ETSI TS 129 598 V17.5.0 (2022-05)

5G;
Unstructured data storage services
(3GPP TS 29.598 version 17.5.0 Release 17)

Reference
RTS/TSGC-0429598vh50

Keywords
5G

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

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

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

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

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

Contents

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.

In the present document, modal verbs have the following meanings:

shall indicates a mandatory requirement to do something

shall not indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

should indicates a recommendation to do something

should not indicates a recommendation not to do something

may indicates permission to do something

need not indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

can indicates that something is possible

cannot indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

will indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

will not indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

might indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

1 Scope

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

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] OpenAPI: "OpenAPI Specification Version 3.0.0", https://spec.openapis.org/oas/v3.0.0.

[7] 3GPP TR 21.900: "Technical Specification Group working methods".

[8] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

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

[10] 3GPP TS 29.510: "5G System; Network Function Repository Services; Stage 3".

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

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

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

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

[15] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

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

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

[18] ISO/IEC 14977: "Information technology – Syntactic metalanguage - Extended BNF".

[19] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces; Stage 3".

[20] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

[21] IETF RFC 2046: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types".

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

3 Definitions of terms, symbols and abbreviations

3.1 Terms

void

3.2 Symbols

void

3.3 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
BNF	Backus–Naur Form
EBNF	Extended BNF
CP	Control Plane
MIME	Multipurpose Internet Mail Extensions
NF	Network Function
UDSF	Unstructured Data Storage Function

4 Overview

The UDSF, as depicted in Figure 4.1-1 below, is described in clause 4.2.5 of 3GPP TS 23.501 [2]. Any of the 5GS NFs can make use of the UDSF to store and retrieve unstructured data, i.e., data that is not defined in 3GPP specifications, and can make use of the UDSF to run timers and get notified on timer expiry. The UDSF is deployed in the same network where the CP NF is located and the same UDSF may be shared by all the NFs in the PLMN to store/retrieve their respective data or an NF may have its own UDSF depending on operator configuration.

NOTE 1: Structured data in this specification refers to data for which the structure is defined in 3GPP specifications. Unstructured data refers to data for which the structure is not defined in 3GPP specifications.

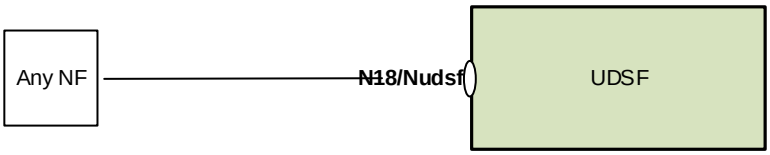

Figure 4.1-1: Reference model – UDSF

5 Services offered by the UDSF

5.1 Introduction

The UDSF offers the following services via the Nudsf service based interface:

- Nudsf_DataRepository Service

NOTE: This service corresponds to the Nudsf_UnstructuredDataManagement service in 3GPP TS 23.501 [2] and 3GPP TS 23.502 [3].

- Nudsf_Timer Service

Editor's Note: Details of the Nudsf_TimerService need to be further reviewed.

Table 5.1-1 summarizes the corresponding APIs defined for this specification.

Table 5.1-1: API Descriptions

Service Name	Clause	Description	OpenAPI Specification File	apiName	Annex
Nudsf_DataRepository	6.1	UDSF Data Repository Service	TS29598_Nudsf_DataRepository.yaml	nudsf-dr	A.2
Nudsf_Timer	6.2	UDSF Timer Service	TS29598_Nudsf_Timer.yaml	nudsf-timer	A.3

5.2 Nudsf_DataRepository Service

5.2.1 Service Description

The UDSF is acting as an NF Service Producer. It provides UDSF data repository service to the NF service consumer. Any NF may use the UDSF to store unstructured data.

NOTE 1: Structured data in this specification refers to data for which the structure is defined in 3GPP specifications. Unstructured data refers to data for which the structure is not defined in 3GPP specifications.

5.2.2 Service Operations

5.2.2.1 Introduction

For the Nudsf_DataRepository service, the following service operations are defined:

- Query
- Create
- Update
- Delete
- Notify
- Subscribe
- Unsubscribe

5.2.2.2 Query

5.2.2.2.1 General

The following procedures using the Query service operation are supported:

- Record Retrieval
- Meta Retrieval
- Blocks Retrieval
- Block Retrieval
- Search
- Subscriptions Retrieval
- Individual Subscription Retrieval
- Meta Schema Retrieval

5.2.2.2.2 Record Retrieval

Figure 5.2.2.2.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve a record that matches the provided recordId and optionally includes the query parameter supported-features.

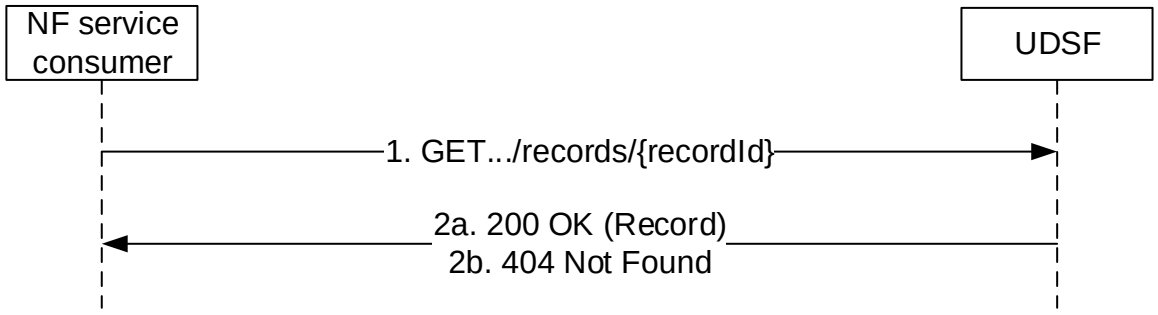

Figure 5.2.2.2.2-1: Requesting a Record

1. The NF service consumer (any NF) sends a GET request to the resource indicated by recordId.

2a. On success, the UDSF responds with "200 OK" with the message body containing the record.

2b. If the record for the given recordId does not exist in the UDSF, the HTTP status code "404 Not Found" shall be returned optionally including additional error information in the response body (in the ProblemDetails element).

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.3 Meta Retrieval

Figure 5.2.2.2.3-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve meta data associated with the provided recordId and optionally includes the query parameter supported-features.

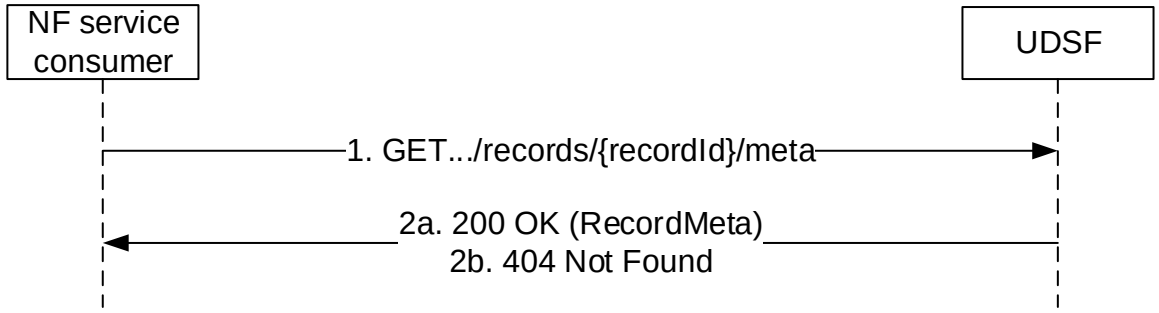

Figure 5.2.2.2.3-1: Requesting Meta for a Record

1. The NF service consumer (any NF) sends a GET request to the meta resource associated with the record indicated by recordId.

2a. On success, the UDSF responds with "200 OK" with the message body containing the RecordMeta.

2b. If the record for the given recordId and thus the RecordMeta does not exist in the UDSF, the HTTP status code "404 Not Found" shall be returned optionally including additional error information in the response body (in the ProblemDetails element).

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.4 Blocks Retrieval

Figure 5.2.2.2.4-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve (all) the blocks associated with the provided recordId and optionally includes the query parameter supported-features.

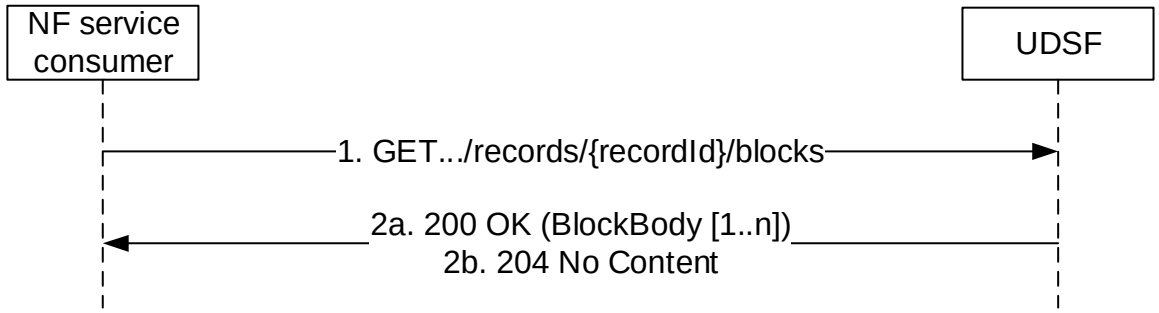

Figure 5.2.2.2.4-1: Requesting Blocks

1. The NF service consumer (any NF) sends a GET request to the resource indicated by recordId.

2a. On success, the UDSF responds with "200 OK" with the message body containing the Blocks associated with the record.

2b. If a Block for the given recordId does not exist in the UDSF, the HTTP status code "204 No Content" shall be returned.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.5 Block Retrieval

Figure 5.2.2.2.5-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve a single block associated with the provided recordId and blockId and optionally includes the query parameter supported-features.

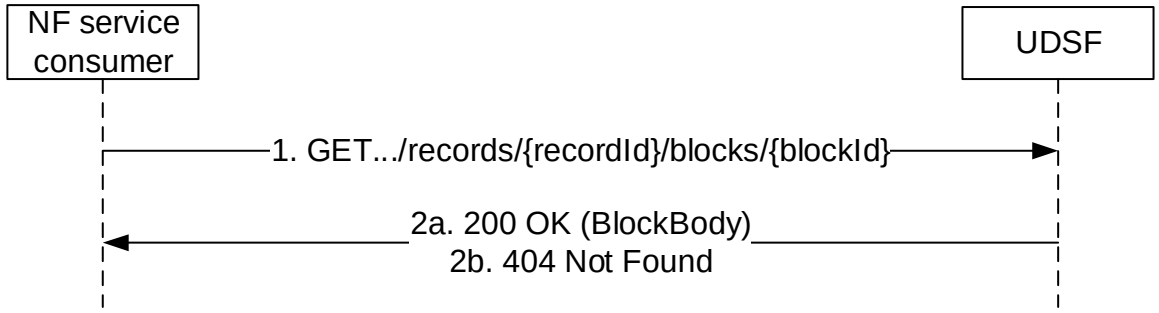

Figure 5.2.2.2.5-1: Requesting a Block

1. The NF service consumer (any NF) sends a GET request to the resource indicated by recordId and blockId.

2a. On success, the UDSF responds with "200 OK" with the message body containing the Block associated with the blockId.

2b. If the Block for the given recordId and blockId does not exist in the UDSF, the HTTP status code "404 Not Found" shall be returned optionally including additional error information in the response body (in the ProblemDetails element).

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.6 Search

Figure 5.2.2.2.6-1 shows a scenario where the NF service consumer sends a request to the UDSF to search a record that matches the provided search criteria.

The request contains the query parameters filter and optionally supported-features, limit-range, and count-indicator, and if the CombinedSearchRetrieve feature is supported may contain the query parameters retrieve-records and max-payload-size. If the BulkOperations feature is supported, the query parameter filter may contain a list of record IDs identifying the records to be retrieved.

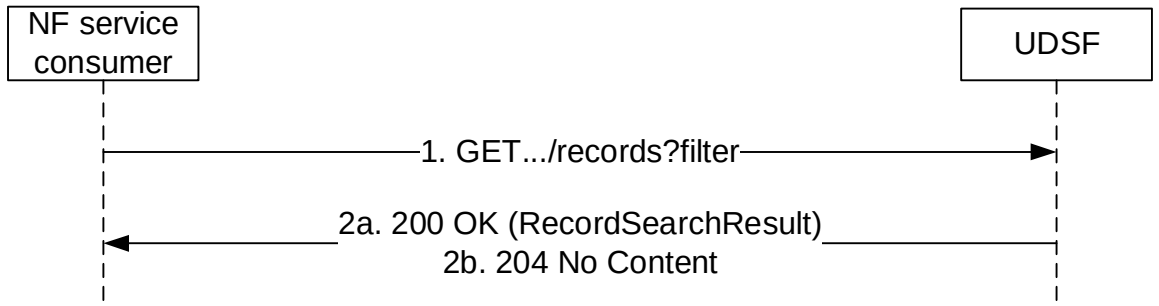

Figure 5.2.2.2.6-1: Searching for Records

1. The NF service consumer (any NF) sends a GET request to the Records resource with the filter query parameter indicating the search criteria.

2a. On success, the UDSF responds with "200 OK" with the message body containing the RecordSearchResult.

2b. If the UDSF is not able to return any record for the given search criteria, the HTTP status code "204 No Content" shall be returned.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.7 Subscriptions Retrieval

Figure 5.2.2.2.7-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve all subscriptions associated with the provided storageId and optionally includes the query parameter supported-features and limit-range.

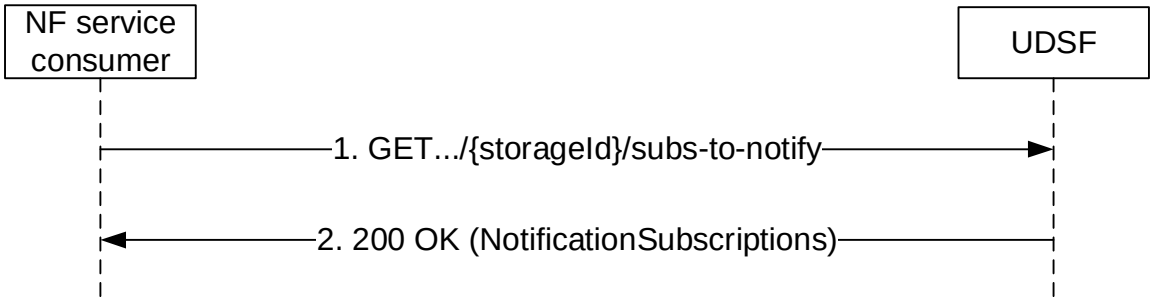

Figure 5.2.2.2.7-1: Requesting Subscriptions

1. The NF service consumer (any NF) sends a GET request to the resource indicated by the storageId.
2. On success, the UDSF responds with "200 OK" with the message body containing the NotificationSubscriptions associated with the storageId (if any).

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.8 Individual Subscription Retrieval

Figure 5.2.2.2.8-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve a subscription associated with the provided storageId and subscriptionId.

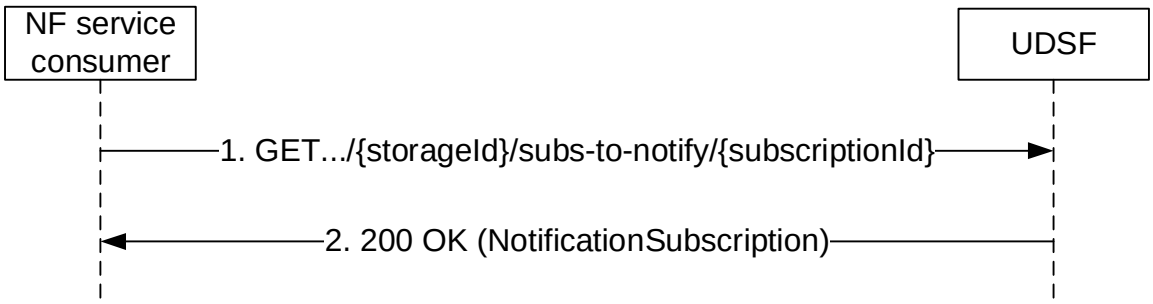

Figure 5.2.2.2.8-1: Requesting an Individual Subscription

1. The NF service consumer (any NF) sends a GET request to the resource indicated by the storageId and the subscriptionId.

2a. On success, the UDSF responds with "200 OK" with the message body containing the NotificationSubscription.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.2.9 Meta Schema Retrieval

Figure 5.2.2.2.9-1 shows a scenario where the NF service consumer sends a request to the UDSF to retrieve a meta schema that matches the provided schemaId and optionally includes the query parameter supported-features.

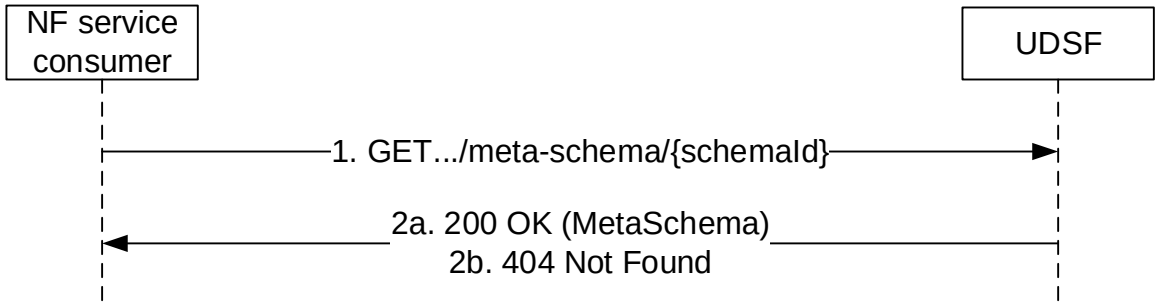

Figure 5.2.2.2.9-1: Requesting a Meta Schema

1. The NF service consumer (any NF) sends a GET request to the resource indicated by schemaId.

2a. On success, the UDSF responds with "200 OK" with the message body containing the meta schema.

2b. If the meta schema for the given schemaId does not exist in the UDSF, the HTTP status code "404 Not Found" shall be returned optionally including additional error information in the response body (in the ProblemDetails element).

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.2.2.3 Create

5.2.2.3.1 General

The following procedures using the Create service operation are supported:

- Record Create
- Block Create
- Meta Schema Create

5.2.2.3.2 Record Create

Figure 5.2.2.3.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to create a record with the provided recordId.

The request contains the recordId and optionally the query parameters supported-features and get-previous.

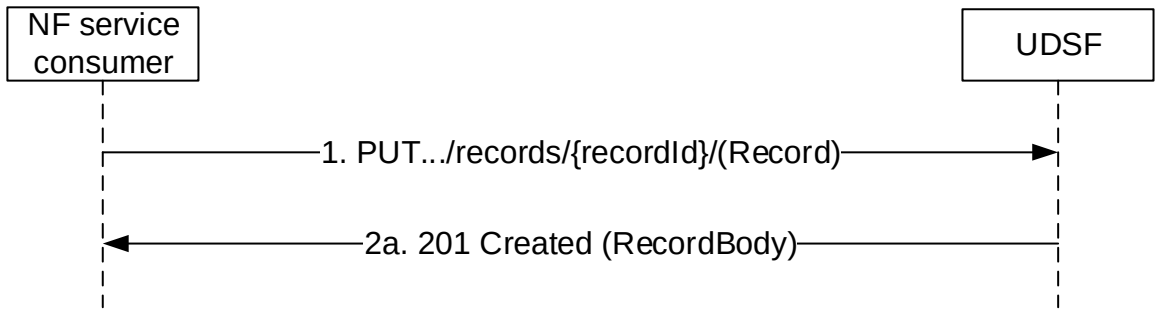

Figure 5.2.2.3.2-1: Create a Record

1. The NF service consumer (any NF) sends a PUT request to create the resource indicated by recordId. The request body contains the meta, zero or more blocks. The record meta information is mandatory and shall be the first part and the remaining parts of the request body (if any) shall be child blocks. If the record meta information is received with record expiry details, UDSF shall create an implicit subscription locally and notify the NF service consumer on record expiry.

2a. On success, "201 Created" shall be returned, the payload body of the PUT response should contain the representation of the created resource, and the "Location" header shall be present and shall contain the URI of the created resource.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.3.3 Block Create

Figure 5.2.2.3.3-1 shows a scenario where the NF service consumer sends a request to the UDSF to create a block with the provided blockId.

The request contains the blockid and optionally the query parameters supported-features and get-previous.

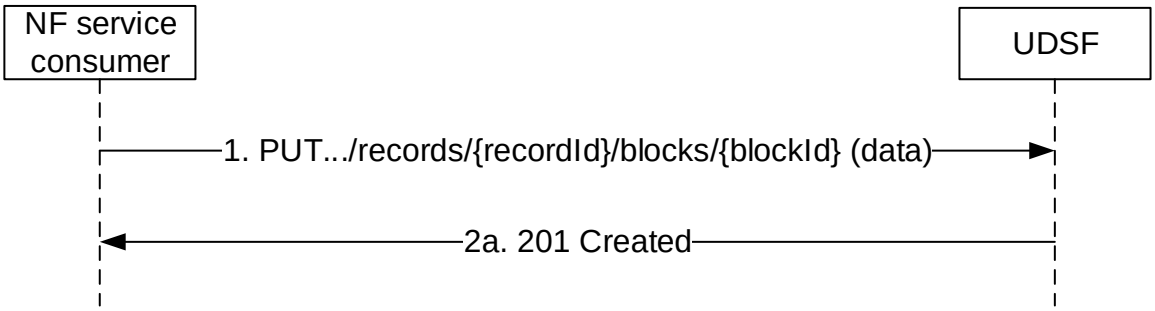

Figure 5.2.2.3.3-1: Create a Block

1. The NF service consumer (any NF) sends a PUT request to create the resource indicated by blockId.

2a. On success, "201 Created" shall be returned, the payload body of the PUT response should contain the representation of the created resource, and the "Location" header shall be present and shall contain the URI of the created resource.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.3.4 Meta Schema Create

Figure 5.2.2.3.4-1 shows a scenario where the NF service consumer sends a request to the UDSF to create a meta schema with the provided schemaId.

The request contains the schemaId and optionally the query parameters supported-features and get-previous.

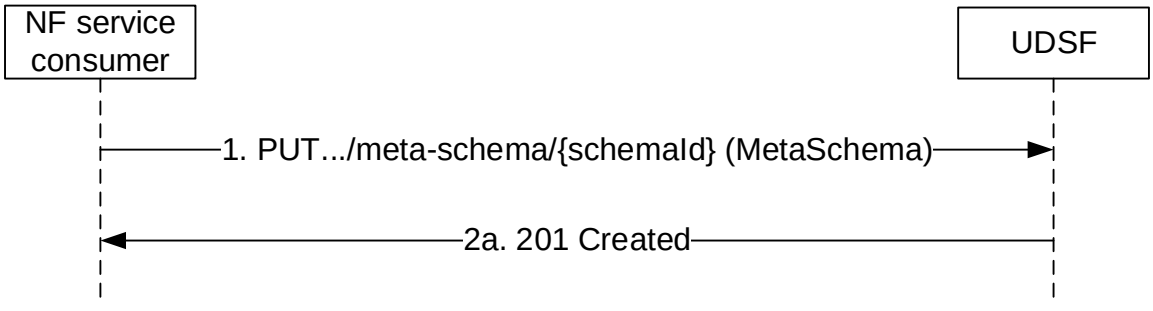

Figure 5.2.2.3.4-1: Create a Meta Schema

1. The NF service consumer (any NF) sends a PUT request to create the resource indicated by schemaId. The request body contains the meta schema.

2a. On success, "201 Created" shall be returned, the payload body of the PUT response should contain the representation of the created resource, and the "Location" header shall be present and shall contain the URI of the created resource.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.4 Update

5.2.2.4.1 General

The following procedures using the Update service operation are supported:

- Record Update
- Block Update
- Meta Update
- Subscription Notification Update
- Meta Schema Update

5.2.2.4.2 Record Update

Figure 5.2.2.4.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to update a record with the provided recordId.

The request contains the recordId and optionally the query parameters supported-features and get-previous.

The update shall include meta, zero or more blocks. The record meta information shall be the first part and is mandatory and the remaining parts of the body (if any) shall be interpreted as child blocks. Existing record, meta and blocks shall be discarded and the new record, meta and blocks (if any) shall be created.

NOTE: The order of the returned blocks in the response is not guaranteed and can be different from the order used to create them.

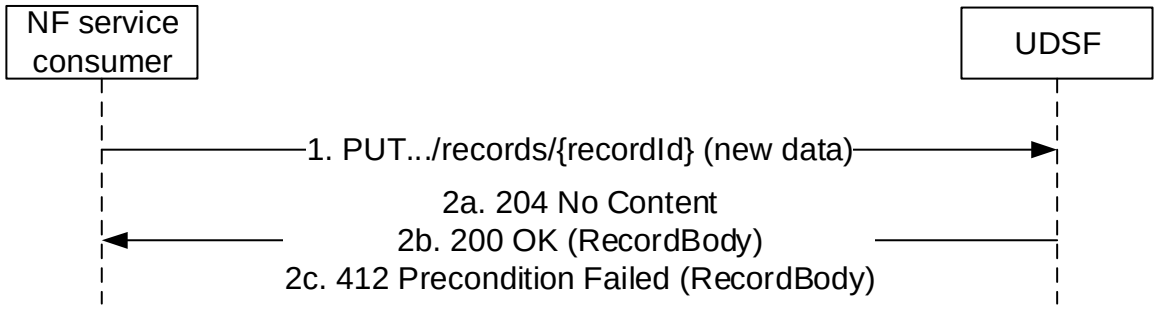

Figure 5.2.2.4.2-1: Update a record

1. The NF service consumer shall send a PUT request to the resource representing the record that is to be updated, and may include meta, zero or more blocks. The record meta information is mandatory and shall be the first part and the remaining parts of the request body (if any) shall be child blocks. An existing record, i.e., meta and blocks shall be discarded and the new record, meta and blocks (if any) shall be created.

2a. On success, the UDSF shall respond with "204 No Content" if no record is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" if a record is returned, i.e. the get-previous query parameter was included in the request, or due to operator's policy, the ttl value in the request exceeded the maximum value allowed.

2c. On failure, the UDSF shall respond with "412 Precondition Failed" if one or more conditions given in the request header fields evaluated to false. The RecordBody shall include the stored Record if the get-previous query parameter was included in the request.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.4.3 Block Update

Figure 5.2.2.4.3-1 shows a scenario where the NF service consumer sends a request to the UDSF to update a block with the provided blockId.

The request contains the recordId, blockId and the optional query parameters supported-features, get-previous and the data that is to be updated.

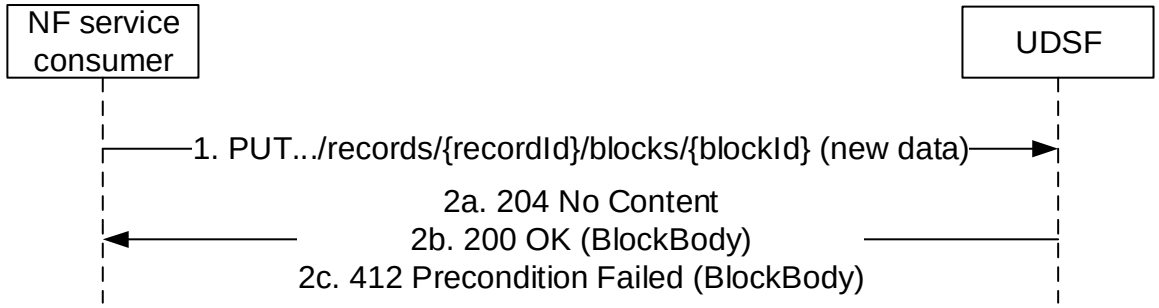

Figure 5.2.2.4.3-1: Update a block

1. The NF service consumer shall send a PUT request to the resource representing the block that is to be updated.

2a. On success, the UDSF shall respond with "204 No Content" if no record is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" if a record is returned, i.e. the get-previous query parameter was included in the request.

2c. On failure, the UDSF shall respond with "412 Precondition Failed" if one or more conditions given in the request header fields evaluated to false. The BlockBody shall be included with the stored Block if the get-previous query parameter was included in the request.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.4.4 Meta Update

Figure 5.2.2.4.4-1 shows a scenario where the NF service consumer sends a request to the UDSF to update the meta data associated with the provided recordId and optionally the query parameter supported-features and the data that is to be updated.

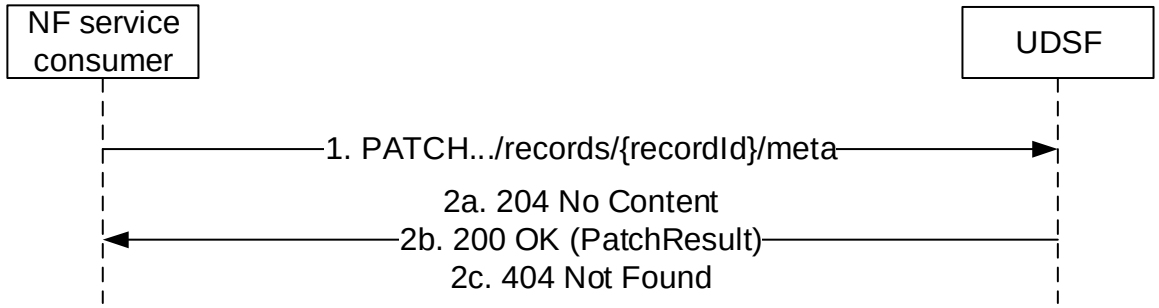

Figure 5.2.2.4.4-1: Update meta

1. The NF service consumer shall send a PATCH request to the resource representing the meta of the record.

2a. On success, if all the modification instructions in the PATCH request have been implemented, the UDSF shall respond with "204 No Content".

2b. On partial success, i.e. if one or more modification instructions have been discarded, "200 OK" with the execution report, shall be returned.

2c. On failure, the UDSF shall respond with "404 Not Found" if the record indicated by the recordId and thus the meta does not exist and may shall include the ProblemDetails.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.4.5 Subscription Notification Update

Figure 5.2.2.4.5-1 shows a scenario where the NF service consumer sends a request to the UDSF to update the Individual Subscription Notification identified with the storageId and subscriptionId and optionally the query parameter supported-features and the data that is to be updated.

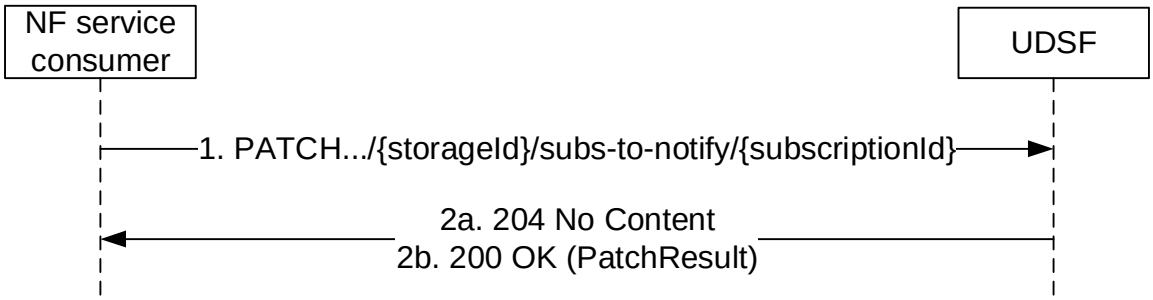

Figure 5.2.2.4.5-1: Update Subscription Notification

1. The NF service consumer shall send a PATCH request to the resource representing the subscriptionId.

2a. On success, if all the modification instructions in the PATCH request have been implemented, the UDSF shall respond with "204 No Content".

2b. On partial success, i.e. if one or more modification instructions have been discarded, "200 OK" with the execution report, shall be returned.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PATCH response body.

5.2.2.4.6 Subscription Notification Update using PUT

Figure 5.2.2.4.6-1 shows a scenario where the NF service consumer sends a request to the UDSF to update a subscription to notifications of data change using PUT. The request contains the subscriptionId and the NotificationSubscription and optionally the query parameter supported-features.

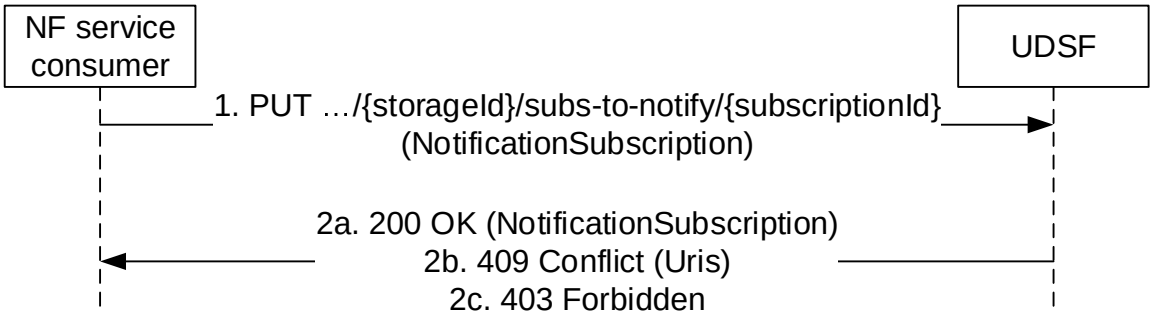

Figure 5.2.2.4.6-1: NF service consumer updates subscription to notifications

1. The NF service consumer sends a PUT request to the resource indicated by the storageId and the subscriptionId. The parameter clientId shall be included. If the resource indicated in URI exists and was created by the Client identified by the clientId, the UDSF shall apply the update of the subscription.

2a. On success, the UDSF shall respond with "200 OK" and include the updated NotificationSubscription. The expiry attribute of the received NotificationSubscription may indicate a value or a value different from the request, if due to an operator policy, an expiry time is enforced or if the value in the request exceeded a maximum allowed expiry time.

2b. On failure, if one or more monitoredResourceUris from the request don't exist in the UDSF, 409 Conflict shall be returned together with the non-existing monitoredResourceUris.

2c. On failure, if the service operation cannot be authorized due to e.g. the resource indicated in URI exists but the clientId in the PUT request does not match the clientId of the existing resource, the UDSF shall respond with "403 Forbidden" and optionally including additional error information in the response body (in "ProblemDetails" element).

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.4.7 Meta Schema Update

Figure 5.2.2.4.7-1 shows a scenario where the NF service consumer sends a request to the UDSF to update a meta schema with the provided schemaId.

The request contains the schemaId, the complete new meta schema and optionally the query parameters supported-features and get-previous.

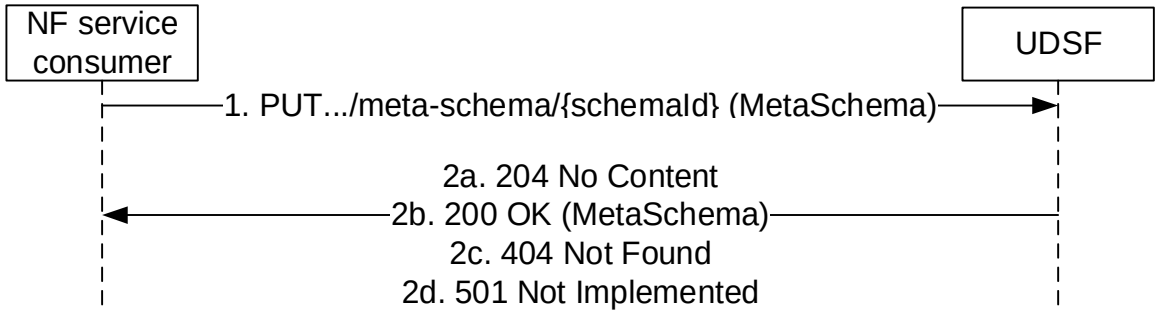

Figure 5.2.2.4.7-1: Update a meta schema

1. The NF service consumer shall send a PUT request to the resource representing the record that is to be updated, the request body shall include the complete new meta schema.

2a. On success, the UDSF shall respond with "204 No Content" if no meta schema is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" if a meta schema is returned, i.e. the get-previous query parameter was included in the request.

2c. On failure, the UDSF shall respond with "404 Not Found" if realm or storage does not exist.

2d On failure, the UDSF shall respond with "501 Not Implemented" if meta schema update is not implemented by the UDSF.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.5 Delete

5.2.2.5.1 General

The following procedures using the Delete service operation are supported:

- Record Delete
- Block Delete
- Meta Schema Delete
- Bulk Records Delete

5.2.2.5.2 Record Delete

Figure 5.2.2.5.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to Delete a record with the provided recordId.

The request contains the record id and optionally the query parameters supported-features and get-previous.

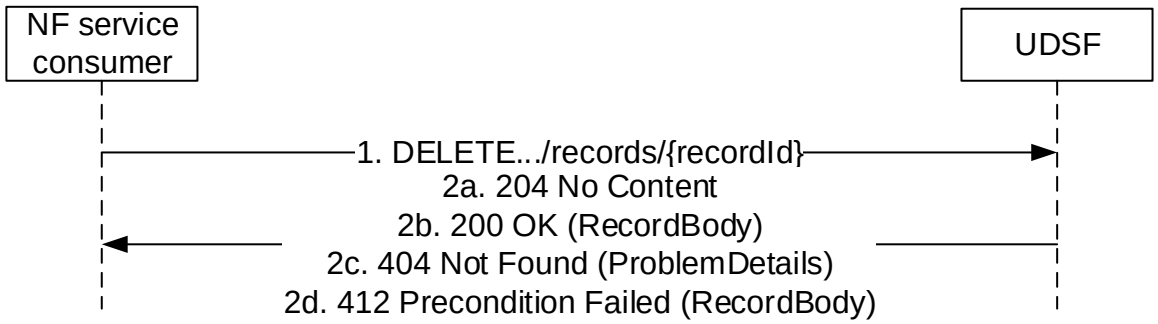

Figure 5.2.2.5.2-1: Delete a record

1. The NF service consumer shall send a DELETE request to the resource representing the record. The UDSF shall delete any resource associated with the resource (meta and block(s)).

2a. On success, the UDSF shall respond with "204 No Content" if no record is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" if a record is returned, i.e. the get-previous query parameter was included in the request.

2c. On failure, the UDSF shall respond with "404 Not Found" if the record does not exist and may include the ProblemDetails.

2d. On failure, the UDSF shall respond with "412 Precondition Failed" if one or more conditions given in the request header fields evaluated to false and the get-previous query parameter was included in the request. The RecordBody shall be included if the get-previous query parameter was included in the request.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the DELETE response body.

5.2.2.5.3 Block Delete

Figure 5.2.2.5.3-1 shows a scenario where the NF service consumer sends a request to the UDSF to Delete a block with the provided blockId.

The request contains the recordId, blockId and optionally the query parameters supported-features and get-previous.

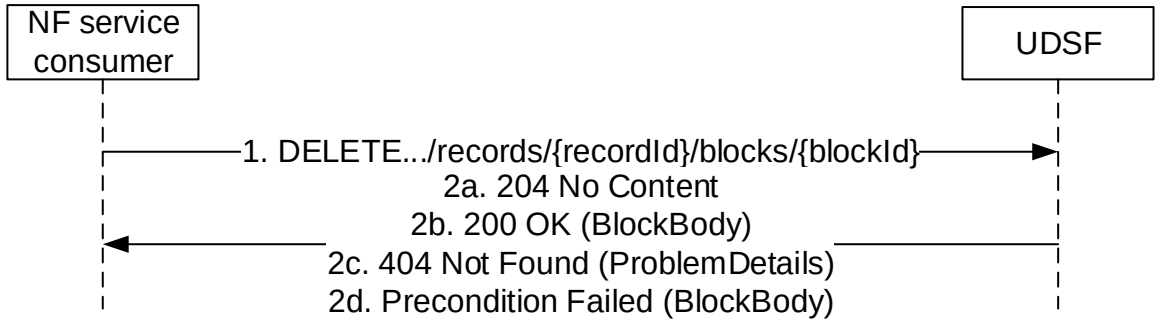

Figure 5.2.2.5.3-1: Delete a block

1. The NF service consumer shall send a DELETE request to the resource representing the block.

2a. On success, the UDSF shall respond with "204 No Content" if no block is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" if a block is returned, i.e. the get-previous query parameter was included in the request.

2c. On failure, the UDSF shall respond with "404 Not Found" if the block does not exist and may include the ProblemDetails.

2d. On failure, the UDSF shall respond with "412 Precondition Failed" if one or more conditions given in the request header fields evaluated to false. The BlockBody shall be included if the get-previous query parameter was included in the request.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the DELETE response body.

5.2.2.5.4 Meta Schema Delete

Figure 5.2.2.5.4-1 shows a scenario where the NF service consumer sends a request to the UDSF to Delete a meta schema with the provided schemaId.

The request contains the schemaId and optionally the query parameters supported-features and get-previous.

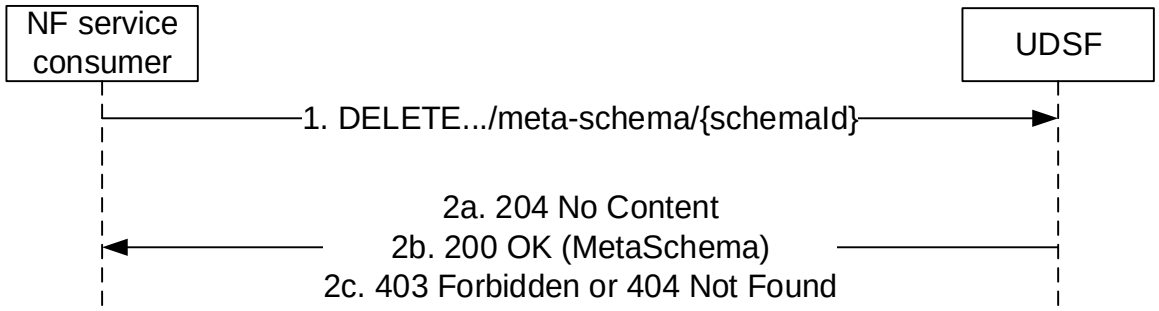

Figure 5.2.2.5.4-1: Delete a meta schema

1. The NF service consumer shall send a DELETE request to the resource representing the meta schema.

2a. On success, the UDSF shall respond with "204 No Content" if no meta schema is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" if a meta schema is returned, i.e. the get-previous query parameter was included in the request.

2c. On failure, the UDSF shall respond with "404 Not Found" if the meta schema does not exist or "403 Forbidden" if the meta schema is still referenced by existing records, and may include the ProblemDetails.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the DELETE response body.

5.2.2.5.5 Bulk Records Delete

Figure 5.2.2.5.5-1 shows a scenario where the NF service consumer sends a request to the UDSF to Delete records identified by a filter.

The request contains the query parameter filter and optionally the query parameter supported-features.

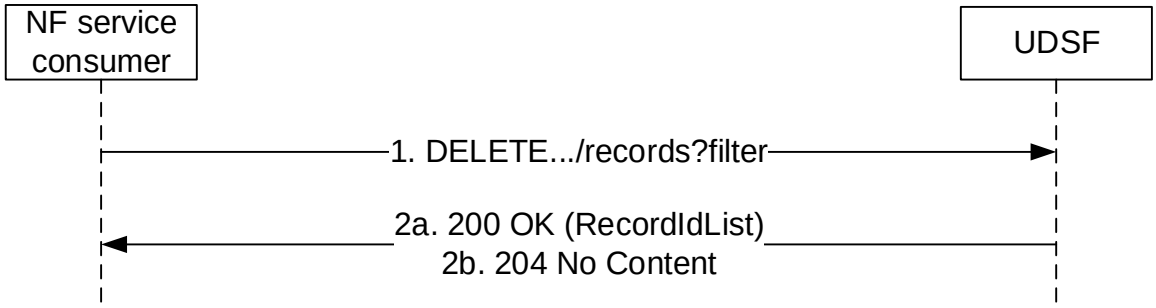

Figure 5.2.2.5.5-1: Delete records matching a filter

1. The NF service consumer shall send a DELETE request to the resource representing the records. The request shall contain a filter query parameter identifying the records to be deleted. The UDSF shall delete any resource associated with the matching records (meta and block(s)).

2a. On success, the UDSF shall respond with "200 OK" with the message body containing the RecordIdList identifying the deleted records.

2b. The UDSF shall respond with "204 No Content" if no records matching the filter could be found.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the DELETE response body.

5.2.2.6 Notify

5.2.2.6.1 General

The following procedures using the Notify service operation are supported:

- Record Expiry Notify
- Notification due to Data Change
- Subscription Expiry Notification

5.2.2.6.2 Record Expiry Notify

Figure5.2.2.6.2-1 shows a scenario where the UDSF notifies the NF service consumer of the expired record.

The Notify is sent by the UDSF to the NF Service Consumer when the record expires as indicated by the time to live (ttl) attribute of RecordMeta.

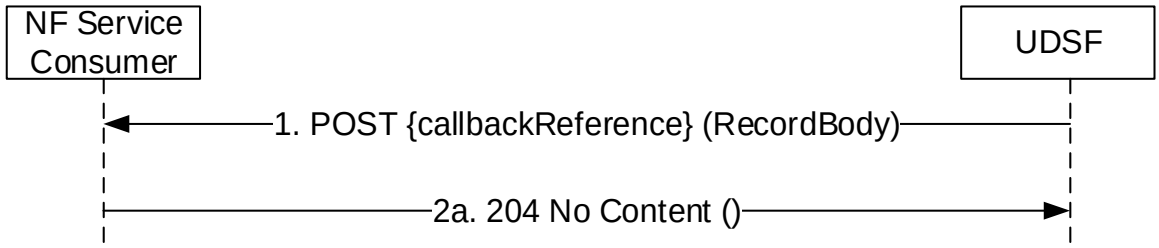

Figure 5.2.2.6.2-1: Record Expiry Notify

1. The UDSF shall send a POST request to the callback URI. The request shall contain the record details.

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

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the POST response body.

5.2.2.6.3 Notification due to Data Change

Figure 5.2.2.6.3-1 shows a scenario where the UDSF notifies the NF service consumer of a change to data associated with a block or a record triggered by one or more a Subscription to Notification.

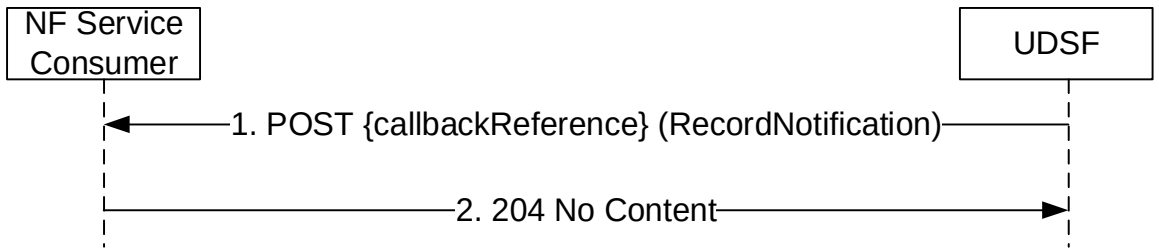

Figure 5.2.2.6.3-1: Notification due to Data Change

1. The UDSF shall send a POST request to the callback URI. The request shall contain the RecordNotification details.
2. On success, "204 No content" shall be returned by the NF Service Consumer to UDSF.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the POST response body.

5.2.2.6.4 Subscription Expiry Notification

Figure 5.2.2.6.4-1 shows a scenario where the UDSF notifies the NF service consumer of an expired subscription to data change.

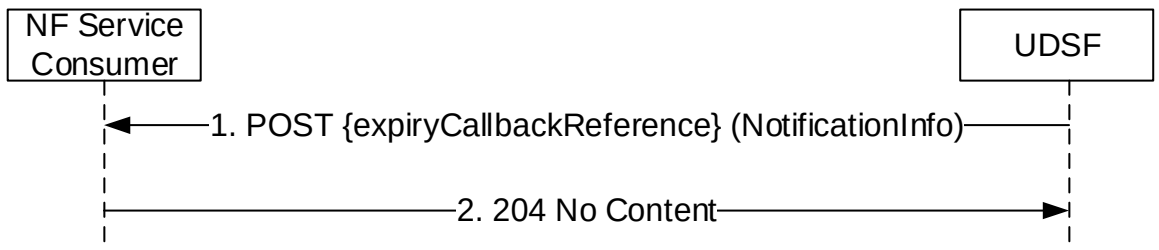

Figure 5.2.2.6.4-1: Subscription Expiry Notification

1. The UDSF shall send a POST request to the expiryCallbackReference URI. The request shall contain the NotificationInfo.

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

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the POST response body.

5.2.2.7 Subscribe

5.2.2.7.1 General

The following procedures using the Subscribe service operation are supported:

- Subscription to notification of data change

5.2.2.7.2 Subscription to notifications of data change

Figure 5.2.2.7.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to subscribe to notifications of data change. The request contains the subscriptionId, the NotificationSubscription and optionally the query parameter supported-features.

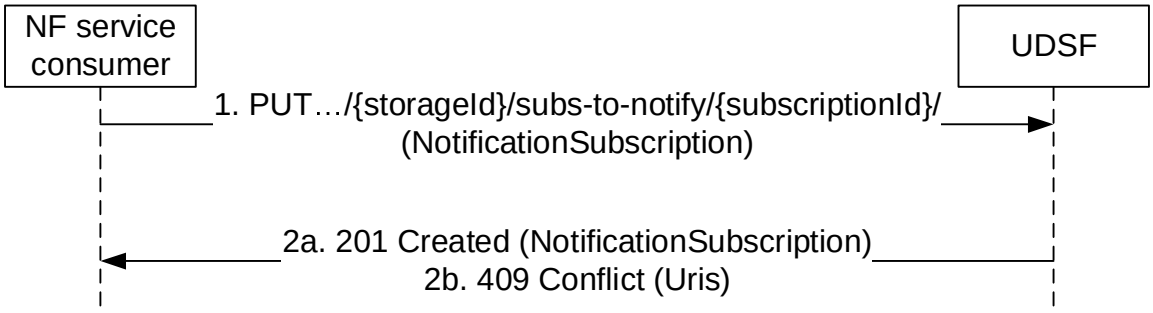

Figure 5.2.2.7.2-1: NF service consumer subscribes to notifications

1. The NF service consumer sends a PUT request to the resource indicated by the storageId and the subscriptionId. The parameter clientId shall be included. If the resource indicated in URI doesn't exist, the UDSF shall trigger the creation of the subscription.

2a. On success, the UDSF responds with "201 Created"with the message body containing the NotificationSubscription. The expiry attribute of the received NotificationSubscription may indicate a value or a value different from the request, if due to an operator policy, an expiry time is enforced or if the value in the request exceeded a maximum allowed expiry time.

2b. On failure, if one or more monitoredResourceUris from the request don't exist in the UDSF, 409 Conflict shall be returned together with the non-existing monitoredResourceUris.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.2.2.8 Unsubscribe

5.2.2.8.1 General

The following procedures using the Unsubscribe service operation are supported:

- Unsubscription to notification of data change

5.2.2.8.2 Unsubscription to notifications of data change

Figure 5.2.2.8.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to Unsubscribe to notifications of data change. The request contains the subscriptionId, and query parameter client-id, optionally query parameters supported-features and get-previous.

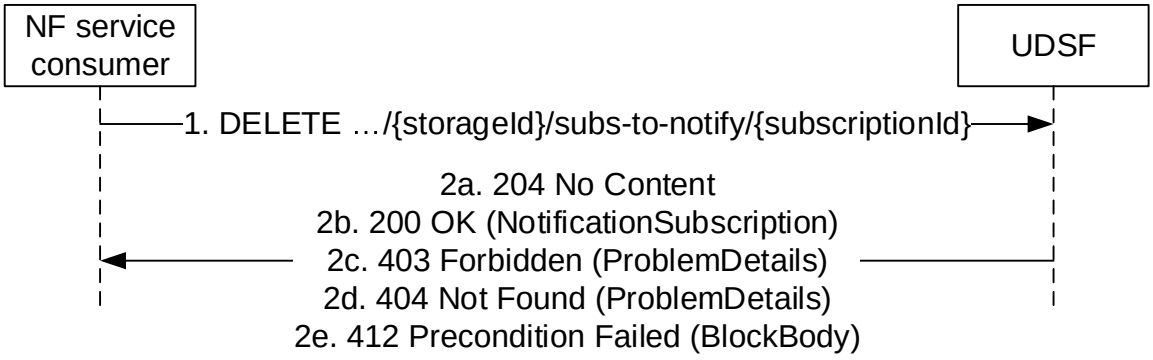

Figure 5.2.2.8.2-1: NF service consumer unsubscribes to notifications

1. The NF service consumer sends a DELETE request to the resource representing the subscription to notification of data change which is indicated by the subscriptionId.

2a. On success, the UDSF shall respond with "204 No Content" if an empty response body is returned, i.e. the get-previous query parameter was not included in the request.

2b. On success, the UDSF shall respond with "200 OK" with NotificationSubscription containing the NotificationSubscription value before the delete if get-previous was indicated in the request.

2c. If the service operation cannot be authorized due to e.g. the client-id query parameter does not match the clientId of the existing resource, the UDSF shall respond with "403 Forbidden" with including additional error information in the response body (in "ProblemDetails" element).

2d. If there is no valid subscription to notification of data change which is indicated by the request, the UDSF shall respond with "404 Not Found" with including additional error information in the response body (in "ProblemDetails" element).

2e. If one or more conditions given in the request header fields evaluated to false, the UDSF shall respond with "412 Precondition Failed". NotificationSubscription shall be included in the response if get-previous was indicated in the request.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.3 Nudsf_Timer Service

5.3.1 Service Description

The UDSF is acting as an NF Service Producer. It provides UDSF timer service to the NF service consumer. Any NF may use the UDSF to run timers, search for timers, and get notifications on expiry.

5.3.2 Service Operations

5.3.2.1 Introduction

For the Nudsf_Timer service, the following service operations are defined:

- Start

- Update
- Stop
- Search
- Notify

5.3.2.2 Start

5.3.2.2.1 General

The following procedures using the Start service operation are supported:

- Timer Start

5.3.2.2.2 Timer Start

Figure 5.3.2.2.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to start a (new or existing) timer identified by the provided timerId.

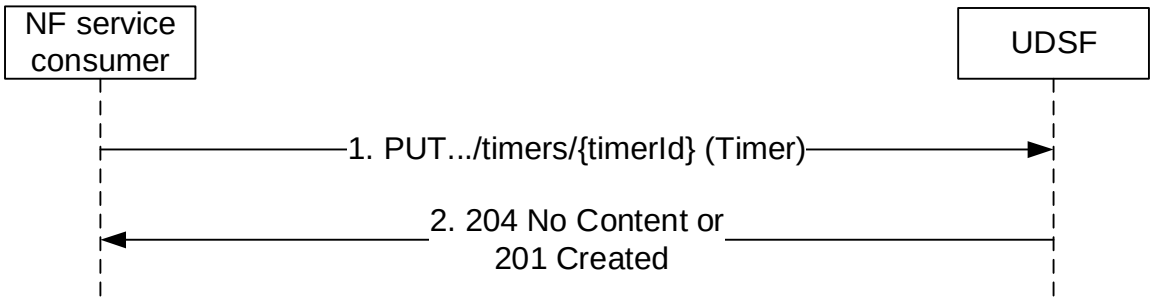

Figure 5.3.2.2.2-1: Starting a Timer

1. The NF service consumer (any NF) sends a PUT request to the resource indicated by timerId.
2. On success, the UDSF responds with "204 No Content" or "201 Created".

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PUT response body.

5.3.2.3 Update

5.3.2.3.1 General

The following procedures using the Update service operation are supported:

- Timer Update

5.3.2.3.2 Timer Update

Figure 5.3.2.3.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to update an existing timer identified by the provided timerId.

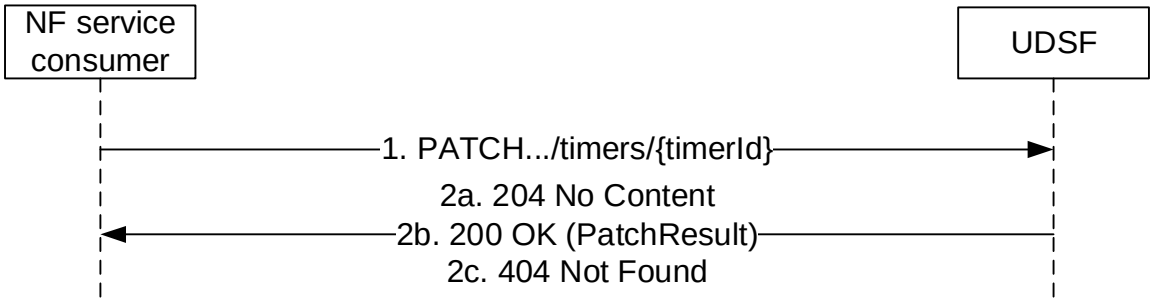

Figure 5.3.2.3.2-1: Modifying a Timer

1. The NF service consumer (any NF) sends a PATCH request to the resource indicated by timerId.

2a. On success, if all the modification instructions in the PATCH request have been implemented, the UDSF shall respond with "204 No Content".

2b. On partial success, i.e. if one or more modification instructions have been discarded, "200 OK" with the execution report, shall be returned.

2c. On failure, the UDSF shall respond with "404 Not Found" if the timer indicated by the timerId does not exist and may include the ProblemDetails.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the PATCH response body.

5.3.2.4 Stop

5.3.2.4.1 General

The following procedures using the Stop service operation are supported:

- Single Timer Stop
- Multiple Timer Stop

5.3.2.4.2 Single Timer Stop

Figure 5.3.2.4.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to stop an existing timer identified by the provided timerId.

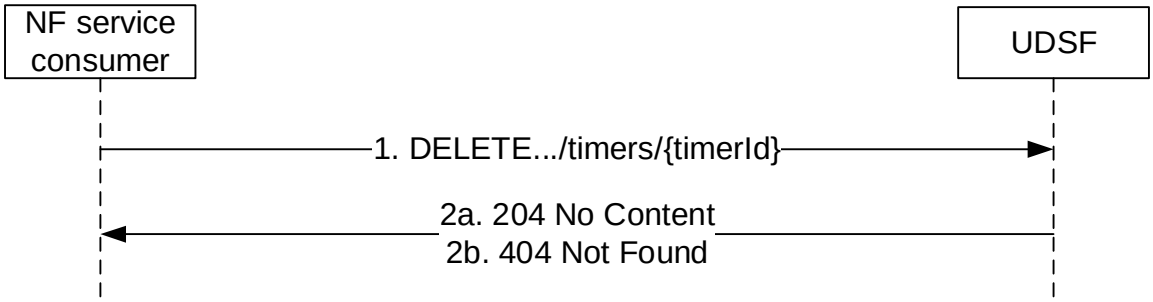

Figure 5.3.2.4.2-1: Stopping a single Timer

1. The NF service consumer (any NF) sends a DELETE request to the resource indicated by timerId.

2a. On success, the UDSF shall respond with "204 No Content".

2b. On failure, the UDSF shall respond with "404 Not Found" if the timer does not exist and may include the ProblemDetails.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the DELETE response body.

5.3.2.4.3 Multiple Timer Stop

Figure 5.3.2.4.3-1 shows a scenario where the NF service consumer sends a request to the UDSF to stop all timers matching a provided filter.

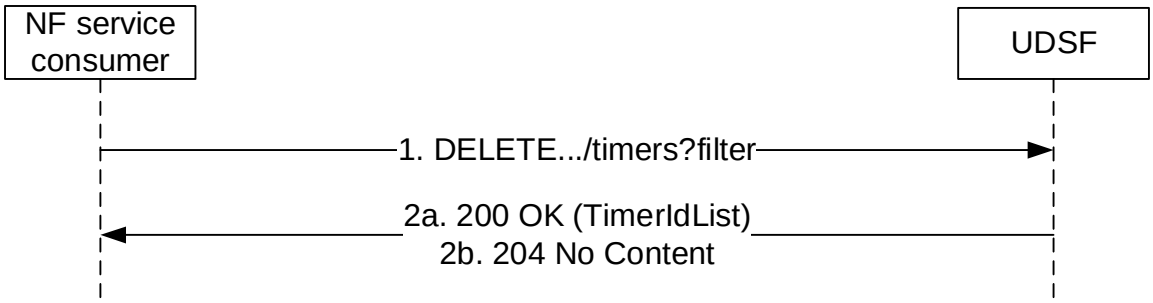

Figure 5.3.2.4.3-1: Stopping multiple Timers

1. The NF service consumer (any NF) sends a DELETE request to the timers resource containing a filter Query Parameter.

2a. On success, the UDSF shall respond with "200 OK" with a message body containing the list of stopped timers.

2b. The UDSF shall respond with "204 No Content" if no timers matching the filter could be found.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the DELETE response body.

5.3.2.5 Search

5.3.2.5.1 General

The following procedures using the Search service operation are supported:

- Expired Timer Search
- Tagged Timer Search

5.3.2.5.2 Expired Timer Search

Figure 5.3.2.5.2-1 shows a scenario where the NF service consumer sends a request to the UDSF to search for expired timers matching a provided expired-filter.

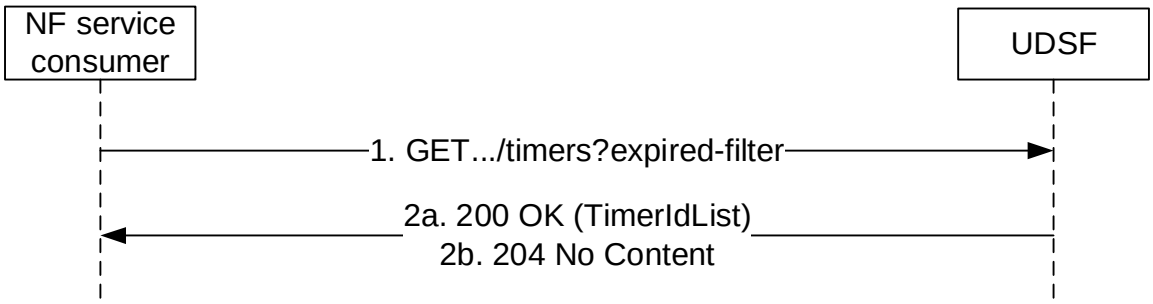

Figure 5.3.2.5.2-1: Searching for expired Timers

1. The NF service consumer (any NF) sends a GET request to the timers resource containing an expired-filter Query Parameter.

2a. On success, the UDSF shall respond with "200 OK" with a message body containing the list of expired timers matching the expired-filter.

2b. The UDSF shall respond with "204 No Content" if no timers matching the filter could be found.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.3.2.5.3 Tagged Timer Search

Figure 5.3.2.5.3-1 shows a scenario where the NF service consumer sends a request to the UDSF to search for tagged timers matching a provided filter.

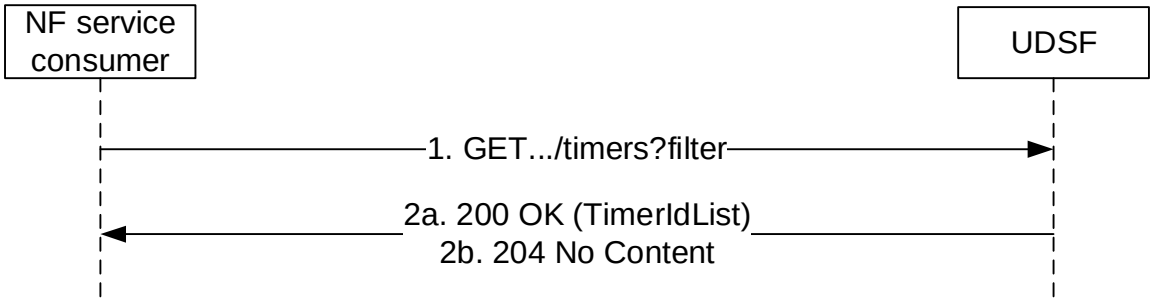

Figure 5.3.2.5.3-1: Searching for tagged Timers

1. The NF service consumer (any NF) sends a GET request to the timers resource containing a filter Query Parameter.

2a. On success, the UDSF shall respond with "200 OK" with a message body containing the list of tagged timers matching the filter.

2b. The UDSF shall respond with "204 No Content" if no timers matching the filter could be found.

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the GET response body.

5.3.2.6 Notify

5.3.2.6.1 General

The following procedures using the Notify service operation are supported:

- Timer Expiry Notify

5.3.2.6.2 Timer Expiry Notify

Figure5.3.2.6.2-1 shows a scenario where the UDSF notifies the NF service consumer of the expired timer.

The Notify is sent by the UDSF to the NF Service Consumer when the timer expires as indicated by the expires attribute of the Timer.

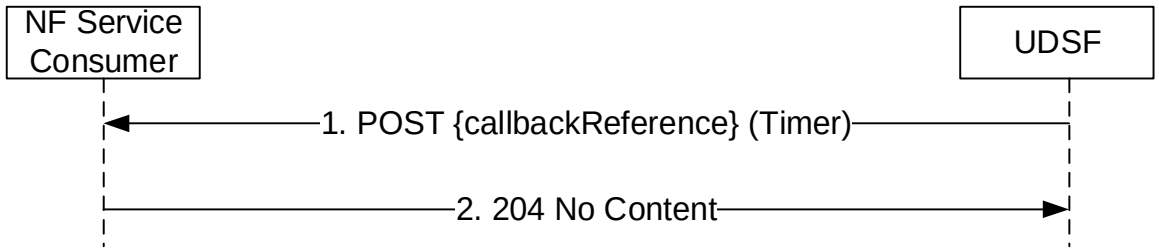

Figure 5.3.2.6.2-1: Timer Expiry Notify

1. The UDSF shall send a POST request to the callback URI. The request shall contain the timer that has expired.

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

On failure, the appropriate HTTP status code indicating the error shall be returned and appropriate additional error information should be returned in the POST response body.

6 API Definitions

6.1 Nudsf_DataRepository Service API

6.1.1 Introduction

The Nudsf_DataRepository service shall use the Nudsf_DataRepository API.

The API URI of the Nudsf_DataRepository API shall be:

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

The request URI used in HTTP requests from the NF service consumer towards the NF service producer shall have the Resource URI structure defined in clause 4.4.1 of 3GPP TS 29.501 [5], i.e.:

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

with the following components:

- The {apiRoot} shall be set as described in 3GPP TS 29.501 [5].
- The <apiName> shall be "nudsf-dr".
- The <apiVersion> shall be "v1".
- The <apiSpecificResourceUriPart> shall be set as described in clause 6.1.3.

6.1.2 Usage of HTTP

6.1.2.1 General

HTTP/2, IETF RFC 7540 [11], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

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

The OpenAPI [6] specification of HTTP messages and content bodies for the Nudsf_DataRepository API is contained in Annex A.

6.1.2.2 HTTP standard headers

6.1.2.2.1 General

See clause 5.2.2 of 3GPP TS 29.500 [4] for the usage of HTTP standard headers.

6.1.2.2.2 Content type

The following content types shall be supported:

- JSON, IETF RFC 8259 [12], shall be used as content type of the HTTP bodies specified in the present specification as specified in clause 5.4 of 3GPP TS 29.500 [4]. The use of the JSON format shall be signalled by the content type "application/json".
- "Problem Details" JSON object shall be used to indicate additional details of the error in a HTTP response body and shall be signalled by the content type "application/problem+json", as defined in IETF RFC 7807 [13].
- JSON Patch (IETF RFC 6902 [14]). The use of the JSON Patch format in a HTTP request body shall be signalled by the content type "application/json-patch+json".

Multipart messages shall also be supported possibly indicating other content-types as described in clause 6.1.2.4.

6.1.2.2.3 Cache-Control

As described in IETF RFC 7234 [17] clause 5.2, a "Cache-Control" header should be included in HTTP responses carrying a representation of cacheable resources. If it is included, it shall contain a "max-age" value, indicating the amount of time in seconds after which the received response is considered stale.

The "max-age" value shall be configurable by operator policy.

6.1.2.2.4 ETag

As described in IETF RFC 7232 [16] clause 2.3, an "ETag" (entity-tag) header should be included in HTTP responses carrying a representation of cacheable or modifiable resources to allow an NF Service Consumer performing a conditional GET request with an "If-None-Match" header or a conditional PUT/PATCH/DELETE request with an "If-Match" header. If it is included, 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.

6.1.2.2.5 If-None-Match

As described in IETF RFC 7232 [16] clause 3.2, an NF Service Consumer may issue conditional GET or PUT requests towards UDSF by including an "If-None-Match" header in HTTP requests containing one or several entity tags received in previous responses for the same resource.

If the If-None-Match header is included with the PUT method, it shall be used with a value of "*" to prevent the inadvertent modification of an existing representation of the target resource when the client believes that the resource does not have a current representation.

6.1.2.2.6 If-Match

As described in IETF RFC 7232 [16] clause 3.1, an NF Service Consumer may issue conditional PUT/PATCH/DELETE request towards UDSF by including an "If-Match" header in HTTP requests containing an entity tag received in previous responses for the same resource.

6.1.2.2.7 Last-Modified

As described in IETF RFC 7232 [16] clause 2.2, a "Last-Modified" header should be included in HTTP responses carrying a representation of cacheable resources to allow an NF Service Consumer performing a conditional request with "If-Modified-Since" header.

6.1.2.2.8 If-Modified-Since

As described in IETF RFC 7232 [16] clause 3.3, an NF Service Consumer may issue conditional GET request towards UDSF, by including an "If-Modified-Since" header in HTTP requests.

6.1.2.2.9 When to Use Entity-Tags and Last-Modified Dates

Both "ETag" and "Last-Modified" headers should be sent in the same HTTP response as stated in IETF RFC 7232 [16] clause 2.4.

NOTE: "ETag" is a stronger validator than the "Last-Modified" and is preferred.

If the UDSF included an "ETag" header with the resource then a conditional GET request for this resource shall be performed with the "If-None-Match" header, and a PUT/PATCH/DELETE request for this resource shall be performed with the "If-Match" header.

6.1.2.2.10 Content-Location

As described in IETF RFC 7231 [15] clause 3.1.4.2, the UDSF shall include the Content-Location header set to the URI of the expired Record when sending a Notification to an NF Consumer.

6.1.2.3 HTTP custom headers

The mandatory HTTP custom header fields specified in clause 5.2.3.2 of 3GPP TS 29.500 [4] shall be applicable.

6.1.2.4 HTTP multipart messages

6.1.2.4.1 General

HTTP multipart messages shall be supported to transfer the opaque Record and Block Information in the following service operations (and HTTP messages):

- Record (PUT/GET/DELETE)
- BlockCollection (GET)
- RecordNotification (POST)

NOTE: In the clauses below, this specification deviates in the definition of the Content-Id header from IETF RFC 2045 [20] in that there is no need for the Content-Id to be "world unique", as in this specification the Content-Id is only required to be unique within the context of a Record, BlockCollection or RecordNotification.

6.1.2.4.2 Record

The Record is encoded as an HTTP multipart message with the multipart/mixed content-type as described in IETF RFC 2046 [21].

The boundary parameter is used to delimit each part (Start of parts to RecordMeta, RecordMeta to Block, Block to Block and Block to End of parts) and shall be set to a value as in accordance with IETF RFC 2046 [21].

The RecordMeta part shall be the first part, and shall always be present. It shall include a Content-Id header that may be set to an unquoted value of "meta" or any other suitable value, for example as described in IETF RFC 2045 [20], the Content-Type (see IETF RFC 2045 [20]) header set to "application/json" and include JSON content for the RecordMeta.

Zero or more Block parts may follow the RecordMeta Part. For each Block part in the HTTP multipart/mixed message, the Block part shall include a Content-Id header identifying the Block with the value of the blockId, a Content-Type (see IETF RFC 2045 [20]) header indicating the MIME type of the Block (any value), e.g. application/octet-stream, application/json or another applicable MIME type and the Content-Transfer-Encoding (see IETF RFC 2045 [20]) header with an appropriate value.

6.1.2.4.3 BlockCollection

The BlockCollection is encoded as an HTTP multipart message with the multipart/parallel content-type as described in IETF RFC 2046 [21].

The boundary parameter is used to delimit each part (Start of parts to Block, Block to Block and Block to End of parts) and shall be set to a value as in accordance with IETF RFC 2046 [21].

For the BlockCollection, zero or more Block parts may be included. For each Block part in the HTTP multipart/parallel message, the Block part shall include a Content-Id header identifying the Block with the value of the blockId, a Content-Type header indicating the MIME type of the Block (any value) e.g. application/octet-stream, application/json or another applicable MIME type and the Content-Transfer-Encoding (see IETF RFC 2045 [20]) header with an appropriate value.

6.1.2.4.4 RecordNotification

The RecordNotification is encoded as an HTTP multipart message with the multipart/mixed content-type as described in IETF RFC 2046 [21].

The boundary parameter is used to delimit each part (Start of parts to RecordNotification, RecordNotification to RecordMeta, RecordMeta to Block, Block to Block and Block to End of parts) and shall be set to a value in accordance with IETF RFC 2046 [21].

The RecordNotification shall be the first part, and shall always be present. It shall include a Content-Id header set to an unquoted value of "meta" or any other suitable value, for example as described in IETF RFC 2045 [20], the Content-Type (see IETF RFC 2045 [20]) header set to "application/json" and include JSON content for the NotificationDescription.

The RecordMeta part shall be the second part, and shall always be present. It shall include a Content-Id header set to an unquoted value of "meta" or any other suitable value, for example as described in IETF RFC 2045 [20], the Content-Type (see IETF RFC 2045 [20]) header set to "application/json" and include JSON content for the RecordMeta.

Zero or more Block parts may follow the RecordMeta Part. For each Block part in the HTTP multipart/mixed message, the Block part shall include a Content-Id header identifying the Block with the value of the blockId, a Content-Type (see IETF RFC 2045 [20]) header indicating the MIME type of the Block (any value), e.g. application/octet-stream, application/json or another applicable MIME type and the Content-Transfer-Encoding (see IETF RFC 2045 [20]) header with an appropriate value.

6.1.3 Resources

6.1.3.1 Overview

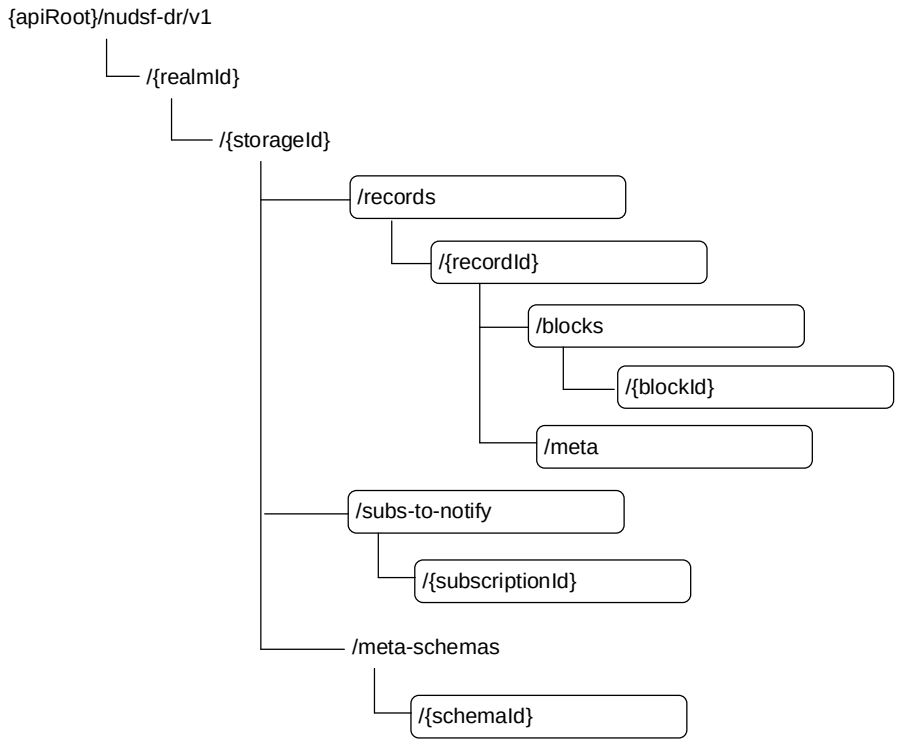

Figure 6.1.3.1-1: Resource URI structure of the nudsf-dr API

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
RecordCollection (Collection)	/{realmId}/{storageId}/records	GET	Search for records
		DELETE	Delete records
Record (Document)	/{realmId}/{storageId}/records/{recordId}	GET	Retrieve a record
		PUT	Create or update a record
		DELETE	Delete a record
Meta (Document)	/{realmId}/{storageId}/records/{recordId}/meta	GET	Retrieve the meta of a record
		PATCH	Modify the meta of a record
BlockCollection (Collection)	/{realmId}/{storageId}/records/{recordId}/blocks	GET	Retrieve all the blocks of a record
Block (Document)	/{realmId}/{storageId}/records/{recordId}/blocks/{blockId}	GET	Retrieve a block
		PUT	Create or update a block
		DELETE	Delete a block
NotificationSubscriptions (Collection)	/{realmId}/{storageId}/subs-to-notify	GET	Retrieve existing subscriptions
Individual NotificationSubscription (Document)	/{realmId}/{storageId}/subs-to-notify/{subscriptionId}	DELETE	Delete the subscription identified by {subscriptionId}, i.e. unsubscribe to notification for change of data
		PATCH	Update an individual Subscription to notification
		PUT	Create or update a subscription to notification,
		GET	Retrieve an individual Subscription to notification
Meta Schema (Document)	/{realmId}/{storageId}/meta-schemas/{schemaId}	GET	Retrieve a Meta Schema
		PUT	Create or update a Meta Schema
		DELETE	Delete a Meta Schema

6.1.3.2 Resource: RecordCollection (Collection)

6.1.3.2.1 Description

This resource represents the collection of records within a storage. It can be used to search for specific records matching specific filter criteria.

6.1.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/records**

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 clause 6.1.1
realmId	Represents the realm Id.
storageId	Represents the storage Id.

6.1.3.2.3 Resource Standard Methods

6.1.3.2.3.1 GET

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	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
filter	SearchExpression	O	0..1	The filter criteria for searching the records of the storage.	
limit-range	Uinteger	C	0..1	When set, the returned response shall contain at the most the number of record references specified by the parameter value. If the count-indicator parameter is set in the request, this parameter shall be ignored.	
count-indicator	boolean	O	0..1	If this parameter is set, the number of records that matched the criteria shall be returned and no record references shall be returned.	
retrieve-records	RetrieveRecords	O	0..1	If this parameter is set, the content of records that matched the criteria shall be returned. If count-indicator is set, this parameter shall not be set.	CombinedSearchRetrieve
max-payload-size	Uinteger	O	0..1	Maximum payload size (before compression, if any) of the response, expressed in kilo octets. When present, the UDSF shall limit the number of Records or Meta returned in the response so as not to exceed the maximum payload size indicated in the request.	CombinedSearchRetrieve

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
RecordSearchResult	M	1	200 OK	The search result containing the record references matching the filter.
n/a			204 No Content	The search did not result in any matching record references.
NOTE: The manadatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.2.3.2 DELETE

This method shall support the URI query parameters specified in table 6.1.3.2.3.2-1.

Table 6.1.3.2.3.2-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
filter	SearchExpression	M	1	The filter criteria for identifying the records to be deleted.	BulkOperations

This method shall support the request data structures specified in table 6.1.3.2.3.2-2 and the response data structures and response codes specified in table 6.1.3.2.3.2-3.

Table 6.1.3.2.3.2-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.2.3.2-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The bulk deletion request did not result in any matching records to be deleted.
RecordIdList	M	1	200 OK	Contains the list of record IDs identifying the deleted records.
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.3 Resource: Record (Document)

6.1.3.3.1 Description

This resource represents a record within a storage.

6.1.3.3.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/records/{recordId}**

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 clause 6.1.1
realmId	Represents the realm Id where the record is stored
storageId	Represents the storage Id where the record is stored
recordId	Represents the record Id of the record

6.1.3.3.3 Resource Standard Methods

6.1.3.3.3.1 GET

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	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6.	

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
RecordBody	M	1	200 OK	A response body containing the record.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND
NOTE: The mandatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.3.3.2 PUT

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	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
get-previous	boolean	O	0..1	Request to return the previous record content if a record already exists in the targeted storage for the same record identifier.	

When creating or replacing the record, meta and zero or more blocks shall be included. The record meta information shall be the first part and is mandatory. The remaining parts of the body (if any) shall be the blocks to be updated or created. See clause 6.1.2.4 for details on the encoding.

If the operation updates an existing record, then the existing record and its blocks shall be discarded and replaced by the meta and blocks supplied with the request. This also applies to the case when no new blocks are included, i.e. the old blocks (if any) shall be deleted from the record.

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
Record	M	1	The record that is to be created including meta and zero or more blocks.

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
RecordBody	M	1	200 OK	Upon successful update of a record, a response body containing the previous record value (if get-previous was indicated in the request, and if one exists) will be returned
RecordBody	O	0..1	201 Created	Upon successful creation of a record, a response body of the created record shall be returned. If due to operator's policy the value of the ttl (if any) in the request exceeded a maximum allowed ttl value with the ttl set to the value applied by the UDSF.
n/a			204 No Content	Upon successful update of a record, an empty response is returned or if no previous record value was requested.
ProblemDetails	O	0..1	403 FORBIDDEN	If the UDSF (based on operator policy), determines to apply a ttl value of the recordMeta different from the ttl value (if any) of the request, the get-previous query-parameter is present in the request and the request applies to a record that already exists in the UDSF, the "cause" attribute shall be set to one of the following application errors: -TTL_VALUE_NOT_ALLOWED
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
RecordBody	O	0..1	412 Precondition Failed	If one or more conditions given in the request header fields evaluated to false and get-previous was indicated in the request, the UDSF shall include the RecordBody in the response.
NOTE: The mandatory HTTP error status code for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.3.3.3 DELETE

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 DELETE method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
get-previous	boolean	O	0..1	Request to return the record content if a record exists in the targeted storage for the same record identifier.	

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 DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

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

Data type	P	Cardinality	Response codes	Description
RecordBody	O	0..1	200 OK	Upon success, and a response body containing the record value with meta and associated blocks (if any), if get-previous was indicated in the request.
n/a			204 No Content	Upon success and no response body containing the record was requested.
RecordBody	O	0..1	412 Preconditio n Failed	If one or more conditions given in the request header fields evaluated to false and get-previous was indicated in the request, the UDSF shall include the RecordBody in the response.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.4 Resource: Meta (Document)

6.1.3.4.1 Description

This resource represents the meta associated with a record.

6.1.3.4.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/records/{recordId}/meta**

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 clause 6.1.1
realmId	Represents the realm Id where the record is stored
storageId	Represents the storage Id where the record is stored
recordId	Represents the record Id of the record

6.1.3.4.3 Resource Standard Methods

6.1.3.4.3.1 GET

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 GET method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

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 GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

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

Data type	P	Cardinality	Response codes	Description
RecordMeta	M	1	200 OK	A response body containing the record meta will be returned.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND
NOTE: The mandatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.4.3.2 PATCH

This method shall support the URI query parameters specified in table 6.1.3.4.3.2-1.

Table 6.1.3.4.3.2-1: URI query parameters supported by the PATCH method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.1.3.4.3.2-2 and the response data structures and response codes specified in table 6.1.3.4.3.2-3.

Table 6.1.3.4.3.2-2: Data structures supported by the PATCH Request Body on this resource

Data type	P	Cardinality	Description
array(patchItem)	M	1..N	A collection of patch items to apply on the record meta.

Table 6.1.3.4.3.2-3: Data structures supported by the PATCH Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Upon successful modification, there is no body in the response message. (NOTE 2)
PatchResult	M	1	200 OK	If one or more modification instructions have been discarded, the execution report is returned. (NOTE 2)
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute maybe used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND
NOTE 1: In addition common data structures as listed in table 6.1.7.3-1are supported. NOTE 2: If all the modification instructions in the PATCH request have been implemented, the UDSF shall respond with 204 No Content response; if some of the modification instructions in the PATCH request have been discarded, the UDSF shall respond with PatchResult.				

6.1.3.5 Resource: BlockCollection (Collection)

6.1.3.5.1 Description

This resource represents the collection of blocks of associated with a record.

6.1.3.5.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/records/{recordId}/blocks**

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 clause 6.1.1
realmId	Represents the realm Id where the record is stored
storageId	Represents the storage Id where the record is stored
recordId	Represents the record Id of the record

6.1.3.5.3 Resource Standard Methods

6.1.3.5.3.1 GET

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 GET method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

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 GET 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 GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
array(Block)	M	1..N	200 OK	A response body containing one or more blocks.
n/a			204 No Content	The BlockCollection did not contain any blocks.
NOTE: The mandatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.6 Resource: Block (Document)

6.1.3.6.1 Description

This resource represents a record within a storage.

6.1.3.6.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/records/{recordId}/blocks/{blockId}**

This resource shall support the resource URI variables defined in table 6.1.3.6.2-1.

Table 6.1.3.6.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 6.1.1
realmId	Represents the realm Id where the record is stored
storageId	Represents the storage Id where the record is stored
recordId	Represents the record Id of the record
blockId	Represents the block Id of the block

6.1.3.6.3 Resource Standard Methods

6.1.3.6.3.1 GET

This method shall support the URI query parameters specified in table 6.1.3.6.3.1-1.

Table 6.1.3.6.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.1.3.6.3.1-2 and the response data structures and response codes specified in table 6.1.3.6.3.1-3.

Table 6.1.3.6.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.6.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
BlockBody	M	1	200 OK	Upon success, a response body containing the requested block is returned.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute maybe used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND -BLOCK_NOT_FOUND
NOTE: The mandatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.6.3.2 PUT

This method shall support the URI query parameters specified in table 6.1.3.6.3.2-1.

Table 6.1.3.6.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
get-previous	boolean	O	0..1	Request to return the previous block content if a block already exists in the targeted storage for the same block identifier.	

If the operation updates an existing block, then the existing block shall be discarded and replaced by the block supplied with the request.

This method shall support the request data structures specified in table 6.1.3.6.3.2-2 and the response data structures and response codes specified in table 6.1.3.6.3.2-3.

Table 6.1.3.6.3.2-2: Data structures supported by the PUT Request Body on this resource

Data type	P	Cardinality	Description
Block	M	1	The block definition that shall be created. A Content-Type http header can be set to specify the media type of the block's opaque content. If the media-type is not included, the media-type shall be set to application/octet-stream.

Table 6.1.3.6.3.2-3: Data structures supported by the PUT Response Body on this resource

Data type	P	Cardinality	Response codes	Description
BlockBody	M	1	200 OK	Upon successful update of a block, a response body containing the previous record value ((if get-previous was indicated in the request and if one exists) will be returned
n/a			201 Created	Upon successful creation of a record, an empty response shall be returned.
n/a			204 No Content	Upon successful update of a record, an empty response is returned if no previous record value was requested.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND
BlockBody	O	0..1	412 Preconditio n Failed	If one or more conditions given in the request header fields evaluated to false and get-previous was indicated in the request, the UDSF shall include the BlockBody in the response.
NOTE: The mandatory HTTP error status code for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.6.3.3 DELETE

This method shall support the URI query parameters specified in table 6.1.3.6.3.3-1.

Table 6.1.3.6.3.3-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
get-previous	boolean	O	0..1	Request to return the previous block content (if any).	

This method shall support the request data structures specified in table 6.1.3.6.3.3-2 and the response data structures and response codes specified in table 6.1.3.6.3.3-3.

Table 6.1.3.6.3.3-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.6.3.3-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
BlockBody	O	0..1	200 OK	Upon success, and a response body containing the block was requested.
n/a			204 No Content	Upon success, and no response body containing the block was requested.
BlockBody	O	0..1	412 Preconditio n Failed	If one or more conditions given in the request header fields evaluated to false and get-previous was indicated in the request, the UDSF shall include the BlockBody in the response.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -RECORD_NOT_FOUND -BLOCK_NOT_FOUND
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.7 Resource: NotificationSubscriptions

6.1.3.7.1 Description

This resource is used to represent notification subscriptions.

6.1.3.7.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/subs-to-notify**

This resource shall support the resource URI variables defined in table 6.1.3.7.2-1.

Table 6.1.3.7.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 6.1.1
realmId	Represents the realm Id.
storageId	Represents the storage Id.

6.1.3.7.3 Standard Methods

6.1.3.7.3.2 GET

This method shall support the URI query parameters specified in table 6.1.3.7.3.2-1.

Table 6.1.3.7.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6
limit-range	Uinteger	C	0..1	When set, the returned response shall contain at the most the number of Notification Subscriptions specified by the parameter value.

This method shall support the request data structures specified in table 6.1.3.7.3.2-2 and the response data structures and response codes specified in table 6.1.3.7.3.2-3.

Table 6.1.3.7.3.2-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.7.3.2-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
array(NotificationSubscription)	M	0..N	200 OK	Upon success, a response body containing the individual subscriptions shall be returned.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
NOTE: The manadatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.8 Resource: IndividualNotificationSubscription

6.1.3.8.1 Description

This resource is used to represent an individual subscriber data subscriptions to notifications.

6.1.3.8.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/subs-to-notify/{subscriptionId}**

This resource shall support the resource URI variables defined in table 6.1.3.8.2-1.

Table 6.1.3.8.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 6.1.1
realmId	Represents the realm Id.
storageId	Represents the storage Id.
subscriptionId	The subscriptionId identifies an individual NotficiationSubscription.

6.1.3.8.3 Resource Standard Methods

6.1.3.8.3.1 DELETE

This method shall support the URI query parameters specified in table 6.1.3.8.3.1-1.

Table 6.1.3.8.3.1-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description
client-id	ClientId	M	1	The client-id is used by the UDSF to guard against deletion of notification subscriptions that do not belong to the same NF or NFSet.
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6
get-previous	boolean	O	0..1	Request to return the associated NotificationSubscription if it exist.

This method shall support the request data structures specified in table 6.1.3.8.3.1-2 and the response data structures and response codes specified in table 6.1.3.8.3.1-3.

Table 6.1.3.8.3.1-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			The request body shall be empty.

Table 6.1.3.8.1-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
NotificationSubscription	O	0..1	200 OK	Upon successful delete of a NotificationSubscription, a response body containing the NotificationSubscription value before the delete shall be returned if get-previous was indicated in the request.
n/a			204 NO CONTENTt	Upon success, an empty response body shall be returned.
ProblemDetails	O	0..1	403 FORBIDDEN	If the client-id query parameter does not match the clientId of the existing resource, 403 FORBIDDEN shall be returned and the "cause" attribute may be used to indicate one of the following application errors: - SUBSCRIPTION_EXISTS
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -SUBSCRIPTION_NOT_FOUND
NotificationSubscription	O	0..1	412 PRECONDITION FAILED	412 PRECONDITION FAILED is returned if one or more conditions given in the request header fields evaluated to false. If get-previous was indicated in the request, the UDSF shall include the NotificationSubscription in the response.
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.8.3.2 PATCH

This method shall support the URI query parameters specified in table 6.1.3.8.3.2-1.

Table 6.1.3.8.3.2-1: URI query parameters supported by the PATCH method on this resource

Name	Data type	P	Cardinality	Description
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6

This method shall support the request data structures specified in table 6.1.3.8.3.2-2 and the response data structures and response codes specified in table 6.1.3.8.3.2-3.

Table 6.1.3.8.3.2-2: Data structures supported by the PATCH Request Body on this resource

Data type	P	Cardinality	Description
array(PatchItem)	M	1..N	Contains the delta data of the Notification Subscription to be updated.

Table 6.1.3.8.3.2-3: Data structures supported by the PATCH Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 NO CONTENT	Upon successful modification, there is no body in the response message. (NOTE 2)
PatchResult	M	1	200 OK	Upon success, the execution report is returned. (NOTE 2)
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -SUBSCRIPTION_NOT_FOUND
NOTE 1: In addition common data structures as listed in table 6.1.7.3-1are supported. NOTE 2: If all the modification instructions in the PATCH request have been implemented, the UDSF shall respond with 204 No Content response; if some of the modification instructions in the PATCH request have been discarded, the UDSF shall respond with 200 OK and include the PatchResult.				

6.1.3.8.3.3 GET

This method shall support the URI query parameters specified in table 6.1.3.8.3.3-1.

Table 6.1.3.8.3.3-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6.

This method shall support the request data structures specified in table 6.1.3.8.3.3-2 and the response data structures and response codes specified in table 6.1.3.8.3.3-3.

Table 6.1.3.8.3.3-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			The request body shall be empty.

Table 6.1.3.8.3.3-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
NotificationSubscription	M	1	200 OK	Upon success, a response body containing the individual subscriptions shall be returned.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -SUBSCRIPTION_NOT_FOUND
NOTE: The mandatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.8.3.4 PUT

This method shall support the URI query parameters specified in table 6.1.3.8.3.4-1.

Table 6.1.3.8.3.4-1: URI query parameters supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.1.3.8.3.4-2 and the response data structures and response codes specified in table 6.1.3.8.3.4-3.

Table 6.1.3.8.3.4-2: Data structures supported by the PUT Request Body on this resource

Data type	P	Cardinality	Description
NotificationSubscription	M	1	The Notification Subscription.

Table 6.1.3.8.3.4-3: Data structures supported by the PUT Response Body on this resource

Data type	P	Cardinality	Response codes	Description
NotificationSubscription	M	1	201 CREATED	Upon success, the created NotificationSubscription is returned. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
NotificationSubscription	M	1	200 OK	Upon success, the updated NotificationSubscription is returned.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
ProblemDetails	O	0..1	403 FORBIDDEN	If the clientId of the PUT request does not match the clientId of the existing resource, 403 FORBIDDEN shall be returned and the "cause" attribute may be used to indicate one of the following application errors: - SUBSCRIPTION_EXISTS
array(Uri)	O	1..N	409 CONFLICT	If one or more monitoredResourceUris from the request don't exist in the UDSF, 409 CONFLICT shall be returned together with the non-existing monitoredResourceUris.
NOTE: The mandatory HTTP error status codes for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.9 Resource: Meta Schema (Document)

6.1.3.9.1 Description

This resource represents a meta schema within a storage.

6.1.3.9.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-dr/v1/{realmId}/{storageId}/meta-schemas/{schemaId}**

This resource shall support the resource URI variables defined in table 6.1.3.9.2-1.

Table 6.1.3.9.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 6.1.1
realmId	Represents the realm Id where the schema is stored
storageId	Represents the storage Id where the schema is stored
schemaId	Represents the schema Id of the schema

6.1.3.9.3 Resource Standard Methods

6.1.3.9.3.1 GET

This method shall support the URI query parameters specified in table 6.1.3.9.3.1-1.

Table 6.1.3.9.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6.	Meta Schema

This method shall support the request data structures specified in table 6.1.3.9.3.1-2 and the response data structures and response codes specified in table 6.1.3.9.3.1-3.

Table 6.1.3.9.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.9.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
MetaSchema	M	1	200 OK	A response body containing the meta schema.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -SCHEMA_NOT_FOUND
NOTE: The mandatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.3.9.3.2 PUT

This method shall support the URI query parameters specified in table 6.1.3.9.3.2-1.

Table 6.1.3.9.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	Meta Schema
get-previous	boolean	O	0..1	Request to return the previous meta schema content if the meta schema already exists in the targeted storage for the same schema identifier.	Meta Schema

This method shall support the request data structures specified in table 6.1.3.9.3.2-2 and the response data structures and response codes specified in table 6.1.3.9.3.2-3.

Table 6.1.3.9.3.2-2: Data structures supported by the PUT Request Body on this resource

Data type	P	Cardinality	Description
MetaSchema	M	1	The schema that is to be created/updated.

Table 6.1.3.9.3.2-3: Data structures supported by the PUT Response Body on this resource

Data type	P	Cardinality	Response codes	Description
MetaSchema	M	1	200 OK	Upon successful update of a schema, a response body containing the previous schema value (if get-previous was indicated in the request, and if one exists) will be returned
MetaSchema	M	1	201 Created	Upon successful creation of a schema, a response body of the created schema schall be returned. The HTTP response shall include a "Location" HTTP header that contains the resource URI of the created resource.
n/a			204 No Content	Upon successful update of a schema, an empty response is returned if no previous schema value was requested.
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
NOTE: The mandatory HTTP error status code for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

Table 6.1.3.9.3.2-4: Headers supported by the 201 Response Code on this resource

Name	Data type	P	Cardinality	Description
Location	string	M	1	Contains the URI of the newly created resource, according to the structure: {apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/meta-schemas/{schemaId}

6.1.3.9.3.3 DELETE

This method shall support the URI query parameters specified in table 6.1.3.9.3.3-1.

Table 6.1.3.9.3.3-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	Meta Schema
get-previous	boolean	O	0..1	Request to return the schema content if a schema exists in the targeted storage for the same schema identifier.	Meta Schema

This method shall support the request data structures specified in table 6.1.3.9.3.3-2 and the response data structures and response codes specified in table 6.1.3.9.3.3-3.

Table 6.1.3.9.3.3-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.1.3.9.3.3-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
MetaSchema	O	0..1	200 OK	Upon success, a response body containing the schema value, if get-previous was indicated in the request.
n/a			204 No Content	Upon success, if get-previous was not indicated in the request..
ProblemDetails	O	0..1	404 Not Found	The "cause" attribute may be used to indicate one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -SCHEMA_NOT_FOUND
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.4 Custom Operations without associated resources

None.

6.1.5 Notifications

6.1.5.1 General

Notifications shall comply to clause 6.2 of 3GPP TS 29.500 [4] and clause 4.6.2.3 of 3GPP TS 29.501 [5].

6.1.5.2 Timer Expiry Notification

6.1.5.2.1 Description

The Timer Expiry Notification is used by the NF service producer to report to an NF Consumer that the Record has expired as indicated by the ttl attribute (if set) of RecordMeta and if a callbackReference was set in the RecordMeta.

6.1.5.2.2 Target URI

The Callback URI **"{callbackReference}"** shall be used with the callback URI variables defined in table 6.1.5.2.2-1.

Table 6.1.5.2.2-1: Target URI variables for this resource

Name	Definition
callbackReference	string formatted as URI with the Callback Uri

6.1.5.2.3 Standard Methods

6.1.5.2.3.1 POST

This method shall support the request data structures specified in table 6.1.5.2.3.1-1 and the response data structures and response codes specified in table 6.1.5.2.3.1-1.

Table 6.1.5.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
RecordBody	M	1	The RecordBody of the record that was deleted.

Table 6.1.5.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Upon success, an empty response body shall be returned.
NOTE: The mandatory HTTP error status codes for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.5.3 Notification due to Data Change

6.1.5.3.1 Description

The Notification due to Data Change is used by the UDSF to report to an NF Consumer that a Record that is part of a NotificationSubscription has been updated or deleted.

6.1.5.3.2 Target URI

The Callback URI **"{callbackReference}"** shall be used with the callback URI variables defined in table 6.1.5.3.2-1.

Table 6.1.5.3.2-1: Callback URI variables for this resource

Name	Definition
callbackReference	String formatted as URI with the Callback Uri

6.1.5.3.3 Standard Methods

6.1.5.3.3.1 POST

This method shall support the request data structures specified in table 6.1.5.3.3.1-1 and the response data structures and response codes specified in table 6.1.5.3.3.1-1.

Table 6.1.5.3.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
RecordNotification	M	1	The notification with the record information.

Table 6.1.5.3.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 NO CONTENT	Upon success, an empty response body shall be returned.
NOTE: The mandatory HTTP error status codes for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.5.4 Subscription Expiry Notification

6.1.5.4.1 Description

The Subscription Expiry Notification is used by the UDSF to report to an NF Consumer that a Subscription to Notification due to Data Change has expired or is about to expire.

6.1.5.4.2 Target URI

The Callback URI **"{expiryCallbackReference}"** shall be used with the callback URI variables defined in table 6.1.5.4.2-1.

Table 6.1.5.4.2-1: Callback URI variables for this resource

Name	Definition
expiryCallbackReference	String formatted as URI with the Callback URI

6.1.5.4.3 Standard Methods

6.1.5.4.3.1 POST

This method shall support the request data structures specified in table 6.1.5.4.3.1-1 and the response data structures and response codes specified in table 6.1.5.4.3.1-1.

Table 6.1.5.4.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
NotificationInfo	M	1	The notification info with expired subscriptions.

Table 6.1.5.4.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Upon success, an empty response body shall be returned.
NOTE: The mandatory HTTP error status codes for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.1.6 Data Model

6.1.6.1 General

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

Table 6.1.6.1-1 specifies the data types defined for the N_{udsf} service based interface protocol. For simple data types defined for the Nudsf_DataRepository service API see table 6.1.6.3.2-1.

Table 6.1.6.1-1: N_{udsf} specific Data Types

Data type	Clause defined	Description	Applicability
RecordSearchResult	6.1.6.2.2	Record Search Result	
RecordMeta	6.1.6.2.3	Record Meta	
RecordBody	6.1.6.2.4	Record Body	
Record	6.1.6.2.5	Record	
BlockBody	6.1.6.2.6	Block Body	
Block	6.1.6.2.7	Block	
SearchCondition	6.1.6.2.8	Search Condition	
SearchComparison	6.1.6.2.9	Search Comparison	
ComparisonOperator	6.1.6.3.3	Comparison Operator	
ConditionOperator	6.1.6.3.4	Condition Operator	
SearchExpression	6.1.6.4.1	Search Expression	
NotificationSubscription	6.1.6.2.10	Notification Subscription	
RecordNotification	6.1.6.2.11	Record Notification	
NotificationDescription	6.1.6.2.12	Notification Description	
SubscriptionFilter	6.1.6.2.13	Subscription Filter	
ClientId	6.1.6.2.14	Client Identity	
MetaSchema	6.1.6.2.15	Meta Schema	Meta Schema
TagType	6.1.6.2.16	Tag Type	Meta Schema
RecordIdList	6.1.6.2.17	List of Record IDs	BulkOperations
NotificationInfo	6.1.6.2.18	Notification Info	
RecordOperation	6.1.6.3.5	Record Operation	
SchemaId	6.1.6.3.2	Identifier of a Meta Schema	Meta Schema
KeyType	6.1.6.3.6	Key Type	Meta Schema
RetrieveRecords	6.1.6.3.7	Request to return matching records	CombinedSearchRetrieve

Table 6.1.6.1-2 specifies data types re-used by the N_{udsf} 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 N_{udsf} service based interface.

Table 6.1.6.1-2: N_{udsf} re-used Data Types

Data type	Reference	Comments	Applicability
SupportedFeatures	3GPP TS 29.571 [19]	see 3GPP TS 29.500 [4] clause 6.6.	
PatchItem	3GPP TS 29.571 [19]	Data structure used for JSON patch.	
PatchResult	3GPP TS 29.571 [19]		
Uri	3GPP TS 29.571 [19]		
DateTime	3GPP TS 29.571 [19]		
NfInstanceId	3GPP TS 29.571 [19]		
NfSetId	3GPP TS 29.571 [19]		

6.1.6.2 Structured data types

6.1.6.2.1 Introduction

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

6.1.6.2.2 Type: RecordSearchResult

Table 6.1.6.2.2-1: Definition of type RecordSearchResult

Attribute name	Data type	P	Cardinality	Description	Applicability
count	Uinteger	M	1	The number of records found by the search.	
references	array(Uri)	O	1..N	The Record references found by the search.	
supportedFeatures	SupportedFeatures	O	0..1	See clause 6.1.8	
matchingRecords	map(Record)	C	1..N	This attribute contains the records that match the search filters provided in the request. The key of the map is the recordId. It shall be present if the search request included an instruction to include record content in the response. The map may contain a subset of the matching records in the case where inclusion of more records would result in the payload size exceeding the max-payload-size received in the request (if any).	CombinedSearchRetrieve

6.1.6.2.3 Type: RecordMeta

Table 6.1.6.2.3-1: Definition of type RecordMeta

Attribute name	Data type	P	Cardinality	Description	Applicability
tags	map(array(string))	O	1..N(1..M)	A map of tag name/values pairs, where the tag name is a unique string name that is the primary key of the map and is paired with an array of string values.	
ttl	DateTime	O	0..1	ttl refers to the lifetime of the record. After the expiry, the record shall be deleted.	
callbackReference	Uri	O	0..1	The Uri where the NF Service Consumer shall receive notification on the expiry of the Record as indicated by the ttl attribute if desired.	
schemaId	SchemaId	O	0..1	Id of the MetaSchema to which the tags comply to	Meta Schema

6.1.6.2.4 Type: RecordBody

Table 6.1.6.2.4-1: Definition of type RecordBody

Attribute name	Data type	P	Cardinality	Description	Applicability
record	Record	M	1	The record.	

6.1.6.2.5 Type: Record

Table 6.1.6.2.5-1: Definition of type Record

Attribute name	Data type	P	Cardinality	Description	Applicability
meta	RecordMeta	M	1	The meta of a record.	
blocks	array(Block)	O	1..N	The block(s) (if any) making up the record.	

6.1.6.2.6 Type: BlockBody

Table 6.1.6.2.6-1: Definition of type BlockBody

Attribute name	Data type	P	Cardinality	Description	Applicability
n/a	Block	M	1	The block.	

6.1.6.2.7 Type: Block

Table 6.1.6.2.7-1: Definition of type Block

Attribute name	Data type	P	Cardinality	Description	Applicability
value	Any Type	M	1	The block value of any data type.	

6.1.6.2.8 Type: SearchCondition

Table 6.1.6.2.8-1: Definition of type SearchCondition

Attribute name	Data type	P	Cardinality	Description	Applicability
cond	ConditionOperator	M	1	Logical operator ("AND", "OR" or "NOT")	
units	array(SearchExpression)	M	1..N	For the logical "NOT" operator indicated in the cond attribute, only one member shall be present in the array. For the logical "AND" or "OR" operators indicated in the cond attribute, at least two members shall be present in the array and all the members in the array shall be interpreted as logically concatenated with the logical operator.	
schemaId	SchemaId	O	0..1	When included, the search is limited to records which have the given schemaId value stored in their RecordMeta	Meta Schema

6.1.6.2.9 SearchComparison

Table 6.1.6.2.9-1: Definition of type SearchComparison

Attribute name	Data type	P	Cardinality	Description	Applicability
op	ComparisonOperator	M	1	Comparison operator	
tag	string	M	1	This attribute contains the tag name of an array of strings.	
value	string	M	1	The array of strings indicated in the tag attribute compares to the value of this attribute.	

6.1.6.2.10 Type: NotificationSubscription

Table 6.1.6.2.10-1: Definition of type NotificationSubscription

Attribute name	Data type	P	Cardinality	Description	Applicability
clientId	ClientId	M	1	Identity of the NF or NFSet for which the subscription applies.	
callbackReference	Uri	M	1	Identifies the NF or NF pool where the notification shall be sent.	
expiryCallbackReference	Uri	C	0..1	Identifies the NF or NF pool where the expiry notification shall be sent. Shall be present if the expiryNotification is present.	
expiry	DateTime	C	0..1	This IE shall be included in a subscription response, if, based on operator policy and taking into account the expiry time included in the request, the UDSF needs to include an expiry time. The expiry time, based on operator policy, may indicate a value that is sooner than the NF consumer requested. The absence of this attribute in the subscription response indicates that the subscription does not have an expiry time.	

expiryNotification	Uinteger	O	0..1	This attribute when present in the request, indicates to the UDSF that the UDSF shall Notify the NF Consumer about the expiry of the subscription if this capability is supported by the UDSF. -A value of 0 in the subscription request indicates that the UDSF shall notify the NF Consumer upon subscription expiry. -A value greater than 0 in the subscription request, indicates the number of seconds before the expiry of the subscription that the UDSF shall notify the NF Consumer about the expiry. -A value that results in the notification to occur in the past shall be treated the same as a value of 0 was received. A UDSF that supports this capability shall include the attribute in the subscription response and in the Notification, set to the value received in the request (or set to 0 if the notification would occur in the past as described above). For example, if the expiry time in the subscription request indicates a time 100 seconds from now and the expiryNotification indicates 110 seconds, the UDSF shall return a value of 0 in the subscription response, meaning the NF Consumer will be notified upon subscription expiry.	
subFilter	SubscriptionFilter	O	0..1	If not included, the subscription applies to all Create, Delete and Update events of the storage.	
supportedFeatures	SupportedFeatures	O	0..1	Used to negotiate the applicability of optional features	

6.1.6.2.11 Type: RecordNotification

Table 6.1.6.2.11-1: Definition of type RecordNotification

Attribute name	Data type	P	Cardinality	Description	Applicability
descriptor	NotificationDescription	M	1	The block value of any data type.	
meta	RecordMeta	M	1	The meta of a record.	
blocks	array(Block)	O	1..n	The block(s) (if any) making up the record.	

6.1.6.2.12 Type: NotificationDescription

Table 6.1.6.2.12-1: Definition of type NotificationDescription

Attribute name	Data type	P	Cardinality	Description	Applicability
recordRef	Uri	M	1	The reference of the record triggering the Notification.	
operationType	RecordOperation	M	1	The operation type.	
subscriptionId	string	O	0..1	This IE shall contain the subscriptionId that uniquely identifies the subscription to notification within a storage when present.	

6.1.6.2.13 Type: SubscriptionFilter

Table 6.1.6.2.13-1: Definition of type SubscriptionFilter

Attribute name	Data type	P	Cardinality	Description	Applicability
monitoredResourceUris	array(Uri)	O	1..N	A set of URIs that identify the records for which a modification of the representation triggers a notification. The URI shall take the form of either an absolute URI or an absolute-path reference as defined in IETF RFC 3986 [22], also see NOTE 1. The monitored resource shall indicate an existing record. If not present, the subscription is applied to all records within the storage and a modification of the representation of any record within the storage triggers a notification.	
operations	array(RecordOperation)	O	0..3	The operations that shall generate a notification. If the monitoredResourceUris is present, only "UPDATED" and "DELETED" are allowed values, any other value shall be ignored. If the attribute is not present, all applicable operations shall apply to the subscription.	

NOTE 1: The UDSF should handle only the relative-path part (apiSpecificResourceUriPart, see 3GPP TS 29.501 [5] clause 4.4.1) and ignore possible inconsistencies (caused by e.g. an SCP) in the base URI part.

6.1.6.2.14 Type: ClientId

Table 6.1.6.2.14-1: Definition of type ClientId

Attribute name	Data type	P	Cardinality	Description	Applicability
nfId	NfInstanceId	C	0..1	The NF Instance Id uniquely identifying the NF Consumer. Shall be present if the nfSetId is absent.	
nfSetId	NfSetId	C	0..1	The NF Set Id of the NF Consumer. Shall be present if the nfId is absent.	

6.1.6.2.15 Type: MetaSchema

Table 6.1.6.2.15-1: Definition of type MetaSchema

Attribute name	Data type	P	Cardinality	Description	Applicability
schemaId	SchemaId	M	1	Id of the schema. Used as reference.	Meta Schema
metaTags	array(TagType)	M	1..N	Array of tag types that describe the schema	Meta Schema

6.1.6.2.16 Type: TagType

Table 6.1.6.2.16-1: Definition of type TagType

Attribute name	Data type	P	Cardinality	Description	Applicability
tagName	string	M	1	Name of the tag. Used as reference. Refers to the unique tag string name that is the primary key of the map and is paired with an array of string values in the RecordMeta tags IE	Meta Schema
keyType	KeyType	M	1	Type of key	Meta Schema
sort	boolean	C	0..1	Shall be present if keyType is "SEARCH_KEY" or "SEARCH_AND_COUNT_KEY" true: indicates that searches based on "GT", "GTE", "LT" and / or "LTE" are expected. false: indicates otherwise	Meta Schema
presence	boolean	O	0..1	true: indicates that presence of the tag is mandatory in the RecordMeta false (default): indicates that presence of the tag is optional in the RecordMeta	Meta Schema

6.1.6.2.17 Type: RecordIdList

Table 6.1.6.2.17-1: Definition of type RecordIdList

Attribute name	Data type	P	Cardinality	Description	Applicability
recordIdList	array(string)	M	1..N	List of Record IDs to be retrieved	BulkOperations

6.1.6.2.18 Type: NotificationInfo

Table 6.1.6.2.18-1: Definition of type NotificationInfo

Attribute name	Data type	P	Cardinality	Description	Applicability
expiredSubscriptions	array(NotificationSubscription)	M	1	An array of one or more expired subscriptions.	

6.1.6.3 Simple data types and enumerations

6.1.6.3.1 Introduction

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

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	Applicability
SchemaId	string	Id of the schema. Used as reference.	Meta Schema

6.1.6.3.3 Enumeration: ComparisonOperator

The enumeration ComparisonOperator represents the comparison of an array of strings to a string value. The comparison shall be based on lexicographical order. It shall comply with the provisions defined in table 6.1.6.3.3-1.

Table 6.1.6.3.3-1: Enumeration ComparisonOperator

Enumeration value	Description	Applicability
"EQ"	The array contains the string value.	
"NEQ"	The array does not contain the string value.	AdvancedQuery
"GT"	The array contains a string that is greater than the string value.	AdvancedQuery
"GTE"	The array contains a string that is greater than or equal to the string value.	AdvancedQuery
"LT"	The array contains a string that is less than the string value.	AdvancedQuery
"LTE"	The array contains a string that is less than or equal to the string value.	AdvancedQuery

NOTE: It's recommended to use GT/GTE/LT/LTE on single value tags. If not, the logical operator "NOT" applied over the comparison operator "GT" evaluates to "true" if there are no members in the array that are greater than the value.

6.1.6.3.4 Enumeration: ConditionOperator

Table 6.1.6.3.4-1: Enumeration ConditionOperator

Enumeration value	Description	Applicability
"AND"	Logical "AND"	
"OR"	Logical "OR"	
"NOT"	Logical "NOT"	

6.1.6.3.5 Enumeration: RecordOperation

Table 6.1.6.3.5-1: Enumeration RecordOperation

Enumeration value	Description	Applicability
"CREATED"	Indicates a Create record operation	
"UPDATED"	Indicates an Update record operation	
"DELETED"	Indicates a Delete record operation	

6.1.6.3.6 Enumeration: KeyType

Table 6.1.6.3.6-1: Enumeration KeyType

Enumeration value	Description	Applicability
"UNIQUE_KEY"	Tags with this key type may be used for search and will result in no or one matching records.	Meta Schema
"SEARCH_KEY"	Tags with this key type may be used for search and will result in no, one or more matching records.	Meta Schema
"COUNT_KEY"	Tags with this key type may be used for count.	Meta Schema
"SEARCH_AND_COUNT_KEY"	Tags with this key type may be used for seach and count.	Meta Schema
"OTHER_TAG"	Tags with this key type may not be used for search or count.	Meta Schema

6.1.6.3.7 Enumeration: RetrieveRecords

Table 6.1.6.3.7-1: Enumeration RetrieveRecords

Enumeration value	Description	Applicability
"ONLY_META"	Only the META of matching records are requested to be included in the RecordSearchResult.	CombinedSearchRetrieve
"META_AND_BLOCKS"	META and BLOCKS of matching records are requested to be included in the RecordSearchResult.	CombinedSearchRetrieve

6.1.6.4 Data types describing alternative data types or combinations of data types

6.1.6.4.1 Type: SearchExpression

Table 6.1.6.4.1-1: Definition of type SearchExpression as a list of mutually exclusive alternatives

Data type	Cardinality	Description	Applicability
SearchCondition	1	A search expression with logic operators	AdvancedQuery
SearchComparison	1	A minimum unit of the search expression	
RecordIdList	1	List of Record IDs identifying records to be retrieved	BulkOperations

6.1.7 Error Handling

6.1.7.1 General

For the Nudsf_DataRepository API, HTTP error responses shall be supported as specified in clause 4.8 of 3GPP TS 29.501 [5]. Protocol errors and application errors specified in table 5.2.7.2-1 of 3GPP TS 29.500 [4] shall be supported for an HTTP method if the corresponding HTTP status codes are specified as mandatory for that HTTP method in table 5.2.7.1-1 of 3GPP TS 29.500 [4].

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

6.1.7.2 Protocol Errors

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

6.1.7.3 Application Errors

The application errors defined for the Nudsf_DataRepository service are listed in Table 6.1.7.3-1.

Table 6.1.7.3-1: Application errors

Application Error	HTTP status code	Description
TTL_VALUE_NOT_ALLOWED	403 Forbidden	The ttl value indicated in the request exceeds the maximum value allowed in the UDSF.
SUBSCRIPTION_EXISTS	403 Forbidden	The subscription indicated in the HTTP/2 request is not authorized to be operated.
SCHEMA_IN_USE	403 Forbidden	The schema cannot be deleted as it is still referenced by existing records.
REALM_NOT_FOUND	404 Not Found	The realm indicated in the HTTP/2 request is unavailable in the UDSF.
STORAGE_NOT_FOUND	404 Not Found	The storage indicated in the HTTP/2 request is unavailable in the UDSF.
RECORD_NOT_FOUND	404 Not Found	The record indicated in the HTTP/2 request is unavailable in the UDSF.
BLOCK_NOT_FOUND	404 Not Found	The block indicated in the HTTP/2 request is unavailable in the UDSF.
SUBSCRIPTION_NOT_FOUND	404 Not Found	The subscription indicated in the HTTP/2 request is unavailable in the UDSF.
SCHEMA_NOT_FOUND	404 Not Found	The schema indicated in the HTTP/2 request is unavailable in the UDSF.

6.1.8 Feature negotiation

The optional features in table 6.1.8-1 are defined for the Nudsf_DataRepository API. They shall be negotiated using the extensibility mechanism defined in clause 6.6 of 3GPP TS 29.500 [4].

Table 6.1.8-1: Supported Features

Feature number	Feature Name	Description
1	AdvancedQuery	If an NF consumer detects that the UDSF supports the AdvancedQuery feature, it may use values of the ComparisonOperator besides "EQ" and may also use the cond attribute of the SearchCondition. If an NF consumer detects that the UDSF does not support the AdvancedQuery feature, it shall only use a value of "EQ" of the ComparisonOperator and shall not use the cond attribute of the SearchCondition.
2	Meta Schema	This feature supports optimization of UDSF data storage and allows the UDSF to know in advance how data search access by the NF consumer is expected. If the NF consumer detects that the UDSF does not support the Meta Schema feature, it shall not make use of the procedures for storing, updating and deleting Meta Schemas.
3	CombinedSearchRetrieve	This feature supports optimization of search and retrieval of records stored in the UDSF. When searching records (see clause 5.2.2.2.6), consumers supporting this feature can instruct the supporting UDSF to return matching records in addition to the matching records' URIs. If the NF consumer detects that the UDSF does not support the CombinedSearchRetrieve feature, it shall not include the query parameters retrieve-records and max-payload-size in record search requests.
4	BulkOperations	This feature supports optimization for retrieval/deletion of huge number of records identified by a list of record IDs. When the CombinedSearchRetrieve feature has been used to request retrieval of matching records but only a subset of matching records have been received due to maximum payload size limitations, the consumers supporting this feature can instruct the supporting UDSF to return remaining records identified by a recordIdList. If the NF consumer detects that the UDSF does not support the BulkOperations feature, it shall not include the RecordIdList in the filter query parameter in record search requests.

6.1.9 Security

As indicated in 3GPP TS 33.501 [8] and 3GPP TS 29.500 [4], the access to the Nudsf_DataRepository API may be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [9]), based on local configuration, using the "Client Credentials" authorization grant, where the NRF (see 3GPP TS 29.510 [10]) plays the role of the authorization server.

If OAuth2 is used, an NF Service Consumer, prior to consuming services offered by the <API Name> API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in 3GPP TS 29.510 [10], clause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF that the NF Service Consumer used for discovering the Nudsf_DataRepository service.

The Nudsf_DataRepository API defines a single scope "nudsf-dr" for the entire service, and it does not define any additional scopes at resource or operation level.

6.2 Nudsf_Timer Service API

6.2.1 Introduction

The Nudsf_Timer service shall use the Nudsf_Timer API.

The API URI of the Nudsf_Timer API shall be:

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

The request URI used in HTTP requests from the NF service consumer towards the NF service producer shall have the Resource URI structure defined in clause 4.4.1 of 3GPP TS 29.501 [5], i.e.:

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

with the following components:

- The {apiRoot} shall be set as described in 3GPP TS 29.501 [5].
- The <apiName> shall be "nudsf-timer".
- The <apiVersion> shall be "v1".
- The <apiSpecificResourceUriPart> shall be set as described in clause 6.2.3.

6.2.2 Usage of HTTP

6.2.2.1 General

HTTP/2, IETF RFC 7540 [11], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

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

The OpenAPI [6] specification of HTTP messages and content bodies for the Nudsf_Timer API is contained in Annex A.

6.2.2.2 HTTP standard headers

6.2.2.2.1 General

See clause 5.2.2 of 3GPP TS 29.500 [4] for the usage of HTTP standard headers.

6.2.2.2.2 Content type

The following content types shall be supported:

- JSON, IETF RFC 8259 [12], shall be used as content type of the HTTP bodies specified in the present specification as specified in clause 5.4 of 3GPP TS 29.500 [4]. The use of the JSON format shall be signalled by the content type "application/json".
- "Problem Details" JSON object shall be used to indicate additional details of the error in a HTTP response body and shall be signalled by the content type "application/problem+json", as defined in IETF RFC 7807 [13].
- JSON Patch (IETF RFC 6902 [14]). The use of the JSON Patch format in a HTTP request body shall be signalled by the content type "application/json-patch+json".

6.2.2.3 HTTP custom headers

The mandatory HTTP custom header fields specified in clause 5.2.3.2 of 3GPP TS 29.500 [4] shall be applicable.

6.2.3 Resources

6.2.3.1 Overview

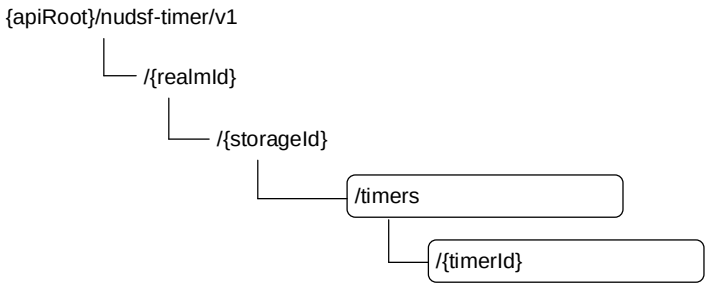

Figure 6.2.3.1-1: Resource URI structure of the nudsf-timer API

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
Timers (Store)	/{realmId}/{storageId}/timers	GET	Search and retrieve timers matching a filter
		DELETE	Stop (delete) timers matching a filter
Individual Timer (Document)	/{realmId}/{storageId}/timers/{timerId}	PUT	Start (create) or update (replace) a timer
		PATCH	Modify a timer
		DELETE	Stop (delete) a timer
		GET	Retrieve a timer

6.2.3.2 Resource: Timers (Store)

6.2.3.2.1 Description

This resource represents the collection of timers within a storage. It can be used to search for specific timers matching specific filter criteria, and to delete specific timers matching specific filter criteria.

6.2.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-timer/<apiVersion>/{realmId}/{storageId}/timers**

This resource shall support the resource URI variables defined in table 6.2.3.2.2-1.

Table 6.2.3.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 6.1.1
realmId	Represents the realm Id.
storageId	Represents the storage Id.

6.2.3.2.3 Resource Standard Methods

6.2.3.2.3.1 GET

This method shall support the URI query parameters specified in table 6.2.3.2.3.1-1.

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	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
filter	SearchExpression	C	0..1	The filter criteria for searching the timers of the storage. Shall be present if expired-filter is absent; otherwise may be present.	
expired-filter	NullValue	C	0..1	A timer matches the expired-filter if the timer's expiry time is earlier than the current time. Shall be present if filter is absent; otherwise may be present.	
NOTE: If the query parameters filter and expired-filter are both present in the request, the result shall contain all timers that match both query parameters.					

This method shall support the request data structures specified in table 6.2.3.2.3.1-2 and the response data structures and response codes specified in table 6.2.3.2.3.1-3.

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
TimerIdList	M	1	200 OK	The search result containing the IDs of timers matching the filter.
n/a			204 No Content	The search did not result in any matching timers.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
NOTE: The manadatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.2.3.2.3.2 DELETE

This method shall support the URI query parameters specified in table 6.2.3.2.3.2-1.

Table 6.2.3.2.3.2-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	
filter	SearchExpression	C	0..1	The filter criteria for identifying the timers of the storage that are to be deleted. Shall be present if expired-filter is absent; otherwise may be present.	
expired-filter	NullValue	C	0..1	Presence of this query parameter indicates that only expired timers are to be deleted. Shall be present if filter is absent; otherwise may be present.	
NOTE: If the query parameters filter and expired-filter are both present in the request, all timers that match both query parameters shall be deleted.					

This method shall support the request data structures specified in table 6.2.3.2.3.2-2 and the response data structures and response codes specified in table 6.2.3.2.3.2-3.

Table 6.2.3.2.3.2-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.2.3.2.3.2-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
TimerIdList	M	1	200 OK	Contains the list of timer IDs identifying the deleted timers.
n/a			204 No Content	The search did not result in any matching timers.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
NOTE: The manadatory HTTP error status code for the GET method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.2.3.3 Resource: Individual Timer (Document)

6.2.3.3.1 Description

This resource represents an individual timer within a storage.

6.2.3.3.2 Resource Definition

Resource URI: **{apiRoot}/nudsf-timer/<apiVersion>/{realmId}/{storageId}/timers/{timerId}**

This resource shall support the resource URI variables defined in table 6.2.3.3.2-1.

Table 6.2.3.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See clause 6.1.1
realmId	Represents the realm Id.
storageId	Represents the storage Id.
timerId	Represents the timer Id.

6.2.3.3.3 Resource Standard Methods

6.2.3.3.3.1 PUT

This method shall support the URI query parameters specified in table 6.2.3.3.3.1-1.

Table 6.2.3.3.3.1-1: URI query parameters supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.2.3.3.3.1-2 and the response data structures and response codes specified in table 6.2.3.3.3.1-3.

Table 6.2.3.3.3.1-2: Data structures supported by the PUT Request Body on this resource

Data type	P	Cardinality	Description
Timer	M	1	The timer to be created or replaced.

Table 6.2.3.3.3.1-3: Data structures supported by the PUT Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			201 Created	Upon successful creation of a timer, an empty response shall be returned.
n/a			204 No Content	Upon successful replacement of a timer, an empty response shall be returned.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND
NOTE: The manadatory HTTP error status code for the PUT method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.2.3.3.3.2 PATCH

This method shall support the URI query parameters specified in table 6.2.3.3.3.2-1.

Table 6.2.3.3.3.2-1: URI query parameters supported by the PATCH method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.2.3.3.3.2-2 and the response data structures and response codes specified in table 6.2.3.3.3.2-3.

Table 6.2.3.3.3.2-2: Data structures supported by the PATCH Request Body on this resource

Data type	P	Cardinality	Description
array(PatchItem)	M	1..N	Contains the delta data of the Timer to be updated.

Table 6.2.3.3.3.2-3: Data structures supported by the PATCH Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Upon successful modification of a timer, an empty response shall be returned.
PatchResult	M	1	200 OK	Upon success, the execution report is returned. (NOTE 2)
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -TIMER_NOT_FOUND
NOTE 1: In addition common data structures as listed in table 6.1.7.3-1 are supported. NOTE 2: If all the modification instructions in the PATCH request have been implemented, the UDSF shall respond with 204 No Content response; if some of the modification instructions in the PATCH request have been discarded, the UDSF shall respond with 200 OK and include the PatchResult.				

6.2.3.3.3.3 DELETE

This method shall support the URI query parameters specified in table 6.2.3.3.3.3-1.

Table 6.2.3.3.3.3-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.2.3.3.3.3-2 and the response data structures and response codes specified in table 6.2.3.3.3.3-3.

Table 6.2.3.3.3.3-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.2.3.3.3.3-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Upon successful deletion of a timer, an empty response shall be returned.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -TIMER_NOT_FOUND
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.2.3.3.4 GET

This method shall support the URI query parameters specified in table 6.2.3.3.4-1.

Table 6.2.3.3.4-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description	Applicability
supported-features	SupportedFeatures	O	0..1	see 3GPP TS 29.500 [4] clause 6.6	

This method shall support the request data structures specified in table 6.2.3.3.4-2 and the response data structures and response codes specified in table 6.2.3.3.4-3.

Table 6.2.3.3.4-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 6.2.3.3.4-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
Timer	M	1	200 OK	Upon success the timer shall be returned.
ProblemDetails	O	0..1	404 NOT FOUND	The "cause" attribute shall be set to one of the following application errors: -REALM_NOT_FOUND -STORAGE_NOT_FOUND -TIMER_NOT_FOUND
NOTE: The mandatory HTTP error status code for the DELETE method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.2.4 Custom Operations without associated resources

None.

6.2.5 Notifications

6.2.5.1 General

Notifications shall comply to clause 6.2 of 3GPP TS 29.500 [4] and clause 4.6.2.3 of 3GPP TS 29.501 [5].

6.2.5.2 Timer Expiry Notification

6.2.5.2.1 Description

The Timer Expiry Notification is used by the NF service producer to report to an NF Consumer that the Timer has expired as indicated by the expires attribute of Timer and if a callbackReference was set in the Timer.

6.2.5.2.2 Target URI

The Callback URI **"{callbackReference}"** shall be used with the callback URI variables defined in table 6.2.5.2.2-1.

Table 6.2.5.2.2-1: Target URI variables for this resource

Name	Definition
callbackReference	string formatted as URI with the Callback Uri

6.2.5.2.3 Standard Methods

6.2.5.2.3.1 POST

This method shall support the request data structures specified in table 6.2.5.2.3.1-1 and the response data structures and response codes specified in table 6.2.5.2.3.1-1.

Table 6.2.5.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
Timer	M	1	The timer that expired

Table 6.2.5.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Upon success, an empty response body shall be returned.
NOTE: The mandatory HTTP error status codes for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.				

6.2.6 Data Model

6.2.6.1 General

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

Table 6.2.6.1-1 specifies the data types defined for the Nudsf_Timer service API. For simple data types defined for the Nudsf_Timer service API see table 6.2.6.3.2-1.

Table 6.2.6.1-1: Nudsf_Timer specific Data Types

Data type	Clause defined	Description	Applicability
Timer	6.2.6.2.2	Timer	

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

Table 6.2.6.1-2: Nudsf_Timer re-used Data Types

Data type	Reference	Comments	Applicability
SupportedFeatures	3GPP TS 29.571 [19]	see 3GPP TS 29.500 [4] clause 6.6.	
PatchItem	3GPP TS 29.571 [19]	Data structure used for JSON patch.	
PatchResult	3GPP TS 29.571 [19]		
Uri	3GPP TS 29.571 [19]		
DateTime	3GPP TS 29.571 [19]		
NullValue	3GPP TS 29.571 [19]		
SearchExpression	6.1.6.4.1		

6.2.6.2 Structured data types

6.2.6.2.1 Introduction

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

6.2.6.2.2 Type: Timer

Table 6.2.6.2.2-1: Definition of type Timer

Attribute name	Data type	P	Cardinality	Description	Applicability
timerId	TimerId	C	0..1	Identifier of the timer; shall be present in Notification request messages; otherwise shall be absent.	
expires	DateTime	M	1	Point in time of expiry	
metaTags	map(array(string))	O	1..N	A map (list of key-value pairs where tagName serves as key) of tagValue lists.	
callbackReference	Uri	O	0..1	The URI where the NF Service Consumer shall receive notification on the expiry of the Timer. Shall be absent from Notification request messages.	
deleteAfter	Uinteger	O	0..1	If specified, the timer resource will be deleted following a period after the expires time of the number of seconds defined by the deleteAfter value. If not specified, the timer shall be deleted immediately after the expires time.	

6.2.6.2.3 Type: TimerIdList

Table 6.2.6.2.3-1: Definition of type TimerIdList

Attribute name	Data type	P	Cardinality	Description	Applicability
timerIds	array(TimerId)	M	1..N	List of Timer IDs	

6.2.6.3 Simple data types and enumerations

6.2.6.3.1 Introduction

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

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	Applicability
TimerId	string	Id of a timer. Used as reference.	

6.2.7 Error Handling

6.2.7.1 General

For the Nudsf_Timer API, HTTP error responses shall be supported as specified in clause 4.8 of 3GPP TS 29.501 [5]. Protocol errors and application errors specified in table 5.2.7.2-1 of 3GPP TS 29.500 [4] shall be supported for an HTTP method if the corresponding HTTP status codes are specified as mandatory for that HTTP method in table 5.2.7.1-1 of 3GPP TS 29.500 [4].

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

6.2.7.2 Protocol Errors

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

6.2.7.3 Application Errors

The application errors defined for the Nudsf_Timer service are listed in Table 6.2.7.3-1.

Table 6.2.7.3-1: Application errors

Application Error	HTTP status code	Description
REALM_NOT_FOUND	404 Not Found	The realm indicated in the HTTP/2 request is unavailable in the UDSF.
STORAGE_NOT_FOUND	404 Not Found	The storage indicated in the HTTP/2 request is unavailable in the UDSF.
TIMER_NOT_FOUND	404 Not Found	The Timer indicated in the HTTP/2 request is unavailable in the UDSF.
EXPIRES_VALUE_NOT_ALLOWED	403 Forbidden	The timer could not be started as the expires time is in the past.

6.2.8 Feature negotiation

The optional features in table 6.1.8-1 are defined for the Nudsf_Timer API. They shall be negotiated using the extensibility mechanism defined in clause 6.6 of 3GPP TS 29.500 [4].

Table 6.2.8-1: Supported Features

Feature number	Feature Name	Description
n/a		

6.2.9 Security

As indicated in 3GPP TS 33.501 [8] and 3GPP TS 29.500 [4], the access to the Nudsf_Timer API may be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [9]), based on local configuration, using the "Client Credentials" authorization grant, where the NRF (see 3GPP TS 29.510 [10]) plays the role of the authorization server.

If OAuth2 is used, an NF Service Consumer, prior to consuming services offered by the Nudsf_Timer API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in 3GPP TS 29.510 [10], clause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF that the NF Service Consumer used for discovering the Nudsf_Timer service.

The Nudsf_Timer API defines a single scope "nudsf-timer" for the entire service, and it does not define any additional scopes at resource or operation level.

Annex A (normative): OpenAPI specification

A.1 General

This Annex specifies the formal definition of the API(s) defined in the present specification. It consists of OpenAPI 3.0.0 specifications in YAML format.

This Annex takes precedence when being discrepant to other parts of the specification with respect to the encoding of information elements and methods within the API(s).

NOTE: The semantics and procedures, as well as conditions, e.g. for the applicability and allowed combinations of attributes or values, not expressed in the OpenAPI definitions but defined in other parts of the specification also apply.

Informative copies of the OpenAPI specification files contained in this 3GPP Technical Specification are available on a Git-based repository that uses the GitLab software version control system (see 3GPP TS 29.501 [5] clause 5.3.1 and 3GPP TR 21.900 [7] clause 5B).

A.2 Nudsf_DataRepository API

```
openapi: 3.0.0
info:
  title: Nudsf_DataRepository
  version: 1.1.0-alpha.5
  description: |
    Nudsf Data Repository Service.
    © 2021, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.

externalDocs:
  description: 3GPP TS 29.598 UDSF Services, V17.4.0.
  url: 'http://www.3gpp.org/ftp/Specs/archive/29_series/29.598/'

servers:
  - url: '{apiRoot}/nudsf-dr/v1'
    variables:
      apiRoot:
        default: https://example.com
        description: apiRoot as defined in clause 4.4 of 3GPP TS 29.501

security:
  - {}
  - oAuth2ClientCredentials:
    - nudsf-dr

paths:

  /{realmId}/{storageId}/records:
    summary: Access to all Records of a Storage
    description: >-
      root of all Records of a Storage
    get:
      summary:     Records search with get
      description: Retrieve one or multiple Records based on filter
      operationId: SearchRecord
      tags:
        - Record CRUD
      parameters:
        - name: realmId
          in: path
          description: Identifier of the Realm
          required: true
          schema:
            type: string
            example: Realm01
        - name: storageId
          in: path
          description: Identifier of the Storage
```

```
          required: true
          schema:
            type: string
            example: Storage01
        - name: limit-range
          in: query
          description: The most number of record references to fetch
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Uinteger'
        - name: filter
          in: query
          description: Query filter using conditions on tags
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/SearchExpression'
        - name: count-indicator
          in: query
          description: Indicates whether the number of records that matched the criteria shall be
returned.
          schema:
            type: boolean
            default: 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: retrieve-records
          in: query
          description: Indicates whether the UDSF is requested to include matching records within
the response.
          schema:
            $ref: '#/components/schemas/RetrieveRecords'
        - name: max-payload-size
          in: query
          description: Indicates the number of kilo octets the consumer is prepared to receive
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Uinteger'
      responses:
        '200':
          description: Successful case. Response contains result of the search.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/RecordSearchResult'
        '204':
          description: >-
            The search condition does not match any Record.
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '406':
          $ref: 'TS29571_CommonData.yaml#/components/responses/406'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    delete:
      summary:     Bulk Deletion of Records
      description: Delete multiple Records based on filter
      operationId: BulkDeleteRecords
      tags:
        - Record CRUD
      parameters:
        - name: realmId
          in: path
          description: Identifier of the Realm
          required: true
```

```
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: filter
        in: query
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SearchExpression'
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        description: Successful case. Response contains RecordIdList.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/RecordIdList'
      '204':
        description: Successful case.
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '406':
        $ref: 'TS29571_CommonData.yaml#/components/responses/406'
      '429':
        $ref: 'TS29571_CommonData.yaml#/components/responses/429'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /{realmId}/{storageId}/records/{recordId}:
    summary: Access to a specific Record, identified by its RecordId
    description: >-
      Access to a specific Record
    get:
      summary: Record access
      description: retrieve one specific Record
      operationId: GetRecord
      tags:
      - Record CRUD
      parameters:
        - name: realmId
          in: path
          description: Identifier of the Realm
          required: true
          schema:
            type: string
            example: Realm01
        - name: storageId
          in: path
          description: Identifier of the Storage
          required: true
          schema:
            type: string
            example: Storage01
        - name: recordId
          in: path
          description: Identifier of the Record
```

```
        required: true
        schema:
          type: string
          example: 'UserRecordValue000000001'
      - name: If-None-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.3
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200' : #result ok
        $ref: '#/components/responses/RecordBody'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  put:
    summary: Create/Modify Record
    description: Create or Modify a Record with a user provided RecordId
    operationId: CreateOrModifyRecord
    tags:
      - Record CRUD
    parameters:
      - name: realmId
        in: path
        description: Identifier(name) of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: recordId
        in: path
        description: Identifier of the Record
        required: true
        schema:
          type: string
          example: UserRecordValue000000001
      - name: If-None-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: If-Match
        in: header
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: get-previous
        in: query
```

```
      description: Retrieve the Record before update
      required: false
      schema:
        type: boolean
        default: false
    - name: supported-features
      in: query
      description: Features required to be supported by the target NF
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
  requestBody:
    $ref: '#/components/requestBodies/RecordBody'
  callbacks:
    recordExpired:
      '{$request.body#/callbackReference}':
        post:
          parameters:
            - name: Content-Location
              in: header
              description: The expired record URI
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
          requestBody:
            $ref: '#/components/requestBodies/RecordBody'
          responses:
            '204':
              description: Callback executed successfully
            '400':
              $ref: 'TS29571_CommonData.yaml#/components/responses/400'
            '401':
              $ref: 'TS29571_CommonData.yaml#/components/responses/401'
            '403':
              $ref: 'TS29571_CommonData.yaml#/components/responses/403'
            '500':
              $ref: 'TS29571_CommonData.yaml#/components/responses/500'
            '503':
              $ref: 'TS29571_CommonData.yaml#/components/responses/503'
            default:
              $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  responses:
    '200' : # Update with return
      $ref: '#/components/responses/RecordBody'
    '201':
      description: >-
        Create case. The resource has been successfully created, location header indicates
        the URI of the created Record.
        $ref: '#/components/responses/RecordBody'
      headers:
        Location:
          $ref: '#/components/headers/Location'
        Cache-Control:
          $ref: '#/components/headers/Cache-Control'
        ETag:
          $ref: '#/components/headers/ETag'
        Last-Modified:
          $ref: '#/components/headers/Last-Modified'
    '204': # Update without return
      description: >-
        Update case. The resource has been successfully updated and no
        additional content is included in the response message.
      headers:
        Cache-Control:
          $ref: '#/components/headers/Cache-Control'
        ETag:
          $ref: '#/components/headers/ETag'
        Last-Modified:
          $ref: '#/components/headers/Last-Modified'
    '304':
      $ref: '#/components/responses/304'
    '400':
      $ref: 'TS29571_CommonData.yaml#/components/responses/400'
    '401':
      $ref: 'TS29571_CommonData.yaml#/components/responses/401'
    '403':
      $ref: 'TS29571_CommonData.yaml#/components/responses/403'
    '404':
      $ref: 'TS29571_CommonData.yaml#/components/responses/404'
    '408':
```

```
        $ref: 'TS29571_CommonData.yaml#/components/responses/408'
      '412': # Return Record value if get-previous=true
        $ref: '#/components/responses/RecordBody'
      '413':
        $ref: 'TS29571_CommonData.yaml#/components/responses/413'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  delete:
    summary: Delete a Record with an user provided RecordId
    operationId: DeleteRecord
    tags:
      - Record CRUD
    parameters:
      - name: realmId
        in: path
        description: Identifier(name) of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: recordId
        in: path
        description: Identifier of the Record
        required: true
        schema:
          type: string
          example: UserRecordValue000000001
      - name: If-Match
        in: header
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: get-previous
        in: query
        description: Retrieve the Record before delete
        required: false
        schema:
          type: boolean
          default: false
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        $ref: '#/components/responses/RecordBodyDelete'
      '204':
        description: Successful case.
        headers:
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '408':
        $ref: 'TS29571_CommonData.yaml#/components/responses/408'
      '412': # Return return value if get-previous=true
```

```
          $ref: '#/components/responses/RecordBody'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /{realmId}/{storageId}/records/{recordId}/meta:
    summary: Access to the meta of a specific Record, identified by its RecordId
    description: >-
      Access to the meta of a specific Record
    get:
      summary: Record's meta access
      description: retrieve meta of a specific Record
      operationId: GetMeta
      tags:
      - Record CRUD
      parameters:
      - name: realmId
        in: path
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: recordId
        in: path
        description: Identifier of the Record
        required: true
        schema:
          type: string
          example: 'UserRecordValue000000001'
      - name: If-None-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.3
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      responses:
        '200':
          description: Expected response to a valid request
          headers:
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/RecordMeta'
        '304':
          $ref: '#/components/responses/304'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
```

```
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    patch: # patch meta data
      summary: Record's meta update
      description: update meta of a specific Record
      operationId: UpdateMeta
      tags:
      - Record CRUD
      parameters:
      - name: realmId
        in: path
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: recordId
        in: path
        description: Identifier of the Record
        required: true
        schema:
          type: string
          example: 'UserRecordValue000000001'
      - name: If-Match
        in: header
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      requestBody:
        description: Meta data to patch
        content:
          application/json-patch+json:
            example: '[{ "op": "replace", "path": "/tags/ueId", "value": "450005" }, { "op":
"remove", "path": "/tags/recordId" }]'
            schema:
              type: array
              items:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchItem'
              minItems: 1
        required: true
      responses:
        '200':
          description: >-
            One or more modification instructions have been discarded, the execution report is
returned in response PatchResult.
          content:
            application/json:
              example:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchResult'
          headers:
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '204':
          description: >-
            Successful case. The meta has been successfully updated and no return is expected.
          headers:
```

```
          Cache-Control:
            $ref: '#/components/headers/Cache-Control'
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '408':
        $ref: 'TS29571_CommonData.yaml#/components/responses/408'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'

/{realmId}/{storageId}/records/{recordId}/blocks:
  summary: Access to the Blocks of a specific Record, identified by its RecordId
  description: >-
    Access to the Blocks of a specific Record
  get:
    summary: Record's Blocks access
    description: retrieve all Blocks of a specific Record
    operationId: GetBlockList
    tags:
    - Block CRUD
    parameters:
    - name: realmId
      in: path
      description: Identifier of the Realm
      required: true
      schema:
        type: string
        example: Realm01
    - name: storageId
      in: path
      description: Identifier of the Storage
      required: true
      schema:
        type: string
        example: Storage01
    - name: recordId
      in: path
      description: Identifier of the Record
      required: true
      schema:
        type: string
        example: 'UserRecordValue000000001'
    - name: supported-features
      in: query
      description: Features required to be supported by the target NF
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        description: Expected response to a successful request
        headers:
          Cache-Control:
            $ref: '#/components/headers/Cache-Control'
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
        content:
          multipart/parallel:
            schema:
              type: object
              properties:
                blocks:
```

```
                  type: array
                  description: >-
                    an array of Block parts, can be empty
                  items:
                    $ref: '#/components/schemas/Block'
            encoding:
              blocks:
                contentType: '*/*' # Block content type can be of any type.
                headers:
                  Content-Id: # Block identifier is defined by the Content-Id header.
                    schema:
                      type: string
                    required: true
                  Content-Transfer-Encoding:
                    schema:
                      type: string
                    required: true
      '204':
        description: Successful response, the record contains no blocks
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'

/{realmId}/{storageId}/records/{recordId}/blocks/{blockId}:
  summary: Access to a Block of a specific Record, identified by its BlockId
  description: >-
    Access to a specific Block of a specific Record
  get:
    summary: Retrieve a specific Block
    description: retrieve a specific Block
    operationId: GetBlock
    tags:
    - Block CRUD
    parameters:
    - name: realmId
      in: path
      description: Identifier of the Realm
      required: true
      schema:
        type: string
        example: Realm01
    - name: storageId
      in: path
      description: Identifier of the Storage
      required: true
      schema:
        type: string
        example: Storage01
    - name: recordId
      in: path
      description: Identifier of the Record
      required: true
      schema:
        type: string
        example: 'UserRecordValue000000001'
    - name: blockId
      in: path
      description: Id of the Block
      required: true
      schema:
        type: string
        example: 'userDefjson01'
    - name: If-None-Match
      in: header
      description: Validator for conditional requests, as described in RFC 7232, 3.2
      schema:
        type: string
```

```
    - name: If-Modified-Since
      in: header
      description: Validator for conditional requests, as described in RFC 7232, 3.3
      schema:
        type: string
    - name: supported-features
      in: query
      description: Features required to be supported by the target NF
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        $ref: '#/components/responses/BlockBody'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  put:
    summary: Create or Update a specific Block in a Record.
    description: Create or update a specific Block, related to a Record
    operationId: CreateOrModifyBlock
    tags:
    - Block CRUD
    parameters:
    - name: realmId
      in: path
      description: Identifier of the Realm
      required: true
      schema:
        type: string
        example: Realm01
    - name: storageId
      in: path
      description: Identifier of the Storage
      required: true
      schema:
        type: string
        example: Storage01
    - name: recordId
      in: path
      description: Identifier of the Record
      required: true
      schema:
        type: string
        example: 'UserRecordValue000000001'
    - name: blockId
      in: path
      description: Id of the Block
      required: true
      schema:
        type: string
        example: 'userDefjson01'
    - name: get-previous
      in: query
      description: Retrieve the Block before update
      required: false
      schema:
        type: boolean
        default: false
    - name: If-None-Match
      in: header
      description: Validator for conditional requests, as described in RFC 7232, 3.2
      schema:
        type: string
    - name: If-Match
      in: header
```

```
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      requestBody:
        description: information on the Block to create
        required: true
        content:
          '*/*':
             schema:
               $ref: '#/components/schemas/Block'
      responses:
        '200':
          $ref: '#/components/responses/BlockBody'
        '201':
          description: >-
            Creation case. The Block has been successfully created. Location header indicates the
URI of the created Block.
          headers:
            Location:
              $ref: '#/components/headers/Location'
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '204':
          description: >-
            Successful case. The resource has been successfully updated.
          headers:
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '408':
          $ref: 'TS29571_CommonData.yaml#/components/responses/408'
        '412': # Return previous Block value if get-previous=true
          $ref: '#/components/responses/BlockBody'
        '413':
          $ref: 'TS29571_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    delete:
      summary: Delete a specific Block. Then update the Record
      description: delete a specific Block, related to a Record
      operationId: DeleteBlock
      tags:
      - Block CRUD
      parameters:
      - name: realmId
        in: path
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
```

```
        required: true
        schema:
          type: string
          example: Storage01
      - name: recordId
        in: path
        description: Identifier of the Record
        required: true
        schema:
          type: string
          example: 'UserRecordValue000000001'
      - name: blockId
        in: path
        description: Id of the Block
        required: true
        schema:
          type: string
          example: 'userDefjson01'
      - name: get-previous
        in: query
        description: Retrieve the Block before delete
        required: false
        schema:
          type: boolean
          default: false
      - name: If-Match
        in: header
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      responses:
        '200':
          $ref: '#/components/responses/BlockBodyDelete'
        '204':
          description: >-
            Successful case. The Block has been successfully deleted.
          headers:
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '408':
          $ref: 'TS29571_CommonData.yaml#/components/responses/408'
        '412': # Return previous Block value if get-previous=true
          $ref: '#/components/responses/BlockBody'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /{realmId}/{storageId}/subs-to-notify:
    summary: The notification subscription collection resource
    description: >-
      Access to the subscription resource
    get:
      summary: Notification subscription retrieval
      description: retrieve all notification subscriptions of the storage
      operationId: GetNotificationSubscriptions
      tags:
      - NotificationSubscriptions CRUD
      parameters:
      - name: realmId
        in: path
```

```
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: limit-range
        in: query
        description: The maximum number of NotificationSubscriptions to fetch
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uinteger'
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      responses:
        '200':
          description: Expected response to a valid request
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/NotificationSubscription'
        '304':
          $ref: '#/components/responses/304'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /{realmId}/{storageId}/subs-to-notify/{subscriptionId}:
    summary: The notification subscription resource
    description: >-
      Access to the subscription resource
    get:
      summary: Notification subscription retrieval
      description: retrieve a single notification subscription of the storage
      operationId: GetNotificationSubscription
      tags:
      - NotificationSubscription CRUD
      parameters:
      - name: realmId
        in: path
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: subscriptionId
        in: path
        description: Identifier of the NotificationSubscription
        required: true
        schema:
```

```
        type: string
        example: Subscription01
    - 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 RFC 7232, 3.2
      schema:
        type: string
    - name: If-Modified-Since
      in: header
      description: Validator for conditional requests, as described in RFC 7232, 3.3
      schema:
        type: string
    responses:
      '200':
        description: Expected response to a valid request
        headers:
          Cache-Control:
            $ref: '#/components/headers/Cache-Control'
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NotificationSubscription'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  delete:
    summary: Delete a Notification Subscription of the storage
    description: delete a single subscriptions of the storage
    operationId: DeleteNotificationSubscription
    tags:
    - NotificationSubscription CRUD
    parameters:
    - name: realmId
      in: path
      description: Identifier of the Realm
      required: true
      schema:
        type: string
        example: Realm01
    - name: storageId
      in: path
      description: Identifier of the Storage
      required: true
      schema:
        type: string
        example: Storage01
    - name: subscriptionId
      in: path
      description: Identifier of the NotificationSubscription
      required: true
      schema:
        type: string
        example: Subscription01
    - name: client-id
      in: query
```

```
      description: Identifies the NF or NFSet
      required: true
      schema:
        $ref: '#/components/schemas/ClientId'
    - name: get-previous
      in: query
      description: Retrieve the NotificationSubscription before delete
      required: false
      schema:
        type: boolean
        default: false
    - name: If-Match
      in: header
      description: Record validator for conditional requests, as described in RFC 7232, 3.2
      schema:
        type: string
    - name: supported-features
      in: query
      description: Features required to be supported by the target NF
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        description: Deleted NotificationSubscription if requested with get-previous
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/NotificationSubscription'
      '204':
        description: >-
          Successful case. The SubscriptionNotification has been successfully deleted.
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '408':
        $ref: 'TS29571_CommonData.yaml#/components/responses/408'
      '412':
        description: Return previous NotificationSubscription value if get-previous=true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NotificationSubscription'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  patch: # patch NotificationSubscription data
    summary: NotificationSubscription update
    description: update a specific NotificationSubscription
    operationId: UpdateNotificationSubscription
    tags:
    - NotificationSubscription CRUD
    parameters:
    - name: realmId
      in: path
      description: Identifier of the Realm
      required: true
      schema:
        type: string
        example: Realm01
    - name: storageId
      in: path
      description: Identifier of the Storage
      required: true
      schema:
        type: string
        example: Storage01
    - name: subscriptionId
```

```
        in: path
        description: Identifier of the NotificationSubscription
        required: true
        schema:
          type: string
          example: Subscription01
      - name: If-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      requestBody:
        description: data to patch
        content:
          application/json-patch+json:
            example: 'TBD'
            schema:
              type: array
              items:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchItem'
              minItems: 1
        required: true
      responses:
        '200':
          description: >-
            One or more modification instructions have been discarded, the execution report is
returned in response PatchResult.
          content:
            application/json:
              example:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchResult'
          headers:
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '204':
          description: >-
            Successful case. The meta has been successfully updated and no return is expected.
          headers:
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '304':
          $ref: '#/components/responses/304'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '408':
          $ref: 'TS29571_CommonData.yaml#/components/responses/408'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

    put:
      summary: NotificationSubscription Create/Update
      operationId: CreateAndUpdateNotificationSubscription
      tags:
        - NotificationSubscription CRUD
```

```
      parameters:
      - name: realmId
        in: path
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: subscriptionId
        in: path
        description: Identifier of the NotificationSubscription
        required: true
        schema:
          type: string
          example: Subscription01
      - 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 RFC 7232, 3.2
        schema:
          type: string
      - name: If-Match
        in: header
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/NotificationSubscription'
        required: true
      responses:
        '200' : # Update
          description: Expected response to a valid update request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NotificationSubscription'
        '201':
          description: Expected response to a valid create request
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/NotificationSubscription'
          headers:
            Location:
              description: 'Contains the URI of the newly created resource according to the
structure: {apiRoot}/nudsf-dr/<apiVersion>/{realmId}/{storageId}/subs-to-notify/{subscriptionId}'
              required: true
              schema:
                type: string
            Cache-Control:
              $ref: '#/components/headers/Cache-Control'
            ETag:
              $ref: '#/components/headers/ETag'
            Last-Modified:
              $ref: '#/components/headers/Last-Modified'
        '304':
          $ref: '#/components/responses/304'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
```

```
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '408':
        $ref: 'TS29571_CommonData.yaml#/components/responses/408'
      '409':
        description: Conflict
        content:
         application/json:
          schema:
            type: array
            items:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
      '412':
        $ref: 'TS29571_CommonData.yaml#/components/responses/412'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        description: Unexpected error
        content:
          application/problem+json:
            schema:
              $ref: 'TS29571_CommonData.yaml#/components/schemas/ProblemDetails'
    callbacks:
      onDataChange:
        '{request.body#/callbackReference}':
         post:
           requestBody:
             $ref: '#/components/requestBodies/RecordNotificationBody'
           responses:
             '204':
               description: Callback executed successfully
             '400':
               $ref: 'TS29571_CommonData.yaml#/components/responses/400'
             '401':
               $ref: 'TS29571_CommonData.yaml#/components/responses/401'
             '403':
               $ref: 'TS29571_CommonData.yaml#/components/responses/403'
             '500':
               $ref: 'TS29571_CommonData.yaml#/components/responses/500'
             '503':
               $ref: 'TS29571_CommonData.yaml#/components/responses/503'
             default:
               $ref: 'TS29571_CommonData.yaml#/components/responses/default'
      subscriptionExpiryNotification:
        '{request.body#/expiryCallbackReference}':
         post:
           requestBody:
             required: true
             content:
               application/json:
                 schema:
                   $ref: '#/components/schemas/NotificationInfo'
           responses:
             '204':
               description: Successful Notification response
             '400':
               $ref: 'TS29571_CommonData.yaml#/components/responses/400'
             '401':
               $ref: 'TS29571_CommonData.yaml#/components/responses/401'
             '403':
               $ref: 'TS29571_CommonData.yaml#/components/responses/403'
             '404':
               $ref: 'TS29571_CommonData.yaml#/components/responses/404'
             '500':
               $ref: 'TS29571_CommonData.yaml#/components/responses/500'
             '503':
               $ref: 'TS29571_CommonData.yaml#/components/responses/503'
             default:
               $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /{realmId}/{storageId}/meta-schemas/{schemaId}:
    summary: Access to a specific Meta Schema, identified by its SchemaId
    description: >-
      Access to a specific Meta Schema
    get:
      summary: Meta Schema access
      description: retrieve one specific Meta Schema
```

```
    operationId: GetMetaSchema
    tags:
    - MetaSchema CRUD
    parameters:
      - name: realmId
        in: path
        description: Identifier of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: schemaId
        in: path
        description: Identifier of the Meta Schema
        required: true
        schema:
          $ref: '#/components/schemas/SchemaId'
      - name: If-None-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: If-Modified-Since
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.3
        schema:
          type: string
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200' : #result ok
        $ref: '#/components/responses/RecordBody'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  put:
    summary: Create/Modify Meta Schema
    description: Create or Modify a Meta Schema with a user provided SchemaId
    operationId: CreateOrModifyMetaSchema
    tags:
      - MetaSchema CRUD
    parameters:
      - name: realmId
        in: path
        description: Identifier(name) of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
```

```
        example: Storage01
      - name: schemaId
        in: path
        description: Identifier of the Meta Schema
        required: true
        schema:
          $ref: '#/components/schemas/SchemaId'
      - name: If-None-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: If-Match
        in: header
        description: Validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: get-previous
        in: query
        description: Retrieve the Meta Schema before update
        required: false
        schema:
          type: boolean
          default: false
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/MetaSchema'
      required: true
    responses:
      '200':
        description: Update with return
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MetaSchema'
      '201':
        description: >-
          Create case. The resource has been successfully created, location header indicates
          the URI of the created Record.
          $ref: '#/components/responses/RecordBody'
        headers:
          Location:
            $ref: '#/components/headers/Location'
          Cache-Control:
            $ref: '#/components/headers/Cache-Control'
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
      '204': # Update without return
        description: >-
          Update case. The resource has been successfully updated and no
          additional content is included in the response message.
        headers:
          Cache-Control:
            $ref: '#/components/headers/Cache-Control'
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '408':
```

```
        $ref: 'TS29571_CommonData.yaml#/components/responses/408'
      '412':
        description: Return Meta Schema value if get-previous=true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MetaSchema'
      '413':
        $ref: 'TS29571_CommonData.yaml#/components/responses/413'
      '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: Delete a Meta Schema with an user provided SchemaId
    operationId: DeleteMetaSchema
    tags:
      - MetaSchema CRUD
    parameters:
      - name: realmId
        in: path
        description: Identifier(name) of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: schemaId
        in: path
        description: Identifier of the Meta Schema
        required: true
        schema:
          $ref: '#/components/schemas/SchemaId'
      - name: If-Match
        in: header
        description: Record validator for conditional requests, as described in RFC 7232, 3.2
        schema:
          type: string
      - name: get-previous
        in: query
        description: Retrieve the Meta Schema before delete
        required: false
        schema:
          type: boolean
          default: false
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MetaSchema'
      '204':
        description: Successful case.
        headers:
          ETag:
            $ref: '#/components/headers/ETag'
          Last-Modified:
            $ref: '#/components/headers/Last-Modified'
      '304':
        $ref: '#/components/responses/304'
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
```

```
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '408':
          $ref: 'TS29571_CommonData.yaml#/components/responses/408'
        '412':
          description: Return value if get-previous=true
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/MetaSchema'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

components:
  securitySchemes:
    oAuth2ClientCredentials:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: '{nrfApiRoot}/oauth2/token'
          scopes:
            nudsf-dr: Access to the nudsf-dr API

  schemas:
    RecordSearchResult:
      description: Count and collection of Record references matching the providing filter.
      type: object
      properties:
        count:      # The total number of elements found.
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uinteger'
        references: # The Record references found. If count-indicator is true, no references are
sent back.
          type: array
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
          minItems: 1
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
        matchingRecords:
          description: A map (list of key-value pairs where recordId serves as key) of Records
          type: object
          additionalProperties:
            $ref: '#/components/schemas/Record'
          minProperties: 1
      required:
        - count
    RecordMeta:
      description: Meta data of a Record
      type: object
      properties:
        ttl:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        callbackReference:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        tags:
          type: object # dictionary type
          description: >-
              A dictionary of {"tagName": [ "tagValue", ...] }. A tag name can be used to retrieve a
Record. The tagValue are unique.
          additionalProperties:
            type: array
            items:
              type: string
            uniqueItems: true
            minItems: 1
          minProperties: 1
          example:  '{"ueId" : [ "455345", "455346" ], "recordId" : [ "1000106" ] }'
      example: >-
        { "tags" : {"ueId" : [ "455345", "455346" ], "recordId" : [ "1000106" ] }}
        schemaId:
```

```
          $ref: '#/components/schemas/SchemaId'

    Record:
      description: Definition of a Record
      type: object
      properties:
        meta:
          # json representation of the Meta Data
          $ref: '#/components/schemas/RecordMeta'
        blocks:
          # List of multipart data
          type: array
          description: list of opaque Block's in this Record
          items:
            $ref: '#/components/schemas/Block'
          minItems: 1
      required:
        - meta
      example: >-
        {"meta": { "tags" : {"tag1" : ["value1"], "tag2" :["value2"] } }, "blocks": [{"Content-Id":
"userDefBinaryBlob", "Content-Type": "text/plain", "content": "QmxvY2sgY29udGVudA=="}, {"Content-
Id": "userDefJsonBlob", "Content-Type": "application/json", "content": "{"key": "ftsimpletype-
999550000000002", "value": "A3E71A78377179B5B91A;imsi-999550000000123"}]}

    RecordIdList:
      description: List of Record IDs
      type: object
      properties:
        recordIdList:
          type: array
          items:
            type: string
          minItems: 1
      required:
        - recordIdList

    Block:
      description: A Block can be of any type
      example: >-
        "QmxvY2sgY29udGVudA=="
    NotificationSubscription:
      description: Definition of a notification subscription
      type: object
      properties:
        clientId:
          $ref: '#/components/schemas/ClientId'
        callbackReference:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        expiryCallbackReference:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        expiry:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        expiryNotification:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uinteger'
        subFilter:
          $ref: '#/components/schemas/SubscriptionFilter'
        supportedFeatures:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      required:
        - clientId
        - callbackReference

    RecordNotification:
      description: Definition of a notification on a record
      type: object
      properties:
        descriptor:
          # json representation of the notification description
          $ref: '#/components/schemas/NotificationDescription'
        meta:
          # json representation of the Meta Data
          $ref: '#/components/schemas/RecordMeta'
        blocks:
          # List of multipart data
          type: array
          description: list of opaque Block's in this Record
          items:
            $ref: '#/components/schemas/Block'
```

```
      required:
        - descriptor
        - meta
      example: >-
        {"descriptor": { "recordRef" : "...", "operationType" : "DELETED"}, "meta": { "tags" :
{"tag1" : ["value1"], "tag2" :["value2"] } }, "blocks": [{"Content-Id": "userDefBinaryBlob",
"Content-Type": "text/plain", "content": "QmxvY2sgY29udGVudA=="}, {"Content-Id": "userDefJsonBlob",
"Content-Type": "application/json", "content": "{"key": "ftsimpletype-999550000000002", "value":
"A3E71A78377179B5B91A;imsi-999550000000123"}]}

    NotificationDescription:
      description: Description of a record notification
      type: object
      properties:
        recordRef:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        operationType:
          $ref: '#/components/schemas/RecordOperation'
        subscriptionId:
          # unique identifier of the NotificationSubscription
          type: string
      required:
        - recordRef
        - operationType
      example: >-
        { "record" : "...", "operationType" : "DELETED"}

    SubscriptionFilter:
      description: A subscription filter
      type: object
      properties:
        monitoredResourceUris:
          type: array
          description: list of resources applicable to the subscription
          items:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
          minItems: 1
        operations:
          type: array
          description: list of resources applicable to the subscription
          items:
            $ref: '#/components/schemas/RecordOperation'
          maxItems: 3

    ClientId:
      description: Defines the identity of the NF Consumer
      type: object
      properties:
        nfId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfInstanceId'
        nfSetId:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NfSetId'

    RecordOperation:
      description: Indicate operation made on a record
      anyOf:
      - type: string
        enum:
          - CREATED
          - UPDATED
          - DELETED
      - type: string

    ConditionOperator:
      description: TBD
      anyOf:
      - type: string
        enum:
          - AND
          - OR
          - NOT
      - type: string

    ComparisonOperator:
      description: TBD
      anyOf:
        - type: string
          enum:
```

```
            # Equals
            - EQ
            # Not Equal
            - NEQ
            # Greater Than
            - GT
            # Greater Than or Equal
            - GTE
            # Less Than
            - LT
            # Less Than or Equal
            - LTE
        - type: string

    SearchExpression:
      description: A logical expression element
      type: object
      oneOf:
        - $ref: '#/components/schemas/SearchCondition'
        - $ref: '#/components/schemas/SearchComparison'
        - $ref: '#/components/schemas/RecordIdList'
      example:
        { "cond": "OR", "units": [ { "op": "EQ", "tag" : "ueId", "value" : "455345" }, { "op": "EQ",
"tag" : "supi", "value" : "imsi-999559807001001" } ] }

    SearchCondition:
      description: A logical condition
      type: object
      properties:
        cond:
          $ref: '#/components/schemas/ConditionOperator'
        units:
          type: array
          items:
            $ref: '#/components/schemas/SearchExpression'
          minItems: 1
        schemaId:
          $ref: '#/components/schemas/SchemaId'
      required:
        - cond
        - units
      example:
        { "cond": "OR", "units": [ { "op": "EQ", "tag" : "ueId", "value" : "455345" }, { "op": "EQ",
"tag" : "supi", "value" : "imsi-999559807001001" } ] }

    SearchComparison:
      description: A comparison to apply on tag/values pairs.
      type: object
      properties:
        op:
          $ref: '#/components/schemas/ComparisonOperator'
        tag:
          type: string
        value:
          type: string
      required:
        - op
        - tag
        - value
      example:
        { "op": "EQ", "tag" : "supi", "value" : "imsi-999559807001001" }

    MetaSchema:
      description: Defines the Meta Schema
      type: object
      required:
        - schemaId
        - metaTags
      properties:
        schemaId:
          $ref: '#/components/schemas/SchemaId'
        metaTags:
          type: array
          items:
            $ref: '#/components/schemas/TagType'

    TagType:
      description: Defines the Tag Type
```

```
      type: object
      required:
        - tagName
        - keyType
      properties:
        tagName:
          type: string
        keyType:
          $ref: '#/components/schemas/KeyType'
        sort:
          type: boolean
          default: false
        presence:
          type: boolean

    SchemaId:
      description: Represents the Identifier of a Meta schema.
      type: string

    KeyType:
      description: Represents the type of a key.
      anyOf:
        - type: string
          enum:
          - UNIQUE_KEY
          - SEARCH_KEY
          - COUNT_KEY
          - SEARCH_AND_COUNT_KEY
          - OTHER_TAG
        - type: string

    RetrieveRecords:
      description: Indicates the data to be retrieved.
      anyOf:
        - type: string
          enum:
          - ONLY_META
          - META_AND_BLOCKS
        - type: string

    NotificationInfo:
      type: object
      required:
        - expiredSubscriptions
      properties:
        expiredSubscriptions:
          type: array
          items:
            $ref: '#/components/schemas/NotificationSubscription'
          minItems: 1

  headers:
    Cache-Control:
      description: Cache-Control containing max-age, as described in RFC 7234, 5.2
      schema:
        type: string
    ETag:
      description: Entity Tag, containing a strong validator, as described in RFC 7232, 2.3
      schema:
        type: string
    Last-Modified:
      description: Timestamp for last modification of the resource, as described in RFC 7232, 2.2
      schema:
        type: string
    Location:
      description: Contains the URI of the newly created resource
      required: true
      schema:
        type: string
    Retry-After:
      description: 'Indicates the time the NF Consumer has to wait before making a new request. It
can be a non-negative integer (decimal number) indicating the number of seconds the NF Consumer has
to wait before making a new request or an HTTP-date after which the AF can retry a new request.'
      schema:
        anyOf:
          - type: integer
          - type: string
```

```
  requestBodies:
    RecordBody:
      description: The record multipart request body. The meta part shall be the first part and is
mandatory but can be empty and zero or more block parts may follow the meta part.
      required: true
      content:
        multipart/mixed:
          schema:
            $ref: '#/components/schemas/Record'
          encoding:
            meta: # The meta part shall be the first part and is mandatory but can be empty
              contentType: application/json
              headers:
                Content-Id:
                  schema:
                    type: string
                  required: true
            blocks: # 0 or more block parts may follow the meta part
              contentType: '*/*' # Block part can be of any type
              headers:
                Content-Id: # Block identifier is defined by the Content-Id header.
                  schema:
                    type: string
                  required: true
                Content-Transfer-Encoding:
                  schema:
                    type: string
                  required: true

    RecordNotificationBody:
      description: The record notification multipart request body. The descriptor part shall be the
first one, followed by record meta part and by zero or more block parts.
      required: true
      content:
        multipart/mixed:
          schema:
            $ref: '#/components/schemas/RecordNotification'
          encoding:
            descriptor: # The descriptor part shall be the first part and is mandatory
              contentType: application/json
              headers:
                Content-Id:
                  schema:
                    type: string
                  required: true
            meta: # The meta part shall be the second part and is mandatory but can be empty
              contentType: application/json
              headers:
                Content-Id:
                  schema:
                    type: string
                  required: true
            blocks: # 0 or more block parts may follow the meta part
              contentType: '*/*' # Block part can be of any type
              headers:
                Content-Id: # Block identifier is defined by the Content-Id header.
                  schema:
                    type: string
                  required: true
                Content-Transfer-Encoding:
                  schema:
                    type: string
                  required: true

  responses:
    '304': # Etag response if the value might differ from that sent
      description: Not Modified
      content:
        application/problem+json:
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/ProblemDetails'
      headers:
        Cache-Control:
          $ref: '#/components/headers/Cache-Control'
        ETag:
          $ref: '#/components/headers/ETag'
        Retry-After:
          $ref: '#/components/headers/Retry-After'
```

```
    'RecordBody': #  Record value with associated headers
      description: >-
         - 200 Update. The resource has been successfully updated and previous value must be sent in
the response message if requested.
          - 200 Get. The resource exists, its value must be sent in the response message
          - 412 Precondition Failed, the previous value must be sent in response message if
requested.
      content:
        multipart/mixed:
          schema:
            $ref: '#/components/schemas/Record'
          encoding:
              meta: # The meta part shall be the first part and is mandatory but can be empty.
                contentType: application/json
                headers:
                  Content-Id: # The meta part is identified by the 'meta' Content-Id header.
                    schema:
                      type: string
                    required: true
              blocks: # Zero or more block parts may follow the meta part
                contentType: '*/*' # Block parts can be of any type.
                headers:
                  Content-Id: # Block identifier is defined by the Content-Id header.
                    schema:
                      type: string
                    required: true
                  Content-Transfer-Encoding:
                    schema:
                      type: string
                    required: true
      headers:
        Cache-Control:
          $ref: '#/components/headers/Cache-Control'
        ETag:
          $ref: '#/components/headers/ETag'
        Last-Modified:
          $ref: '#/components/headers/Last-Modified'

    'RecordBodyDelete': #  Record value with associated headers
      description: >-
         - 200 Delete. The resource has been successfully delete and previous value must be sent in
the response message if requested.
      content:
        multipart/mixed:
          schema:
            $ref: '#/components/schemas/Record'
          encoding:
              meta: # The meta part shall be the first par and is mandatory but can be empty.
                contentType: application/json
                headers:
                  Content-Id: # The meta part is identified by the Content-Id header.
                    schema:
                      type: string
                    required: true
              blocks: # Zero or more block parts may follow the meta part.
                contentType: '*/*' # Block parts can be of any type.
                headers:
                  Content-Id: # Block identifier is defined by the Content-Id header.
                    schema:
                      type: string
                    required: true
                  Content-Transfer-Encoding:
                    schema:
                      type: string
                    required: true
      headers:
        ETag:
          $ref: '#/components/headers/ETag'
        Last-Modified:
          $ref: '#/components/headers/Last-Modified'

    'BlockBody': #  Block value with associated headers
      description: >-
         - 200 Update: The resource has been successfully updated and previous value must be sent in
the response message if requested.
          - 200 Get: The resource exists, its value must be sent in the response message
```

```
          - 412 Precondition Failed: the previous value must be sent in response message if
requested.
      content:
        '*/*':
          schema:
              $ref: '#/components/schemas/Block'
      headers:
        Cache-Control:
          $ref: '#/components/headers/Cache-Control'
        ETag:
          $ref: '#/components/headers/ETag'
        Last-Modified:
          $ref: '#/components/headers/Last-Modified'

    'BlockBodyDelete': #  Block value with associated headers
      description: >-
          - 200 Delete: The resource has been successfully delete and previous value must be sent in
the response message if requested.
      content:
        '*/*':
          schema:
              $ref: '#/components/schemas/Block'
      headers:
        ETag:
          $ref: '#/components/headers/ETag'
        Last-Modified:
          $ref: '#/components/headers/Last-Modified'
```

A.3 Nudsf_Timer API

```
openapi: 3.0.0
info:
  title: Nudsf_Timer
  version: 1.0.0-alpha.3
  description: |
    Nudsf Timer Service.
    © 2022, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).
    All rights reserved.

externalDocs:
  description: 3GPP TS 29.598 UDSF Services, V17.5.0.
  url: 'https://www.3gpp.org/ftp/Specs/archive/29_series/29.598/'

servers:
  - url: '{apiRoot}/nudsf-timer/v1'
    variables:
      apiRoot:
        default: https://example.com
        description: apiRoot as defined in clause 4.4 of 3GPP TS 29.501

security:
  - {}
  - oAuth2ClientCredentials:
    - nudsf-timer

paths:

  /{realmId}/{storageId}/timers:
    summary: Access to all Timers of a Storage
    description: >-
      root of all Timers of a Storage
    get:
      summary: Timers search with get
      description: Retrieve one or multiple TimerIDs based on filter
      operationId: SearchTimer
      tags:
        - Timer Search
      parameters:
        - name: realmId
          in: path
          description: Identifier of the Realm
          required: true
          schema:
            type: string
            example: Realm01
        - name: storageId
          in: path
```

```
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: filter
        in: query
        description: Query filter using conditions on tags
        content:
          application/json:
            schema:
              $ref: 'TS29598_Nudsf_DataRepository.yaml#/components/schemas/SearchExpression'
      - name: expired-filter
        in: query
        description: Used to query for expired timers.
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/NullValue'
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    responses:
      '200':
        description: Successful case. Response contains result of the search.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TimerIdList'
      '204':
        description: >-
          The search condition does not match any Timer.
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '429':
        $ref: 'TS29571_CommonData.yaml#/components/responses/429'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  delete:
    summary: Delete one or multiple timers based on filter
    operationId: DeleteTimers
    tags:
      - Timers Delete
    parameters:
      - name: realmId
        in: path
        description: Identifier(name) of the Realm
        required: true
        schema:
          type: string
          example: Realm01
      - name: storageId
        in: path
        description: Identifier of the Storage
        required: true
        schema:
          type: string
          example: Storage01
      - name: supported-features
        in: query
        description: Features required to be supported by the target NF
        schema:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      - name: filter
        in: query
        description: A filter that determines the set of timers to be deleted
        content:
          application/json:
```

```
            schema:
              $ref: 'TS29598_Nudsf_DataRepository.yaml#/components/schemas/SearchExpression'
        - name: expired-filter
          in: query
          description: Presence indicates that only expired timers are to be deleted.
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/NullValue'
      responses:
        '200':
          description: Successful case. Response contains result of the search.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TimerIdList'
        '204':
          description: Successful case.
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '429':
          $ref: 'TS29571_CommonData.yaml#/components/responses/429'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'

  /{realmId}/{storageId}/timers/{timerId}:
    summary: Access to a specific Timer, identified by its TimerId
    description: >-
      Access to a specific Timer
    put:
      summary: Create/Replace Timer
      description: Create or Modify a Timer with a user provided TimerId
      operationId: CreateOrModifyTimer
      tags:
        - Timer Start
      parameters:
        - name: realmId
          in: path
          description: Identifier(name) of the Realm
          required: true
          schema:
            type: string
            example: Realm01
        - name: storageId
          in: path
          description: Identifier of the Storage
          required: true
          schema:
            type: string
            example: Storage01
        - name: timerId
          in: path
          description: Identifier of the Timer
          required: true
          schema:
            $ref: '#/components/schemas/TimerId'
        - name: supported-features
          in: query
          description: Features required to be supported by the target NF
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Timer'
        required: true
      responses:
        '201':
          description: Timer successfully created
```

```
      '204':
        description: Timer successfully replaced
      '400':
        $ref: 'TS29571_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29571_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29571_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29571_CommonData.yaml#/components/responses/404'
      '500':
        $ref: 'TS29571_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29571_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    callbacks:
      timerExpiry:
        '{request.body#/callbackReference}':
          post:
            requestBody:
              required: true
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/Timer'
            responses:
              '204':
                description: Callback executed successfully
              '400':
                $ref: 'TS29571_CommonData.yaml#/components/responses/400'
              '401':
                $ref: 'TS29571_CommonData.yaml#/components/responses/401'
              '403':
                $ref: 'TS29571_CommonData.yaml#/components/responses/403'
              '500':
                $ref: 'TS29571_CommonData.yaml#/components/responses/500'
              '503':
                $ref: 'TS29571_CommonData.yaml#/components/responses/503'
              default:
                $ref: 'TS29571_CommonData.yaml#/components/responses/default'
  patch:
    summary: Timer modification
    description: update a specific Timer
    operationId: UpdateTimer
    tags:
    - Timer Update
    parameters:
    - name: realmId
      in: path
      description: Identifier of the Realm
      required: true
      schema:
        type: string
        example: Realm01
    - name: storageId
      in: path
      description: Identifier of the Storage
      required: true
      schema:
        type: string
        example: Storage01
    - name: timerId
      in: path
      description: Identifier of the Timer
      required: true
      schema:
        $ref: '#/components/schemas/TimerId'
    - name: supported-features
      in: query
      description: Features required to be supported by the target NF
      schema:
        $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
    requestBody:
      description: Timer data to patch
      content:
        application/json-patch+json:
          schema:
```

```
              type: array
              items:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchItem'
              minItems: 1
        required: true
      responses:
        '200':
          description: >-
            One or more modification instructions have been discarded, the execution report is
returned in response PatchResult.
          content:
            application/json:
              schema:
                $ref: 'TS29571_CommonData.yaml#/components/schemas/PatchResult'
        '204':
          description: >-
            Successful case. The timer has been successfully updated and no return is expected.
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    delete:
      summary: Delete a Timer with an user provided TimerId
      operationId: DeleteTimer
      tags:
        - Timer Stop
      parameters:
        - name: realmId
          in: path
          description: Identifier(name) of the Realm
          required: true
          schema:
            type: string
            example: Realm01
        - name: storageId
          in: path
          description: Identifier of the Storage
          required: true
          schema:
            type: string
            example: Storage01
        - name: timerId
          in: path
          description: Identifier of the Timer
          required: true
          schema:
            $ref: '#/components/schemas/TimerId'
        - name: supported-features
          in: query
          description: Features required to be supported by the target NF
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      responses:
        '204':
          description: Successful case.
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
```

```
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'
    get:
      summary: Timer access
      description: retrieve one specific Timer
      operationId: GetTimer
      tags:
      - Timer Get
      parameters:
        - name: realmId
          in: path
          description: Identifier of the Realm
          required: true
          schema:
            type: string
            example: Realm01
        - name: storageId
          in: path
          description: Identifier of the Storage
          required: true
          schema:
            type: string
            example: Storage01
        - name: timerId
          in: path
          description: Identifier of the Timer
          required: true
          schema:
            $ref: '#/components/schemas/TimerId'
        - name: supported-features
          in: query
          description: Features required to be supported by the target NF
          schema:
            $ref: 'TS29571_CommonData.yaml#/components/schemas/SupportedFeatures'
      responses:
        '200' : #result ok
          description: Successful case.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Timer'
        '400':
          $ref: 'TS29571_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29571_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29571_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29571_CommonData.yaml#/components/responses/404'
        '500':
          $ref: 'TS29571_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29571_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29571_CommonData.yaml#/components/responses/default'


components:
  securitySchemes:
    oAuth2ClientCredentials:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: '{nrfApiRoot}/oauth2/token'
          scopes:
            nudsf-timer: Access to the nudsf-timer API

  schemas:

# COMPLEX TYPES:

    Timer:
      description: Represents a timer.
      type: object
      required:
        - expires
      properties:
        timerId:
          $ref: '#/components/schemas/TimerId'
```

```
        expires:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/DateTime'
        metaTags:
          description: A map (list of key-value pairs where a tagName of type string serves as key)
of tagValue lists
          type: object
          additionalProperties:
            type: array
            items:
              type: string
            minItems: 1
          minProperties: 1
        callbackReference:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uri'
        deleteAfter:
          $ref: 'TS29571_CommonData.yaml#/components/schemas/Uinteger'

    TimerIdList:
      description: Represents a list of timer IDs.
      type: object
      required:
        - timerIds
      properties:
        timerIds:
          type: array
          items:
            $ref: '#/components/schemas/TimerId'
          minItems: 1


# SIMPLE TYPES:

    TimerId:
      description: Represents the identifier of a timer.
      type: string
```

Annex B (informative): Search Examples

The conditional expression is defined by the following Extended Backus-Naur Form (EBNF) [18].

Figure B-1: Search Expression

Search_Expression ::= Search_Comparison | Search_Condition

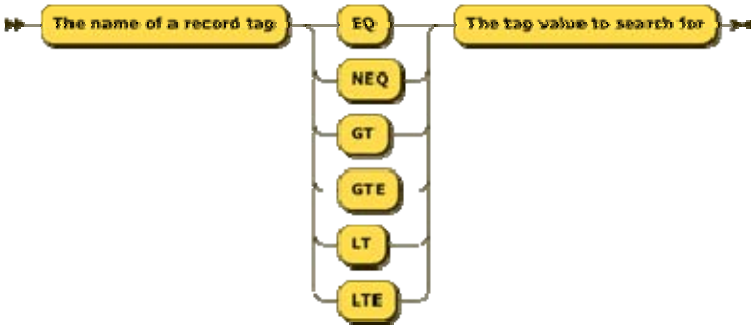

Figure B-2: Search Comparison

Search_Comparison ::= Tag ('EQ' | 'NEQ' | 'GT' | 'GTE' | 'LT' | 'LTE') Value

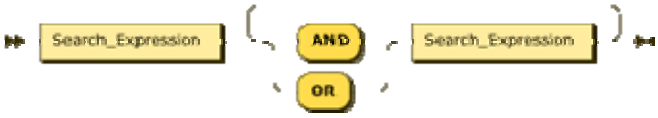

Figure B-3: Search Condition

Search_Condition ::= Search_Expression (('AND' | 'OR') Search_Expression)+

Figure B-4: Search ConditionNot

Search_ConditionNot ::= 'NOT' Search_Expression

Example:

Find all records where the tag "ueId" is equal to "455345" OR the tag "supi" is equal to "imsi-999559807001001":

```
{
  "cond": "OR",
     "items": [
     {
        "op": "EQ",
        "tag" : "ueId",
        "value" : "455345"
     },
```

```
      {
         "op": "EQ",
         "tag" : "supi",
         "value" : "imsi-999559807001001"
      }
   ]
}
```

Annex C (informative): HTTP Multipart Examples

C.1 General

This clause provides examples of the HTTP multipart messages. The examples do not aim to be a complete representation of the HTTP message, e.g. additional information or headers can be included.

This Annex is informative and the normative descriptions in this specification prevail over the description in this Annex if there is any difference.

C.2 Example HTTP multipart Record

```
Content-Type: multipart/mixed; boundary=partboundary

--partboundary
Content-Id: meta
Content-Type: application/json; charset=UTF-8

{"tags":  { "ueId" : ["455345"], "supi" : ["imsi-999559807001001"] } }
--partboundary

Content-Id: block1
Content-Type: application/json; charset=UTF-8
Content-Transfer-Encoding: binary

{ "firstName": "John", "lastName": "Doe"}
--partboundary

Content-Id: block2
Content-Type: image/png
Content-Transfer-Encoding: binary

<binary representation of png>

--partboundary--'
```

C.3 Example HTTP multipart BlockCollection

```
Content-Type: multipart/parallel; boundary=partboundary

--partboundary
Content-Id: block1
Content-Type: application/json; charset=UTF-8
Content-Transfer-Encoding: binary

{ "firstName": "John", "lastName": "Doe"}
--partboundary

Content-Id: block2
Content-Type: image/png
Content-Transfer-Encoding: base64

iVBORw0KGgoAAAANSUhEUgAAADIAAAAeCAYAAABuUU38AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMA
AA7DAcdvqGQAAAAhdEVYdENyZWF0aW9uIFRpbWUAMjAyMDowMjoyMiAxMTowMToxNeLKX7MAAAUTSURBVFhH7Zh5qFVVFIdv2XsN
2qRZZoWUPWnSbHrEwwYapMQKbKCRNJ+GmpEN9EdBhCIVRHMREYKWIWqWYTQpKZZmEZTaHPIipcGyySzN9Pvuuys2h3PvPcW70B/v
Bx/33DPss/faa6+19ulRKqbdIO7dUfn9X2mXym+e9oOx0FqhJ2yE9+BZeB3+hkZpfzgZjoXe8Bu8C8thGxTSQfAlaP1qzIFm6Epp
rKvgDdgCvmczfAu/VP47mMOhkKZCdPgPeAgeh58q54I7INQCl8PNMAHOhl5QRN53G3wPtrsG7oZTYS/QtTXa8TALvoIBUFcXg26z
HSZ6oqKTwPMxkBUwGF6F9HzwKzwIh0ItXQve/z6cAbVcXt0KK6HQGh8EWjnVCEg7/DP8mfyvxu9wL+wLQ+AyuB3uhydgBnjf86Cl
6w3EGVoLo8r//oWa4DpwwWU7WRQH7dpbl/z/Gj6Fz8DZ87xesBiOg1rSCE91HtYfeR+4AG6BtOG/QKukshMfwhL4HFykPq+FD4EX
4RnQfTbApeBsRuSz/Q9ANz0GDoCzYBXkaQF8A67HqroQjFpZtzFijIYfk3OiVc+EeoZRr4CulKcX4CVwgc8F3SdvHbhW7ZvvrKpd
YROkHRUfvBMeSM6JFjbnFNVsMPLkaTgYdp3JPeATuAhSmdOciaehruF0pfWwFdJOZzFB6jbKRn1uHjwHdjiLIdQw7j150vodcH35
X6k0GeaDa1TX89kf4C4oWpWUZ6YfXAFvQl54nQQh/TrvnkBL+/L7wKogTyZEr30Hrqd3QE94ufJ/HDhb/1kOygSVdszFnCY8F6eW
znNLMck5a1PAHJAnA0h/cDA3gm71FnSptJb5IDpmhMqTITYdQPAk6CK6n9Gplk4EQ7MRy2RZV1q6qE6B3TsPy3KK8553sHlaBHuD
s3g0uI6sq0y6DjBkFWDUdGaOANdWIe0D+q3rYCacA2k0MBvfAFG0Ba6HMZDVCRDXrVI91t3CCA7eBe1AroGHwTVg9DP/eM2c48y5
Pk6DQrIgzHbQSGEJoRWz2Xw1RPHoArYui4HbScOi16aB1zw2J6RyDRSxtG3Zl0KyAItO1sKK0wJSNxgJkSwd+NtgxeyseS6SY09w
Fty7WFimsrxY2HlYUwPBoFKo0jUkPgbWOGnHxdlw2q+E7N7jdIi6KcUk1sKoe7OJOJITzR7zm7qrdZsbpCJ6BB7tPKyutHEXnbWN
a8b18BG4G9R9qsns64KNUmEpzKLTJtIeZMvz+R3ElK05sFR6jZc5UGV21m2yM5Wng8EodxS4YzR66gFutv5ROpAuk9Znh9TO7wBe
```

```
sAU6YCGJRgMpt7F6gTNdRBpLo1oh94UvQJdrrJiJVlJzCwNphiYSQh+KI0Npw9SQGcF0TYeVSuNpvI2B7IClMBsfabwVu9WtbuUq
NifngaEtFqP54Saw9DavnAtuZ61ELegMf8PgEjCxGdvdtxjvDZVmdhOsagNrLjdg5gJzju1eDR/DdLC0sdr1u5glUPTHfnjdPT6p
qFxIukfyA13sFO+BVTZgNeqHAHdjIW+0c35jsiQxETloPxG5c/S6+4+h4PV2sGGCVfmFbrJCVrzeo0GsBuyIstx3gO7Ll4ER1GfT
/rgbdLAOzpLHax67wXLvb7s+48eQsjWsiXxhKl/sww7AjvsiLRR1j416j4qvgd7vueisims+K1HqxD7f99q2WGln+xPviOeVA1D2
jXqs1LwTlVGQ4+/C82oAAAAASUVORK5CYII=
--partboundary--'
```

C.4 Example HTTP multipart RecordNotification

```
Content-Type: multipart/mixed; boundary=partboundary

--partboundary
Content-Id: recordnotification
Content-Type: application/json; charset=UTF-8

{"recordRef" : "...", "operationType" : "DELETED" }

--partboundary

Content-Id: meta
Content-Type: application/json; charset=UTF-8

{"tags":  { "ueId" : ["455345"], "supi" : ["imsi-999559807001001"] } }

--partboundary

Content-Id: block1
Content-Type: application/json; charset=UTF-8
Content-Transfer-Encoding: binary

{"firstName": "John", "lastName": "Doe"}

--partboundary

Content-Id: block2
Content-Type: image/png
Content-Transfer-Encoding: binary

<binary representation of png>

--partboundary--'
```

Annex D (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2019-11	CT4#94	C4-195489				Initial Draft.	0.1.0
2020-03	CT4#96	C4-200359 C4-200360 C4-200584 C4-200585 C4-200586 C4-200654 C4-200905 C4-200920 C4-201226 C4-201230 C4-201235 C4-201236 C4-201237				Implementation of pCRs agreed at CT4#96.	0.2.0
2020-03	CT#87e	CP-200066				TS presented for information and approval	1.0.0
2020-03	CT#87e					Approved at CT#87e	16.0.0
2020-06	CT#88e	CP-201175	0001	4	B	Subscribe To Notify	16.1.0
2020-06	CT#88e	CP-201041	0002		C	Removal of Editor's Note	16.1.0
2020-06	CT#88e	CP-201041	0003		C	Add supportedFeatures to RecordSearchResult	16.1.0
2020-06	CT#88e	CP-201041	0004		F	SearchExpression	16.1.0
2020-06	CT#88e	CP-201041	0006	1	F	Miscellaneous Corrections	16.1.0
2020-06	CT#88e	CP-201041	0007	1	B	Storage of YAML files in ETSI Forge	16.1.0
2020-06	CT#88e	CP-201041	0008		F	204 missing in OpenAPI	16.1.0
2020-06	CT#88e	CP-201073	0009		F	Rel-16 API version and External doc update	16.1.0
2020-09	CT#89e	CP-202116	0010		F	Optionality of ProblemDetails in TS29.598 cleanup	16.2.0
2020-12	CT#90e	CP-203052	0011	1	F	Misc corrections	16.3.0
2020-12	CT#90e	CP-203035	0012		F	Removal of the reference to ETSI Forge	16.3.0
2020-12	CT#90e	CP-203052	0013	1	F	Corrections on Subscription and Notification	16.3.0
2020-12	CT#90e	CP-203052	0014	1	F	Corrections on yaml of Nudsf_DataRepository OpenAPI	16.3.0
2020-12	CT#90e	CP-203052	0015	1	F	Define Unsubscription to notifications serivce operation	16.3.0
2020-12	CT#90e	CP-203052	0016	1	F	Incorrect data type	16.3.0
2020-12	CT#90e	CP-203052	0018		F	Resource URI problems clean up	16.3.0
2020-12	CT#90e	CP-203018	0019	3	B	Meta Schema	17.0.0
2020-12	CT#90e	CP-203055	0022		F	29.598 Rel-17 API version and External doc update	17.0.0
2021-03	CT#91e	CP-210039	0024	1	A	Removal of paging parameters	17.1.0
2021-03	CT#91e	CP-210039	0026	1	A	BlockId clarification	17.1.0
2021-03	CT#91e	CP-210034	0027	1	F	OpenAPI Reference	17.1.0
2021-03	CT#91e	CP-210090	0028	2	B	Nudsf_Timer Service	17.1.0
2021-03	CT#91e	CP-210029	0029		F	29.598 Rel-17 API version and External doc update	17.1.0
2021-06	CT#92e	CP-211028	0031		F	metaTags cardinality in type Timer	17.2.0
2021-06	CT#92e	CP-211028	0032		F	Adding some missing description fields to data type definitions in OpenAPI specification files of the Nudsf_DataRepository API	17.2.0
2021-06	CT#92e	CP-211028	0033		F	Adding some missing description fields to data type definitions in OpenAPI specification files of the Nudsf_Timer API	17.2.0
2021-06	CT#92e	CP-211065	0035		A	Nested cardinality R17	17.2.0
2021-06	CT#92e	CP-211065	0037	1	A	Monitored Resource URI	17.2.0
2021-06	CT#92e	CP-211061	0039		A	Content-ID	17.2.0
2021-06	CT#92e	CP-211050	0040		F	29.598 Rel-17 API version and External doc update	17.2.0
2021-09	CT#93e	CP-212026	0042		B	Combined Search and Retrieval	17.3.0
2021-09	CT#93e	CP-212067	0044		A	Removal of enum	17.3.0
2021-09	CT#93e	CP-212059	0045		F	29.598 Rel-17 API version and External doc update	17.3.0
2021-12	CT#94e	CP-213087	0047	2	B	Subscription Expiry Notification	17.4.0
2021-12	CT#94e	CP-213086	0048	1	B	Support of Bulk Operations	17.4.0
2021-12	CT#94e	CP-213133	0050	2	F	Content-id	17.4.0
2021-12	CT#94e	CP-213121	0051		F	29.598 Rel-17 API version and External doc update	17.4.0
2022-03	CT#95e	CP-220024	0052		F	Query parameter formatting	17.5.0
2022-03	CT#95e	CP-220066	0053		F	29.598 Rel-17 API version and External doc update	17.5.0

History

Document history		
V17.5.0	May 2022	Publication