**3GPP TSG-CT WG4 Meeting #107-eC4-216xxx**

**E-Meeting, 15th – 23rd November 2021 *Was C4-216365***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **29.598** | **CR** | **0047** | **rev** | **2** | **Current version:** | **17.3.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Subscription Expiry Notification | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Hewlett Packard Enterprise, Vodafone, Samsung | | | | | | | | | |
| ***Source to TSG:*** | CT4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SBIProtoc17 | | | | |  | ***Date:*** | | | 2021-11-04 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **B** |  | | | | | ***Release:*** | | | Rel-17 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The subs-to-notify resource, provides an option to set an "expiry" time indicating a time when the subscription expires. An NF Consumer can subsequently extend (using e.g. the PUT or the PATCH operation) to update the expiry time or create a new subscription if need be.  Given that many NFs in the 5GC are stateless, maintaning local timers is not practical and using e.g. an external service such as the Timer service of the UDSF introduces additional signalling and overhead.  An NF Consumer that subscribes to be notified of a data change from the UDSF would subsequently have to also create a Timer in the UDSF to be notified about the UDSF timeout so that the expiry time can be updated (if needed) creating a lot of messaging overhead. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | As an optimization, to allow the NF Consumer to forgo using the UDSF Timer service explicitly to be notified about subscripiton expiry, an option to request to be notified about subscription expiry is added to the subs-to-notify resource.  The NF Service Consumer may indicate an offset, i.e. a time in seconds to be notified about the expiry before it expires so that the NF Service Consumer has time to extend the subscription expiry before it expires.  A new notification to convey subscription expiry is added to the subs-to-notify resource. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Expiry timer management must be implemented and supported in each NF that wish to subscribe to be notified of data change which may result in large number of stale or orphaned subscriptions that must be maintained by the UDSF. Alternatively the NF Consumer can rely on the UDSF Timer service, which will result much additional signalling overhead. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.2.2.6.1, 5.2.2.6.X1 (new), 6.1.5.X2 (new), 6.1.6.1, 6.1.6.2.10, 6.1.6.2.Y1 (new), A.2 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | | This CR introduces backward compatible changes to the Nudsf\_DataRepository API OpenAPI - TS29598\_Nudsf\_DataRepository.yaml | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

**\*\*\*\*\*\*\***

\* \* \* First Change \* \* \* \*

##### 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

\* \* \* Next Change \* \* \* \*

##### 5.2.2.6.X1 Subscription Expiry Notification

Figure 5.2.2.6.X1-1 shows a scenario where the UDSF notifies the NF service consumer of an expired subscription to data change.



Figure 5.2.2.6.X1-1: 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.

\* \* \* Next Change \* \* \* \*

#### 6.1.5.X2 Subscription Expiry Notification

##### 6.1.5.X2.1 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.X2.2 Target URI

The Callback URI **"{expiryCallbackReference}"** shall be used with the callback URI variables defined in table 6.1.5.X2.2-1.

Table 6.1.5.X2.2-1: Callback URI variables for this resource

|  |  |
| --- | --- |
| Name | Definition |
| expiryCallbackReference | String formatted as URI with the Callback URI |

##### 6.1.5.X2.3 Standard Methods

###### 6.1.5.X2.3.1 POST

This method shall support the request data structures specified in table 6.1.5.X2.3.1-1 and the response data structures and response codes specified in table 6.1.5.X2.3.1-1.

Table 6.1.5.X2.3.1-2: 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.X2.3.1-3: 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. | | | | |

\* \* \* Next Change \* \* \* \*

#### 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 Nudsf 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: Nudsf 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 | |
| NotificationInfo | | 6.1.6.2.Y1 | | 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 Nudsf 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 Nudsf service based interface.

Table 6.1.6.1-2: Nudsf 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] |  |  |

\* \* \* Next Change \* \* \* \*

##### 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 |  |

\* \* \* Next Change \* \* \* \*

##### 6.1.6.2.Y1 Type: NotificationInfo

Table 6.1.6.2.Y1-1: 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. |  |

\* \* \* Next Change \* \* \* \*

## A.2 Nudsf\_DataRepository API

openapi: 3.0.0

info:

title: Nudsf\_DataRepository

version: 1.1.0-alpha.4

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, V17.3.0.

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'

/{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"}]}

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'

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'

\* \* \* End of Changes \* \* \* \*