**3GPP TSG-CT WG4 Meeting #106-eC4-215**

**E-Meeting, 11th – 15th October 2021 revision of C4-215276**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **29.598** | **CR** | **0048** | **rev** | **1** | **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:*** | Support of Bulk Operations | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Nokia, Nokia Shanghai Bell | | | | | | | | | |
| ***Source to TSG:*** | CT4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SBIProtoc17 | | | | |  | ***Date:*** | | | 2021-10-12 |
|  |  | | | |  | |  | | |  |
| ***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:*** | | UDSF consumers may need to perform retrieval or deletion of a large number of records. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Add protocol support allowing to retrieve or delete a large number of records. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | UDSF consumers cannot request bulk record retrieval/deletion. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.2.2.2.6, 5.2.2.5.1, 5.2.2.5.5 (new), 6.1.3.1, 6.1.3.2.3.2 (new), 6.1.6.1, 6.1.6.2.x (new), 6.1.6.4.1, 6.1.8, 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 additions to the following API:  - TS29598\_Nudsf\_DataRepository.yaml | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

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

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

##### 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.

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

##### 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

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

##### 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.

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

#### 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 |

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

###### 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. | | | | |

\* \* \* 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 | |
| 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 | |
| RecordIdList | | 6.1.6.2.x | | List of Record IDs | | BulkOperations | |

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.x Type: RecordIdList

Table 6.1.6.2.x-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 |

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

##### 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 |

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

## 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. |
| y | 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. |

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

## A.2 Nudsf\_DataRepository API

openapi: 3.0.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*text not shown for clarity\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*text not shown for clarity\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*text not shown for clarity\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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" } ] }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*text not shown for clarity\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \* \* End Of Change \* \* \* \*