integer

minimum: 0

maximum: 65535

load:

type: integer

minimum: 0

maximum: 100

recoveryTime:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

NFType:

anyOf:

- type: string

enum:

- NRF

- UDM

- AMF

- SMF

- AUSF

- NEF

- PCF

- SMSF

- NSSF

- UDR

- LMF

- GMLC

- 5G\_EIR

- SEPP

- UPF

- N3IWF

- AF

- UDSF

- BSF

- CHF

- NWDAF

- type: string

Fqdn:

type: string

IpEndPoint:

type: object

properties:

ipv4Address:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

ipv6Address:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Addr'

transport:

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

port:

type: integer

SubscriptionData:

type: object

required:

- nfStatusNotificationUri

- subscriptionId

properties:

nfStatusNotificationUri:

type: string

subscriptionId:

type: string

pattern: '^([0-9]{5,6}-)?[^-]+$'

readOnly: true

validityTime:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

reqNotifEvents:

type: array

items:

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

plmnId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

nfInstanceId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

nfType:

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

serviceName:

type: $ref: '#/components/schemas/ServiceName'

amfSetId:

type: string

amfRegionId:

type: string

guamiList:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Guami'

UdrInfo:

type: object

properties:

groupId:

type: string

supiRanges:

type: array

items:

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

gpsiRanges:

type: array

items:

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

externalGroupIdentityfiersRanges:

type: array

items:

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

supportedDataSets:

type: array

items:

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

SupiRange:

type: object

properties:

start:

type: string

end:

type: string

pattern:

type: string

IdentityRange:

type: object

properties:

start:

type: string

end:

type: string

pattern:

type: string

DataSetId:

anyOf:

- type: string

enum:

- SUBSCRIPTION

- POLICY

- EXPOSURE

- APPLICATION

- type: string

UdmInfo:

type: object

properties:

groupId:

type: string

supiRanges:

type: array

items:

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

gpsiRanges:

type: array

items:

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

externalGroupIdentityfiersRanges:

type: array

items:

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

routingIndicators:

type: array

items:

type: string

AusfInfo:

type: object

properties:

groupId:

type: string

supiRanges:

type: array

items:

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

routingIndicators:

type: array

items:

type: string

AmfInfo:

type: object

required:

- amfSetId

- amfRegionId

- guamiList

properties:

amfSetId:

type: string

amfRegionId:

type: string

guamiList:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Guami'

taiList:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Tai'

taiRangeList:

type: array

items:

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

backupInfoAmfFailure:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Guami'

backupInfoAmfRemoval:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Guami'

n2InterfaceAmfInfo:

$ref: '#/components/schemas/N2InterfaceAmfInfo'

SmfInfo:

type: object

required:

- dnnList

properties:

dnnList:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Dnn'

taiList:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Tai'

taiRangeList:

type: array

items:

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

pgwFqdn:

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

UpfInfo:

type: object

required:

- sNssaiUpfInfoList

properties:

sNssaiUpfInfoList:

type: array

items:

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

minItems: 1

smfServingArea:

type: array

items:

type: string

interfaceUpfInfoList:

type: array

items:

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

SnssaiUpfInfoItem:

type: object

required:

- sNssai

- dnnUpfInfoList

properties:

sNssai:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Snssai'

dnnUpfInfoList:

type: array

items:

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

minItems: 1

DnnUpfInfoItem:

type: object

required:

- dnn

properties:

dnn:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Dnn'

InterfaceUpfInfoItem:

type: object

required:

- interfaceType

properties:

interfaceType:

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

ipv4EndpointAddress:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

ipv6EndpointAddress:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Addr'

endpointFqdn:

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

networkInstance:

type: string

UPInterfaceType:

anyOf:

- type: string

enum:

- N3

- N6

- N9

- type: string

PcfInfo:

type: object

properties:

dnnList:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Dnn'

supiRangeList:

type: array

items:

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

BsfInfo:

type: object

properties:

ipv4AddressRanges:

type: array

items:

$ref: '#/components/schemas/Ipv4AddressRange'

ipv6PrefixRanges:

type: array

items:

$ref: '#/components/schemas/Ipv6PrefixRange'

Ipv4AddressRange:

type: object

properties:

start:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

end:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

Ipv6PrefixRange:

type: object

properties:

start:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Prefix'

end:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Prefix'

DefaultNotificationSubscription:

type: object

required:

- notificationType

- callbackUri

properties:

notificationType:

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

callbackUri:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Uri'

n1MessageClass:

$ref: 'TS29518\_Namf\_Communication.yaml#/components/schemas/N1MessageClass'

n2InformationClass:

$ref: 'TS29518\_Namf\_Communication.yaml#/components/schemas/N2InformationClass'

NotificationType:

anyOf:

- type: string

enum:

- N1\_MESSAGES

- N2\_INFORMATION

- LOCATION\_NOTIFICATION

- DATA\_REMOVAL\_NOTIFICATION

- DATA\_CHANGE\_NOTIFICATION

- type: string

TransportProtocol:

anyOf:

- type: string

enum:

- TCP

- type: string

NotificationEventType:

anyOf:

- type: string

enum:

- NF\_REGISTERED

- NF\_DEREGISTERED

- NF\_PROFILE\_CHANGED

- type: string

NotificationData:

type: object

required:

- event

- nfInstanceUri

properties:

event:

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

nfInstanceUri:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Uri'

newProfile:

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

NFStatus:

anyOf:

- type: string

enum:

- REGISTERED

- SUSPENDED

- type: string

NFRegistrationData:

type: object

required:

- heartBeatTimer

- nfProfile

properties:

heartBeatTimer:

type: integer

nfProfile:

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

NFServiceVersion:

type: object

required:

- apiVersionInUri

- apiFullVersion

properties:

apiVersionInUri:

type: string

apiFullVersion:

type: string

expiry:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

ServiceName:

anyOf:

- type: string

enum:

- nnrf-nfm

- nnrf-disc

- nudm-sdm

- nudm-uecm

- nudm-ueau

- nudm-ee

- nudm-pp

- namf-comm

- namf-evts

- namf-mt

- namf-location

- nsmf-pdusession

- nsmf-event-exposure

- nausf-auth

- nausf-sorprotection

- nnef-pfdmanagement

- npcf-am-policy-control

- npcf-smpolicycontrol

- npcf-policyauthorization

- npcf-bdtpolicycontrol

- nsmsf-sms

- nnssf-nsselection

- nnssf-nssaiavailability

- nudr-dr

- nlmf-loc

- n5g-eir-eic

- nbsf-management

- nchf-spendinglimitcontrol

- nnwdaf-eventsubscription

- nnwdaf-analyticsinfo

- type: string

N2InterfaceAmfInfo:

type: object

properties:

ipv4EndpointAddress:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

ipv6EndpointAddress:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Addr'

amfName:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/AmfName'

NFServiceStatus:

anyOf:

- type: string

enum:

- REGISTERED

- SUSPENDED

- type: string

TaiRange:

type: object

required:

- plmnId

properties:

plmnId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

tacRangeList:

type: array

items:

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

TacRange:

type: object

properties:

start:

type: string

pattern: '^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$'

end:

type: string

pattern: '^([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})$'

pattern:

type: string

externalDocs:

description: Documentation

url: 'http://www.3gpp.org/ftp/Specs/archive/29\_series/29.510/'

## A.3 Nnrf\_NFDiscovery API

openapi: 3.0.0

info:

version: '1.PreR15.1.0'

title: 'NRF NFDiscovery Service'

description: 'NRF NFDiscovery Service'

servers:

- url: '{apiRoot}/nnrf-disc/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in subclause subclause 4.4 of 3GPP TS 29.501

security:

- {}

- oAuth2ClientCredentials: []

paths:

/nf-instances:

get:

summary: Search a collection of NF Instances

operationId: SearchNFInstances

tags:

- NF Instances (Store)

parameters:

- name: target-nf-type

in: query

description: Type of the target NF

required: true

schema:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFType'

- name: requester-nf-type

in: query

description: Type of the requester NF

required: true

schema:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFType'

- name: service-names

in: query

description: Name of the service offered by the NF

schema:

type: array

items:

type: string

minItems: 1

style: form

explode: false

- name: requester-nf-instance-fqdn

in: query

description: FQDN of the requester NF

schema:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

- name: target-plmn

in: query

description: Id of the PLMN where the target NF is located

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

- name: requester-plmn

in: query

description: Id of the PLMN where the NF issuing the Discovery request is located

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

- name: target-nf-instance-id

in: query

description: Identity of the NF instance being discovered

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

- name: target-nf-fqdn

in: query

description: FQDN of the NF instance being discovered

schema:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

- name: hnrf-uri

in: query

description: Uri of the home NRF

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Uri'

- name: snssais

in: query

description: Slice info of the target NF

content:

application/json:

schema:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Snssai'

- name: dnn

in: query

description: Dnn supported by the SMF

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Dnn'

- name: nsi-list

in: query

description: NSI IDs that are served by the services being discovered

schema:

type: array

items:

type: string

style: form

explode: false

- name: smf-serving-area

in: query

schema:

type: string

- name: tai

in: query

description: Tracking Area Identity

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Tai'

- name: amf-region-id

in: query

description: AMF Region Identity

schema:

type: string

- name: amf-set-id

in: query

description: AMF Set Identity

schema:

type: string

- name: guami

in: query

description: Guami used to search for an appropriate AMF

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Guami'

- name: supi

in: query

description: SUPI of the user

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Supi'

- name: ue-ipv4-address

in: query

description: IPv4 address of the UE

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

- name: ue-ipv6-prefix

in: query

description: IPv6 prefix of the UE

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Prefix'

- name: pgw-ind

in: query

description: Combined PGW-C and SMF or a standalone SMF

schema:

type: boolean

- name: pgw

in: query

description: PGW FQDN of a combined PGW-C and SMF

schema:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

- name: gpsi

in: query

description: GPSI of the user

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Gpsi'

- name: external-group-identity

in: query

description: external group identifier of the user

schema:

type: string

- name: data-set

in: query

description: data set supported by the NF

schema:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/DataSetId'

- name: routing-indicator

in: query

description: routing indicator in SUCI

schema:

type: string

- name: group-id-list

in: query

description: Group IDs of the NFs being discovered

schema:

type: array

items:

type: string

minItems: 1

style: form

explode: false

- name: supported-features

in: query

description: Features required to be supported by the target NF

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

- name: If-None-Match

in: header

description: Validator for conditional requests, as described in IETF RFC 7232, 3.2

schema:

type: string

responses:

'200':

description: Expected response to a valid request

content:

application/json:

schema:

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

headers:

Cache-Control:

description: Cache-Control containing max-age, described in IETF RFC 7234, 5.2

schema:

type: string

ETag:

description: Entity Tag containing a strong validator, described in IETF RFC 7232, 2.3

schema:

type: string

'307':

description: Temporary Redirect

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

components:

securitySchemes:

oAuth2ClientCredentials:

type: oauth2

flows:

clientCredentials:

tokenUrl: '/oauth2/token'

scopes: {}

schemas:

SearchResult:

type: object

properties:

validityPeriod:

type: integer

nfInstances:

type: array

items:

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

NFProfile:

type: object

required:

- nfInstanceId

- nfType

- nfStatus

properties:

nfInstanceId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

nfType:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFType'

nfStatus:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFStatus'

plmn:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/PlmnId'

sNssais:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Snssai'

nsiList:

type: array

items:

type: string

fqdn:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

ipv4Address:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv4Addr'

ipv6Address:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Ipv6Addr'

capacity:

type: integer

minimum: 0

maximum: 65535

load:

type: integer

minimum: 0

maximum: 100

locality:

type: string

priority:

type: integer

minimum: 0

maximum: 65535

udrInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/UdrInfo'

udmInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/UdmInfo'

ausfInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/AusfInfo'

amfInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/AmfInfo'

smfInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/SmfInfo'

upfInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/UpfInfo'

pcfInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/PcfInfo'

bsfInfo:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/BsfInfo'

customInfo:

type: object

recoveryTime:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

nfServices:

type: array

items:

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

NFService:

type: object

required:

- serviceInstanceId

- serviceName

- versions

- scheme

- nfServiceStatus

properties:

serviceInstanceId:

type: string

serviceName:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/ServiceName'

versions:

type: array

items:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFServiceVersion'

scheme:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/UriScheme'

nfServiceStatus:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFServiceStatus'

fqdn:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

ipEndPoints:

type: array

items:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/IpEndPoint'

apiPrefix:

type: string

defaultNotificationSubscriptions:

type: array

items:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/DefaultNotificationSubscription'

capacity:

type: integer

minimum: 0

maximum: 65535

load:

type: integer

minimum: 0

maximum: 100

priority:

type: integer

minimum: 0

maximum: 65535

recoveryTime:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

externalDocs:

description: Documentation

url: 'http://www.3gpp.org/ftp/Specs/archive/29\_series/29.510/'

## A.4 NRF OAuth2 Authorization

openapi: 3.0.0

info:

version: '1.PreR15.1.0'

title: 'NRF OAuth2'

description: 'NRF OAuth2 Authorization'

paths:

/oauth2/token:

post:

summary: Access Token Request

operationId: AccessTokenRequest

tags:

- Access Token Request

security:

- basic: []

requestBody:

content:

application/x-www-form-urlencoded:

schema:

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

required: true

responses:

'200':

description: Successful Access Token Request

content:

application/json:

schema:

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

headers:

Cache-Control:

$ref: '#/components/headers/cache-control'

Pragma:

$ref: '#/components/headers/pragma'

'400':

description: Error in the Access Token Request

content:

application/json:

schema:

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

headers:

Cache-Control:

$ref: '#/components/headers/cache-control'

Pragma:

$ref: '#/components/headers/pragma'

components:

securitySchemes:

basic:

type: http

scheme: basic

headers:

cache-control:

required: true

schema:

type: string

enum:

- no-store

pragma:

required: true

schema:

type: string

enum:

- no-cache

schemas:

AccessTokenReq:

format: x-www-form-urlencoded

required:

- grant\_type

- nfInstanceId

- nfType

- targetNfType

- scope

properties:

grant\_type:

type: string

enum:

- client\_credentials

nfInstanceId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

nfType:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFType'

targetNfType:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/NFType'

scope:

type: string

pattern: '^([a-zA-Z0-9\_]\*[\*]{0,1})$'

AccessTokenRsp:

type: object

required:

- access\_token

- token\_type

properties:

access\_token:

type: string

description: JWS Compact Serialized representation of JWS signed JSON object (AccessTokenClaims)

token\_type:

type: string

expires\_in:

type: integer

scope:

type: string

pattern: '^([a-zA-Z0-9\_]\*[\*]{0,1})$'

AccessTokenClaims:

type: object

required:

- issuer

- subject

- audience

- scope

- expiration

properties:

issuer:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

subject:

type: string

audience:

type: array

items:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/NfInstanceId'

scope:

type: string

pattern: '^([a-zA-Z0-9\_]\*[\*]{0,1})$'

expiration:

type: integer

AccessTokenErr:

type: object

required:

- error

properties:

error:

type: string

enum:

- invalid\_request

- invalid\_client

- invalid\_grant

- unauthorized\_client

- unsupported\_grant\_type

- invalid\_sope

error\_description:

type: string

error\_uri:

type: string

externalDocs:

description: Documentation

url: 'http://www.3gpp.org/ftp/Specs/archive/29\_series/29.510/'

Annex B (informative): Change history

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Meeting** | **TDoc.** | **CR** | **Rev** | **Subject/Comment** | **New** |
| 2017-10 | CT4#80 | C4-175271 |  |  | Initial draft | 0.1.0 |
| 2017-10 | CT4#80 | C4-175395 |  |  | Incorporation of agreed pCRs from CT4#80: C4-175109, C4-175272, C4-175274, C4-175363 | 0.2.0 |
| 2017-12 | CT4#81 | C4-176438 |  |  | Incorporation of agreed pCRs from CT4#81: C4-176184, C4-176278, C4-176280, C4-176281, C4-176282 | 0.3.0 |
| 2018-01 | CT4#82 | C4-181392 |  |  | Incorporation of agreed pCRs from CT4#82: C4-181348, C4-181351 | 0.4.0 |
| 2018-03 | CT4#83 | C4-182435 |  |  | Incorporation of agreed pCRs from CT4#83: C4-182098, C4-182327, C4-182328, C4-182365, C4-182413 | 0.5.0 |
| 2018-04 | CT4#84 | C4-183517 |  |  | Incorporation of agreed pCRs from CT4#84: C4-183450, C4-183451, C4-183452, C4-183487, C4-183488, C4-183490, C4-183491 | 0.6.0 |
| 2018-05 | CT4#85 | C4-184625 |  |  | Incorporation of agreed pCRs from CT4#85: C4-184207,  C4-184208, C4-184280, C4-184466, C4-184469, C4-184478, C4-184517, C4-184519, C4-184545, C4-184595, C4-184596, C4-184597, C4-184600, C4-184615, C4-184616, C4-184626 | 0.7.0 |
| 2018-06 | CT#80 | CP-181105 |  |  | Presented for information and approval | 1.0.0 |
| 2018-06 | CT#80 |  |  |  | Approved in CT#80. | 15.0.0 |
| 2018-09 | CT#81 | CP-182012 | 0001 | 2 | Implementing the Indirect Delivery method for the GET method to retrieve NF instances | 15.1.0 |
| 2018-09 | CT#81 | CP-182093 | 0003 | 3 | Defining the range of the priority and capacity attributes and aligning their usage with SRV RFC 2782 | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0004 | - | Corrections to descriptions, references and SUPI parameter in Discovery Request | 15.1.0 |
| 2018-09 | CT#81 | CP-182047 | 0006 | 2 | SubscriptionData | 15.1.0 |
| 2018-09 | CT#81 | CP-182045 | 0008 | 2 | Error Cases | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0009 | 2 | Heart Beat Procedure | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0010 | 1 | Vendor-Specific NF Types | 15.1.0 |
| 2018-09 | CT#81 | CP-182044 | 0011 | 3 | Presence condition of service discovery query parameters | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0012 | 4 | Description of Inter-PLMN scenarios | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0013 | 1 | NF Service Versions | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0014 | 1 | Custom Headers | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0015 | 1 | Overall Clean-up | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0016 | - | Formatting of query parameters | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0017 | - | Editorial corrections | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0018 | 2 | Backup AMF | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0020 | 1 | NF Service Names | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0023 | - | CHF as service consumer | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0024 | 3 | Hierarchical NF discovery in recursion mode | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0025 | 2 | Hierarchical NF discovery in iteration mode | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0026 | - | Correction of Allowed NF Domains | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0027 | - | Correction of BsfInfo data type | 15.1.0 |
| 2018-09 | CT#81 | CP-182161 | 0028 | 1 | IPv6 Prefix for NF / NF Service Address | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0030 | 1 | NF Set Id | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0031 | 1 | URI Scheme | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0032 | 2 | NRF service registration | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0034 | 2 | Discovery of combined SMF and PGW-C | 15.1.0 |
| 2018-09 | CT#81 | CP-182163 | 0035 | 3 | Support TAI Range for AMF/SMF and SUPI Range for PCF | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0036 | 1 | SUPI Range for PCF | 15.1.0 |
| 2018-09 | CT#81 | CP-182164 | 0037 | 2 | Scope for OAuth 2.0 Access Token Request | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0039 | 1 | Corrections to NotificationData and "supi" parameter in Discovery Request | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0040 | 1 | Group ID in Discovery Request | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0041 | 1 | Registering multiple Routing Indicators | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0045 | - | Description of Structured data types | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0046 | 1 | Service names in Discovery Request | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0047 | 1 | Resource structure presentation | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0048 | - | Default Notifications for UDM | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0049 | - | Cell ID in Discovery Request | 15.1.0 |
| 2018-09 | CT#81 | CP-182046 | 0050 | 2 | NRF Subscription Data | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0051 | 1 | AMF Discovery by 5G-AN | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0052 | 1 | Detecting NF Failure and Restart using the NRF | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0053 | 2 | NRF Subscription Lifespan | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0054 | 1 | NRF servers section in OpenAPI | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0056 | 2 | Default port number | 15.1.0 |
| 2018-09 | CT#81 | CP-182162 | 0057 | 1 | AMF Discovery Based on AMF Name | 15.1.0 |
| 2018-09 | CT#81 | CP-182060 | 0058 | - | API Version Update | 15.1.0 |