**3GPP TSG-CT WG4 Meeting #107-bis-eC4-220xyz**

**E-Meeting, 17th – 21st January 2022 (was C4-220200)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **29.571** | **CR** | **0326** | **rev** | **1** | **Current version:** | **17.4.0** |  |
|  | | | | | | | | |
| *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:*** | Fqdn data type definition | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Ericsson | | | | | | | | | |
| ***Source to TSG:*** | CT4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SBIProtoc17 | | | | |  | ***Date:*** | | | 2022-01-10 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***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:*** | | Several data types are meant to convey FQDNs, but their type definition is simply stated as "string".  This is not optimal, since having a more strict check on the correctness of the information sent on the different APIs, can help to avoid faults and inter-op issues on the field.  There is an Fqdn data type defined in TS 29.510 (defined simply as "string"), but this shoud not preclude to define it as a common data type in TS 29.571, as long as the current definition in TS 29.510 is kept, and then the different APIs can be progressively updated to start making use of the common data type in TS 29.571 and stop using the data type in TS 29.510 (once there are no APIs making use of the definition in TS 29.510, it will be removed).  Regarding the regex pattern for the FQDN, there is currently a pattern for DiameterIdentity (which is an FQDN), defined in clause 5.2.2, as:  **pattern: '^([A-Za-z0-9]+([-A-Za-z0-9]+)\.)+[a-z]{2,}$'**  This pattern applies the following constraints:   * There is a sequence of labels separated by "." * Each label (except last one) consists of letters (upper-case or lower-case), digits and "-", but they cannot start with "-" (The pattern misses to check that each label cannot end with "-", see RFC 1035 and TS 23.003 clause 19.4.2) * The last label allows only 2 or more lower-case letters (The pattern incorrectly requires lower-case, while upper-case is allowed for any label; all labels are case insensitive)   A more complete and strict check can be defined as:  **pattern: '^([0-9A-Za-z]([-0-9A-Za-z]{0,61}[0-9A-Za-z])?\.)+[A-Za-z]{2,63}\.?$'**  **minLength: 4**  **maxLength: 253**  This construct applies similar constraints as before, and in addition it ensures that:   * The total length cannot exceed 253 chars * Each label cannot exceed a length of 63 chars * Each label cannot start or end with "-" * The last label can consist of upper-case or lower-case letters * The FQDN can end with "."   Note that the pattern (similarly as with the original pattern used in DiameterIdentity) does not allow a single-label hostname since that's not a valid Fully-Qualified Domain Name.  It should be noted that the main benefit of defining such constraints is to formally define which FQDN values (strings) are allowed, or not, by the API specification, but implementations are not necessarily forced to implement a regex pattern check, by using a regex engine, if they can rely on the validation of FQDNs by other means (e.g. by services provided by the underlying platform). | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | - Define Fqdn as a Common Data type  - Use it in this spec for those data types that should contain FQDNs | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The API descriptions are vague in which values are allowed or not as data paratemers in signaling messages, leading to inter-operability issues. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.2.1A, 5.2.2, 5.2.4.1, 5.4.2, 5.4.4.69, 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 backwards-compatible corrections with impact on the following APIs:  - TS29571\_CommonData.yaml | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | | Rev. 1: Change CR category to F | | | | | | | | |

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

### 5.2.1A Re-used Data Types

This clause specifies the re-used data types from other specifications.

Table 5.2.1A-1: Re-used Data Types

|  |  |  |
| --- | --- | --- |
| Data Type | Reference | Comments |
| NFType | 3GPP TS 29.510 [29] |  |
| ServiceName | 3GPP TS 29.510 [29] |  |
| DataSetId | 3GPP TS 29.510 [29] |  |
| PlmnSnssai | 3GPP TS 29.510 [29] |  |

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

### 5.2.2 Simple Data Types

This clause specifies common simple data types.

Table 5.2.2-1: Simple Data Types

|  |  |  |
| --- | --- | --- |
| Type Name | Type Definition | Description |
| Binary | string | String with format "binary" as defined in OpenAPI Specification [3] |
| BinaryRm | string | This data type is defined in the same way as the "Binary" data type, but with the OpenAPI "nullable: true" property. |
| Bytes | string | String with format "byte" as defined in OpenAPI Specification [3], i.e, base64-encoded characters, |
| BytesRm | string | This data type is defined in the same way as the "Bytes" data type, but with the OpenAPI "nullable: true" property. |
| Date | string | String with format "date" as defined in OpenAPI Specification [3] |
| DateRm | string | This data type is defined in the same way as the "Date" data type, but with the OpenAPI "nullable: true" property. |
| DateTime | string | String with format "date-time" as defined in OpenAPI Specification [3] |
| DateTimeRm | string | This data type is defined in the same way as the "DateTime" data type, but with the OpenAPI "nullable: true" property. |
| DiameterIdentity | Fqdn | String containing a Diameter Identity (FQDN), according to clause 4.3 of IETF RFC 6733 [18].  DiameterIdentity is defined as a simple data type because Fqdn is also a simple data type (string). |
| DiameterIdentityRm | FqdnRm | This data type is defined in the same way as the "DiameterIdentity" data type, but with the OpenAPI "nullable: true" property.  DiameterIdentityRm is defined as a simple data type because FqdnRm is also a simple data type (either a string or null). |
| Double | number | Number with format "double" as defined in OpenAPI Specification [3] |
| DoubleRm | number | This data type is defined in the same way as the "Double" data type, but with the OpenAPI "nullable: true" property. |
| DurationSec | integer | Unsigned integer identifying a period of time in units of seconds. |
| DurationSecRm | integer | This data type is defined in the same way as the "DurationSec" data type, but with the OpenAPI "nullable: true" property. |
| Float | number | Number with format "float" as defined in OpenAPI Specification [3] |
| FloatRm | number | This data type is defined in the same way as the "Float" data type, but with the OpenAPI "nullable: true" property. |
| Uint16 | integer | Integer where the allowed values correspond to the value range of an unsigned 16-bit integer, i.e. 0 to 65535.  Minimum = 0. Maximum = 65535. |
| Uint16Rm | integer | This data type is defined in the same way as the "Uint16" data type, but with the OpenAPI "nullable: true" property. |
| Int32 | integer | Integer with format "int32" as defined in OpenAPI Specification [3] |
| Int32Rm | integer | This data type is defined in the same way as the "Int32" data type, but with the OpenAPI "nullable: true" property. |
| Int64 | integer | Integer with format "int64" as defined in OpenAPI Specification [3] |
| Int64Rm | integer | This data type is defined in the same way as the "Int64" data type, but with the OpenAPI "nullable: true" property. |
| Ipv4Addr | string | String identifying a IPv4 address formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [4].  Pattern: '^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])$' |
| Ipv4AddrRm | string | This data type is defined in the same way as the "Ipv4Addr" data type, but with the OpenAPI "nullable: true" property. |
| Ipv4AddrMask | string | String identifying a IPv4 address mask formatted in the "dotted decimal" notation as defined in IETF RFC 1166 [4].  Pattern: '^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(\/([0-9]|[1-2][0-9]|3[0-2]))$' |
| Ipv4AddrMaskRm | string | This data type is defined in the same way as the "Ipv4AddrMask" data type, but with the OpenAPI "nullable: true" property. |
| Ipv6Addr | string | String identifying an IPv6 address formatted according to clause 4 of IETF RFC 5952 [5]. The mixed IPv4 IPv6 notation according to clause 5 of IETF RFC 5952 [5] shall not be used.  Pattern: '^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))$'  and  Pattern: '^((([^:]+:){7}([^:]+))|((([^:]+:)\*[^:]+)?::(([^:]+:)\*[^:]+)?))$' |
| Ipv6AddrRm | string | This data type is defined in the same way as the "Ipv6Addr" data type, but with the OpenAPI "nullable: true" property. |
| Ipv6Prefix | string | String identifying an IPv6 address prefix formatted according to clause 4 of IETF RFC 5952 [5]. IPv6Prefix data type may contain an individual /128 IPv6 address.  Pattern: '^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))(\/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))$'  and  Pattern: '^((([^:]+:){7}([^:]+))|((([^:]+:)\*[^:]+)?::(([^:]+:)\*[^:]+)?))(\/.+)$' |
| Ipv6PrefixRm | string | This data type is defined in the same way as the "Ipv6Prefix" data type, but with the OpenAPI "nullable: true" property. |
| MacAddr48 | string | String identifying a MAC address formatted in the hexadecimal notation according to clause 1.1 and clause 2.1 of IETF RFC 7042 [17].  Pattern: '^([0-9a-fA-F]{2})((-[0-9a-fA-F]{2}){5})$' |
| MacAddr48Rm | string | This data type is defined in the same way as the "MacAddr48" data type, but with the OpenAPI "nullable: true" property. |
| SupportedFeatures | string | A string used to indicate the features supported by an API that is used as defined in clause 6.6 in 3GPP TS 29.500 [25]. The string shall contain a bitmask indicating supported features in hexadecimal representation:  Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent the support of 4 features as described in table 5.2.2-3. The most significant character representing the highest-numbered features shall appear first in the string, and the character representing features 1 to 4 shall appear last in the string. The list of features and their numbering (starting with 1) are defined separately for each API. If the string contains a lower number of characters than there are defined features for an API, all features that would be represented by characters that are not present in the string are not supported. |
| Uinteger | integer | Unsigned Integer, i.e. only value 0 and integers above 0 are permissible.  Minimum = 0. |
| UintegerRm | integer | This data type is defined in the same way as the "Uinteger" data type, but with the OpenAPI "nullable: true" property. |
| Uint16 | integer | Integer where the allowed values correspond to the value range of an unsigned 16-bit integer, i.e. 0 to 65535.  Minimum = 0. Maximum = 65535. |
| Uint16Rm | integer | This data type is defined in the same way as the "UInt32" data type, but with the OpenAPI "nullable: true" property. |
| Uint32 | integer | Integer where the allowed values correspond to the value range of an unsigned 32-bit integer, i.e. 0 to (2^32)-1.  Minimum = 0. Maximum = 4294967295. |
| Uint32Rm | integer | This data type is defined in the same way as the "UInt32" data type, but with the OpenAPI "nullable: true" property. |
| Uint64 | integer | Integer where the allowed values correspond to the value range of an unsigned 64-bit integer, i.e. 0 to (2^64)-1.  Minimum = 0. Maximum = 18446744073709551615. |
| Uint64Rm | integer | This data type is defined in the same way as the "Uint64" data type, but with the OpenAPI "nullable: true" property. |
| Uri | string | String providing an URI formatted according to IETF RFC 3986 [6]. |
| UriRm | string | This data type is defined in the same way as the "Uri" data type, but with the OpenAPI "nullable: true" property. |
| VarUeId | string | String represents the SUPI or GPSI.  Pattern: "^(imsi-[0-9]{5,15}|nai-.+|msisdn-[0-9]{5,15}|extid-[^@]+@[^@]+|gci-.+|gli-.+|.+)$". |
| VarUeIdRm | string | This data type is defined in the same way as the "VarUeId" data type, but with the OpenAPI "nullable: true" property. |
| TimeZone | string | String with format "<time-numoffset>" optionally appended by "<daylightSavingTime>", where:  - <time-numoffset> shall represent the time zone adjusted for daylight saving time and be encoded as time-numoffset as defined in clause 5.6 of IETF RFC 3339 [10];  - <daylightSavingTime> shall represent the adjustment that has been made and shall be encoded as "+1" or "+2" for a +1 or +2 hours adjustment.  Example: "-08:00+1" (for 8 hours behind UTC, +1 hour adjustment for Daylight Saving Time). |
| TimeZoneRm | string | This data type is defined in the same way as the "TimeZone" data type, but with the OpenAPI "nullable: true" property. |
| StnSr | string | String representing the STN-SR as defined in clause 18.6 of 3GPP TS 23.003 [7]. |
| StnSrRm | string | This data type is defined in the same way as the "StnSr" data type, but with the OpenAPI "nullable: true" property. |
| CMsisdn | string | String representing the C-MSISDN as defined in clause 18.7 of 3GPP TS 23.003 [7]).  Pattern: "^[0-9]{5,15}$". |
| CMsisdnRm | string | This data type is defined in the same way as the "CMsisdn" data type, but with the OpenAPI "nullable: true" property. |
| DayOfWeek | integer | Integer between and including 1 and 7 denoting a weekday. "1" shall indicate "Monday", and the subsequent weekdays shall be indicated with the next higher numbers. "7" shall indicate "Sunday". |
| TimeOfDay | string | String with format "partial-time" or "full-time" as defined in clause 5.6 of IETF RFC 3339 [10].  Examples: "20:15:00", "20:15:00-08:00" (for 8 hours behind UTC). |
| EmptyObject | object | Empty JSON object: { }  It is defined with the keyword: "additionalProperties: false". |
| Fqdn | string | Fully Qualified Domain Name  Pattern: '^([0-9A-Za-z]([-0-9A-Za-z]{0,61}[0-9A-Za-z])?\.)+[A-Za-z]{2,63}\.?$'  minLength: 4  maxLength: 253 |
| FqdnRm | string | This data type is defined in the same way as the "Fqdn" data type, but it also allows the null value. |

Table 5.2.2-2: Reused OpenAPI data types

|  |  |
| --- | --- |
| Type Name | Description |
| boolean | As defined in OpenAPI Specification [3] |
| integer | As defined in OpenAPI Specification [3] |
| number | As defined in OpenAPI Specification [3] |
| string | As defined in OpenAPI Specification [3] |
| object | As defined in OpenAPI Specification [3] |
| array | As defined in OpenAPI Specification [3] |
| NOTE Data types defined in OpenAPI Specification [3] do not follow the UpperCamel convention for data types in 3GPP TS 29.501 [2] | |

Table 5.2.2-3: Meaning of a Hexadecimal Character in SupportedFeatures Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Character | Feature n+3 supported | Feature n+2 supported | Feature n+1 supported | Feature n supported |
| "0" | no | no | no | no |
| "1" | no | no | no | yes |
| "2" | no | no | yes | no |
| "3" | no | no | yes | yes |
| "4" | no | yes | no | no |
| "5" | no | yes | no | yes |
| "6" | no | yes | yes | no |
| "7" | no | yes | yes | yes |
| "8" | yes | no | no | no |
| "9" | yes | no | no | yes |
| "A" | yes | no | yes | no |
| "B" | yes | no | yes | yes |
| "C" | yes | yes | no | no |
| "D" | yes | yes | no | yes |
| "E" | yes | yes | yes | no |
| "F" | yes | yes | yes | yes |
| NOTE 1 "n" shall be i \* 4 + 1, where "i" is zero or a natural number, i.e permissible values of "n" are 1, 5, 9, …  NOTE 2 If a feature is not defined, it shall be indicated with value "no". | | | | |

For example, if only the first feature defined in the feature list is set to 1, the corresponding SupportedFeatures attribute would have a value of "1", or "001" (any amount of 0's to the left of the 1 would result into an equivalent feature list). If we have 32 features defined, and only the last feature in a feature list is set to 1, the corresponding SupportedFeatures attribute would have a value of "80000000".

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

#### 5.2.4.1 Type: ProblemDetails

Table 5.2.4.1-1: Definition of type ProblemDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| type | Uri | O | 0..1 | A URI reference according to IETF RFC 3986 [6] that identifies the problem type. |
| title | string | O | 0..1 | A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem. |
| status | integer | O | 0..1 | The HTTP status code for this occurrence of the problem. |
| detail | string | O | 0..1 | A human-readable explanation specific to this occurrence of the problem. |
| instance | Uri | O | 0..1 | A URI reference that identifies the specific occurrence of the problem. |
| cause | string | C | 0..1 | A machine-readable application error cause specific to this occurrence of the problem  This IE should be present and provide application-related error information, if available. |
| invalidParams | array(InvalidParam) | O | 1..N | Description of invalid parameters, for a request rejected due to invalid parameters. |
| supportedFeatures | SupportedFeatures | C | 0..1 | Features supported by the NF Service Producer.  This IE shall be present when rejecting a request due to an unsupported query parameter, if at least one feature is defined for the corresponding service in the version of the specification that the NF Service Producer implements (see clause 5.2.9 of 3GPP TS 29.500 [25]).  When present, this IE shall indicate the features supported by the NF Service Producer; if the NF Service Producer supports no features, this IE shall be set to the character "0". |
| accessTokenError | AccessTokenErr | C | 0..1 | This IE should be present if an SCP request to get an access token was rejected by the NRF.  When present, it should contain the Access Token Error payload received from the NRF. |
| accessTokenRequest | AccessTokenReq | O | 0..1 | This IE may be present if an SCP request to get an access token was rejected by the NRF.  When present, it shall contain the Access Token Request that was sent by the SCP. |
| nrfId | Fqdn | O | 0..1 | This IE may be present if an SCP request to get an access token was rejected by the NRF.  When present, it shall contain the Identity (i.e. FQDN) of the NRF that rejected the access token request. |
| NOTE 1: See IETF RFC 7807 [9] for detailed information and guidance for each attribute, and 3GPP TS 29.501 [2] for guidelines on error handling support by 5GC SBI APIs.  NOTE 2: Additional attributes may be defined per API. | | | | |

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

### 5.4.2 Simple Data Types

This clause specifies common simple data types.

Table 5.4.2-1: Simple Data Types

|  |  |  |
| --- | --- | --- |
| Type Name | Type Definition | Description |
| ApplicationId | string | String providing an application identifier. |
| ApplicationIdRm | string | This data type is defined in the same way as the "ApplicationId" data type, but with the OpenAPI "nullable: true" property. |
| PduSessionId | integer | Unsigned integer identifying a PDU session, within the range 0 to 255, as specified in clause 11.2.3.1b, bits 1 to 8, of 3GPP TS 24.007 [13]. If the PDU Session ID is allocated by the Core Network for UEs not supporting N1 mode, reserved range 64 to 95 is used. PDU Session ID within the reserved range is only visible in the Core Network (NOTE). |
| Mcc | string | Mobile Country Code part of the PLMN, comprising 3 digits, as defined in clause 9.3.3.5 of 3GPP TS 38.413 [11].  Pattern: '^[0-9]{3}$' |
| MccRm | string | This data type is defined in the same way as the "Mcc" data type, but with the OpenAPI "nullable: true" property. |
| Mnc | string | Mobile Network Code part of the PLMN, comprising 2 or 3 digits, as defined in clause 9.3.3.5 of 3GPP TS 38.413 [11].  Pattern: '^[0-9]{2,3}$' |
| MncRm | string | This data type is defined in the same way as the "Mnc" data type, but with the OpenAPI "nullable: true" property. |
| Tac | string | 2 or 3-octet string identifying a tracking area code as specified in clause 9.3.3.10 of 3GPP TS 38.413 [11], in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC shall appear last in the string.  Examples:  A legacy TAC 0x4305 shall be encoded as "4305".  An extended TAC 0x63F84B shall be encoded as "63F84B" |
| TacRm | string | This data type is defined in the same way as the "Tac" data type, but with the OpenAPI "nullable: true" property. |
| EutraCellId | string | 28-bit string identifying an E-UTRA Cell Id as specified in clause 9.3.1.9 of 3GPP TS 38.413 [11], in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the Cell Id shall appear first in the string, and the character representing the 4 least significant bit of the Cell Id shall appear last in the string.  Pattern: '^[A-Fa-f0-9]{7}$'  Example:  An E-UTRA Cell Id 0x5BD6007 shall be encoded as "5BD6007". |
| EutraCellIdRm | string | This data type is defined in the same way as the "EutraCellId" data type, but with the OpenAPI "nullable: true" property. |
| NrCellId | string | 36-bit string identifying an NR Cell Id as specified in clause 9.3.1.7 of 3GPP TS 38.413 [11], in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the Cell Id shall appear first in the string, and the character representing the 4 least significant bit of the Cell Id shall appear last in the string.  Pattern: '^[A-Fa-f0-9]{9}$'  Example:  An NR Cell Id 0x225BD6007 shall be encoded as "225BD6007". |
| NrCellIdRm | string | This data type is defined in the same way as the "NrCellId" data type, but with the OpenAPI "nullable: true" property. |
| Dnai | string | DNAI (Data network access identifier), see clause 5.6.7 of 3GPP TS 23.501 [8]. |
| DnaiRm | string | This data type is defined in the same way as the "Dnai" data type, but with the OpenAPI "nullable: true" property. |
| 5GMmCause | Uinteger | This represents the 5GMM cause code values as specified in 3GPP TS 24.501 [20]. |
| AreaCodeRm | string | This data type is defined in the same way as the "AreaCode" data type, but with the OpenAPI "nullable: true" property. |
| AmfName | Fqdn | FQDN (Fully Qualified Domain Name) of the AMF as defined in clause 28.3.2.5 of 3GPP TS 23.003 [7]. |
| AreaCode | string | Values are operator specific. |
| N3IwfId | string | This represents the identifier of the N3IWF ID as specified in clause 9.3.1.57 of 3GPP TS 38.413 [11] in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the N3IWF ID shall appear first in the string, and the character representing the 4 least significant bit of the N3IWF ID shall appear last in the string.  Pattern: '^[A-Fa-f0-9]+$'  Example:  The N3IWF Id 0x5BD6 shall be encoded as "5BD6". |
| WAgfId | string | This represents the identifier of the W-AGF ID as specified in clause 9.3.1.162 of 3GPP TS 38.413 [11] in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the W-AGF ID shall appear first in the string, and the character representing the 4 least significant bit of the W-AGF ID shall appear last in the string.  Pattern: '^[A-Fa-f0-9]+$'  Example:  The W-AGF Id 0x5BD6 shall be encoded as "5BD6". |
| TngfId | string | This represents the identifier of the TNGF ID as specified in clause 9.3.1.161 of 3GPP TS 38.413 [11] in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TNGF ID shall appear first in the string, and the character representing the 4 least significant bit of the TNGF ID shall appear last in the string.  Pattern: '^[A-Fa-f0-9]+$'  Example:  The TNGF Id 0x5BD6 shall be encoded as "5BD6". |
| NgeNbId | string | This represents the identifier of the ng-eNB ID as specified in clause 9.3.1.8 of 3GPP TS 38.413 [11].  The string shall be formatted with following pattern:  Pattern: '^('MacroNGeNB-[A-Fa-f0-9]{5}|  LMacroNGeNB-[A-Fa-f0-9]{6}|  SMacroNGeNB-[A-Fa-f0-9]{5})$'  The value of the ng-eNB ID shall be encoded in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The padding 0 shall be added to make multiple nibbles, so the most significant character representing the padding 0 if required together with the 4 most significant bits of the ng-eNB ID shall appear first in the string, and the character representing the 4 least significant bit of the ng-eNB ID (to form a nibble) shall appear last in the string.  Examples:  " SMacroNGeNB-34B89" indicates a Short Macro NG-eNB ID with value 0x34B89. |
| Nid | string | This represents the Network Identifier, which together with a PLMN ID is used to identify an SNPN (see 3GPP TS 23.003 [7] and 3GPP TS 23.501 [8] clause 5.30.2.1).  Pattern: '^[A-Fa-f0-9]{11}$' |
| NidRm | string | This data type is defined in the same way as the "Nid" data type, but with the OpenAPI "nullable: true" property. |
| NfSetId | string | NF Set Identifier (see clause 28.12 of 3GPP TS 23.003 [7]), formatted as the following string:  " set<Set ID>.<nftype>set.5gc.mnc<MNC>.mcc<MCC>", or  "set<SetID>.<NFType>set.5gc.nid<NID>.mnc<MNC>.mcc<MCC>"  with  <MCC> encoded as defined in clause 5.4.2 ("Mcc" data type definition)  <MNC> encoded as defined in clause 5.4.2 ("Mnc" data type definition)  <NFType> encoded as a value defined in Table 6.1.6.3.3-1 of 3GPP TS 29.510 [29] but with lower case characters  <Set ID> encoded as a string of characters consisting of alphabetic characters (A-Z and a-z), digits (0-9) and/or the hyphen (-) and that shall end with either an alphabetic character or a digit.  Pattern: '^([A-Za-z0-9\-]\*[A-Za-z0-9])$'    Examples:   "setxyz.smfset.5gc.mnc012.mcc345"  "set12.pcfset.5gc.mnc012.mcc345" |
| NfServiceSetId | string | NF Service Set Identifier (see clause 28.12 of 3GPP TS 23.003 [7]) formatted as the following string:  " set<Set ID>.sn<Service Name>.nfi<NF Instance ID>.5gc.mnc<MNC>.mcc<MCC>">", or  "set<SetID>.sn<ServiceName>.nfi<NFInstanceID>.5gc.nid<NID>.mnc<MNC>.mcc<MCC>"  with  <MCC> encoded as defined in clause 5.4.2 ("Mcc" data type definition)  <MNC> encoded as defined in clause 5.4.2 ("Mnc" data type definition)  <NID> encoded as defined in clause 5.4.2 ("Nid" data type definition)  <NFInstanceId> encoded as defined in clause 5.3.2  <ServiceName> encoded as defined in 3GPP TS 29.510 [29]  <Set ID> encoded as a string of characters consisting of alphabetic characters (A-Z and a-z), digits (0-9) and/or the hyphen (-) and that shall end with either an alphabetic character or a digit.  Pattern: '^([A-Za-z0-9\-]\*[A-Za-z0-9])$  Examples:  "setxyz.snnsmf-pdusession.nfi54804518-4191-46b3-955c-ac631f953ed8.5gc.mnc012.mcc345"  "set2.snnpcf-smpolicycontrol.nfi54804518-4191-46b3-955c-ac631f953ed8.5gc.mnc012.mcc345" |
| PlmnAssiUeRadioCapId | Bytes | String with format "byte" as defined in OpenAPI Specification [3], i.e. base64-encoded characters, encoding the "UE radio capability ID" IE as specified in clause 9.11.3.68 of 3GPP TS 24.501 [20] (starting from octet 1). |
| ManAssiUeRadioCapId | Bytes | String with format "byte" as defined in OpenAPI Specification [3], i.e. base64-encoded characters, encoding the "UE radio capability ID" IE as specified in clause 9.11.3.68 of 3GPP TS 24.501 [20] (starting from octet 1). |
| TypeAllocationCode | string | Type Allocation Code (TAC) of the UE, comprising the initial eight-digit portion of the 15-digit IMEI and 16-digit IMEISV codes. See clause 6.2 of 3GPP TS 23.003 [7].  Pattern: '^[0-9]{8}$' |
| HfcNId | string | This IE represents the identifier of the HFC node Id as specified in CableLabs WR-TR-5WWC-ARCH [32]. It is provisioned by the wireline operator as part of wireline operations and may contain up to six characters. |
| HfcNIdRm | string | This data type is defined in the same way as the "HfcNId" data type, but with the OpenAPI "nullable: true" property. |
| ENbId | string | This represents the identifier of the eNB ID as specified in clause 9.2.1.37 of 3GPP TS 36.413 [16].  The string shall be formatted with following pattern:  Pattern: '^('MacroeNB-[A-Fa-f0-9]{5}|LMacroeNB-[A-Fa-f0-9]{6}|SMacroeNB-[A-Fa-f0-9]{5}|HomeeNB-[A-Fa-f0-9]{7})$'  The value of the eNB ID shall be encoded in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The padding 0 shall be added to make multiple nibbles, so the most significant character representing the padding 0 if required together with the 4 most significant bits of the eNB ID shall appear first in the string, and the character representing the 4 least significant bit of the eNB ID (to form a nibble) shall appear last in the string.  Examples:  "SMacroeNB-34B89" indicates a Short Macro eNB ID with value 0x34B89. |
| Gli | Bytes | Global Line Identifier uniquely identifying the line connecting the 5G-BRG or FN-BRG to the 5GS. See clause 28.16.3 of 3GPP TS 23.003 [7].  This shall be encoded as a string with format "byte" as defined in OpenAPI Specification [3], i.e. base64-encoded characters, representing the GLI value (up to 150 bytes) encoded as specified in BBF WT-470 [37]. |
| Gci | string | Global Cable Identifier uniquely identifying the connection between the 5G-CRG or FN-CRG to the 5GS. See clause 28.15.4 of 3GPP TS 23.003 [7].  This shall be encoded as a string per clause 28.15.4 of 3GPP TS 23.003 [7], and compliant with the syntax specified in clause 2.2 of IETF RFC 7542 [126] for the username part of a NAI. The GCI value is specified in CableLabs WR-TR-5WWC-ARCH [32]. |
| NsSrg | string | String representing Network Slice Simultaneous Registration Group (see clause 5.15.12 of 3GPP TS 23.501 [8]) |
| NsSrgRm | string | This data type is defined in the same way as the " NsSrg" data type, but with the OpenAPI "nullable: true" property. |
| NOTE: For a PDN connection established via MME, the PDU Session ID value is set to 64 plus the EPS bearer ID of the default EPS bearer of the PDN connection; for a PDN connection established via ePDG, the PDU Session ID value is set to 80 plus the EPS bearer ID of the default EPS bearer of the PDN connection. | | |

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

#### 5.4.4.69 Type: EcsServerAddr

Table 5.4.4.69-1: Definition of type EcsServerAddr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| ecsFqdnList | array(Fqdn) | C | 1..N | This IE shall be included if available.  When present, it shall contain the list of FQDN(s) of Edge Configuration Server(s). |
| ecsIpAddressList | array(IpAddr) | C | 1..N | This IE shall be included if available.  When present, it shall contain the list of IP Address (es) of Edge Configuration Server(s). |
| ecsProviderId | string | C | 0..1 | This IE shall be included if available.  When present, it shall contain the identifier of the Edge Configuration Server Provider. |

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

## A.2 Data related to Common Data Types

openapi: 3.0.0

info:

version: '1.3.0-alpha.4'

title: 'Common Data Types'

description: |

Common Data Types for Service Based Interfaces.

© 2021, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

externalDocs:

description: 3GPP TS 29.571 Common Data Types for Service Based Interfaces, version 17.4.0

url: 'http://www.3gpp.org/ftp/Specs/archive/29\_series/29.571/'

paths: {}

components:

schemas:

#

# Common Data Types for Generic usage definitiones as defined in clause 5.2

#

#

# COMMON SIMPLE DATA TYPES

#

Binary:

format: binary

type: string

description: string with format "binary" as defined in OpenAPI

BinaryRm:

format: binary

type: string

nullable: true

description: string with format "binary" as defined in OpenAPI OpenAPI with "nullable: true" property.

Bytes:

format: byte

type: string

description: string with format "bytes" as defined in OpenAPI

BytesRm:

format: byte

type: string

nullable: true

description: string with format "bytes" as defined in OpenAPI OpenAPI with "nullable: true" property.

Date:

format: date

type: string

description: string with format "date" as defined in OpenAPI

DateRm:

format: date

type: string

nullable: true

description: string with format "date" as defined in OpenAPI OpenAPI with "nullable: true" property.

DateTime:

format: date-time

type: string

description: string with format "date-time" as defined in OpenAPI.

DateTimeRm:

format: date-time

type: string

nullable: true

description: string with format "date-time" as defined in OpenAPI with "nullable: true" property.

DiameterIdentity:

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

DiameterIdentityRm:

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

property.

Double:

format: double

type: number

description: string with format "double" as defined in OpenAPI

DoubleRm:

format: double

type: number

nullable: true

description: string with format "double" as defined in OpenAPI with "nullable: true" property.

DurationSec:

type: integer

description: indicating a time in seconds.

DurationSecRm:

type: integer

nullable: true

description: indicating a time in seconds with OpenAPI defined "nullable: true" property.

Float:

format: float

type: number

description: string with format "float" as defined in OpenAPI.

FloatRm:

format: float

type: number

nullable: true

description: string with format "float" as defined in OpenAPI with the OpenAPI defined "nullable: true" property.

Int32:

format: int32

type: integer

description: string with format "int32" as defined in OpenAPI.

Int32Rm:

format: int32

type: integer

nullable: true

description: string with format "int32" as defined in OpenAPI with the OpenAPI defined "nullable: true" property.

Int64:

type: integer

format: int64

description: string with format "int64" as defined in OpenAPI.

Int64Rm:

format: int64

type: integer

nullable: true

description: string with format "int64" as defined in OpenAPI with the OpenAPI defined "nullable: true" property.

Ipv4Addr:

type: string

pattern: '^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])$'

example: '198.51.100.1'

description: String identifying a IPv4 address formatted in the "dotted decimal" notation as defined in RFC 1166.

Ipv4AddrRm:

type: string

pattern: '^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])$'

example: '198.51.100.1'

nullable: true

description: String identifying a IPv4 address formatted in the "dotted decimal" notation as defined in RFC 1166 with the OpenAPI defined "nullable: true" property.

Ipv4AddrMask:

type: string

pattern: '^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(\/([0-9]|[1-2][0-9]|3[0-2]))$'

example: '198.51.0.0/16'

description: String identifying a IPv4 address mask formatted in the "dotted decimal" notation as defined in RFC 1166.

Ipv4AddrMaskRm:

type: string

pattern: '^(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])(\/([0-9]|[1-2][0-9]|3[0-2]))$'

example: '198.51.0.0/16'

nullable: true

description: String identifying a IPv4 address mask formatted in the "dotted decimal" notation as defined in RFC 1166 with the OpenAPI defined "nullable: true" property.

Ipv6Addr:

type: string

allOf:

- pattern: '^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))$'

- pattern: '^((([^:]+:){7}([^:]+))|((([^:]+:)\*[^:]+)?::(([^:]+:)\*[^:]+)?))$'

example: '2001:db8:85a3::8a2e:370:7334'

description: String identifying an IPv6 address formatted according to clause 4 of RFC5952. The mixed IPv4 IPv6 notation according to clause 5 of RFC5952 shall not be used

Ipv6AddrRm:

type: string

allOf:

- pattern: '^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))$'

- pattern: '^((([^:]+:){7}([^:]+))|((([^:]+:)\*[^:]+)?::(([^:]+:)\*[^:]+)?))$'

example: '2001:db8:85a3::8a2e:370:7334'

nullable: true

description: String identifying an IPv6 address formatted according to clause 4 of RFC5952 with the OpenAPI "nullable: true" property. The mixed IPv4 IPv6 notation according to clause 5 of RFC5952 shall not be used.

Ipv6Prefix:

type: string

allOf:

- pattern: '^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))(\/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))$'

- pattern: '^((([^:]+:){7}([^:]+))|((([^:]+:)\*[^:]+)?::(([^:]+:)\*[^:]+)?))(\/.+)$'

example: '2001:db8:abcd:12::0/64'

description: String identifying an IPv6 address prefix formatted according to clause 4 of RFC 5952. IPv6Prefix data type may contain an individual /128 IPv6 address.

Ipv6PrefixRm:

type: string

allOf:

- pattern: '^((:|(0?|([1-9a-f][0-9a-f]{0,3}))):)((0?|([1-9a-f][0-9a-f]{0,3})):){0,6}(:|(0?|([1-9a-f][0-9a-f]{0,3})))(\/(([0-9])|([0-9]{2})|(1[0-1][0-9])|(12[0-8])))$'

- pattern: '^((([^:]+:){7}([^:]+))|((([^:]+:)\*[^:]+)?::(([^:]+:)\*[^:]+)?))(\/.+)$'

nullable: true

description: String identifying an IPv6 address prefix formatted according to clause 4 of RFC 5952 with the OpenAPI "nullable: true" property. IPv6Prefix data type may contain an individual /128 IPv6 address.

MacAddr48:

type: string

pattern: '^([0-9a-fA-F]{2})((-[0-9a-fA-F]{2}){5})$'

description: String identifying a MAC address formatted in the hexadecimal notation according to clause 1.1 and clause 2.1 of RFC 7042

MacAddr48Rm:

type: string

pattern: '^([0-9a-fA-F]{2})((-[0-9a-fA-F]{2}){5})$'

nullable: true

description: String identifying a MAC address formatted in the hexadecimal notation according to clause 1.1 and clause 2.1 of RFC 7042 with the OpenAPI "nullable: true" property.

SupportedFeatures:

type: string

pattern: '^[A-Fa-f0-9]\*$'

description: A string used to indicate the features supported by an API that is used as defined in clause 6.6 in 3GPP TS 29.500. The string shall contain a bitmask indicating supported features in hexadecimal representation Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent the support of 4 features as described in table 5.2.2-3. The most significant character representing the highest-numbered features shall appear first in the string, and the character representing features 1 to 4 shall appear last in the string. The list of features and their numbering (starting with 1) are defined separately for each API. If the string contains a lower number of characters than there are defined features for an API, all features that would be represented by characters that are not present in the string are not supported

Uinteger:

type: integer

minimum: 0

description: Unsigned Integer, i.e. only value 0 and integers above 0 are permissible.

UintegerRm:

type: integer

minimum: 0

description: Unsigned Integer, i.e. only value 0 and integers above 0 are permissible with the OpenAPI "nullable: true" property.

nullable: true

Uint16:

type: integer

minimum: 0

maximum: 65535

description: Integer where the allowed values correspond to the value range of an unsigned 16-bit integer.

Uint16Rm:

type: integer

minimum: 0

maximum: 65535

nullable: true

description: Integer where the allowed values correspond to the value range of an unsigned 16-bit integer with the OpenAPI "nullable: true" property.

Uint32:

type: integer

minimum: 0

maximum: 4294967295 #(2^32)-1

description: Integer where the allowed values correspond to the value range of an unsigned 32-bit integer.

Uint32Rm:

format: int32

type: integer

minimum: 0

maximum: 4294967295 #(2^32)-1

nullable: true

description: Integer where the allowed values correspond to the value range of an unsigned 32-bit integer with the OpenAPI "nullable: true" property.

Uint64:

type: integer

minimum: 0

maximum: 18446744073709551615 #(2^64)-1

description: Integer where the allowed values correspond to the value range of an unsigned 64-bit integer.

Uint64Rm:

type: integer

minimum: 0

maximum: 18446744073709551615 #(2^64)-1

nullable: true

description: Integer where the allowed values correspond to the value range of an unsigned 16-bit integer with the OpenAPI "nullable: true" property.

Uri:

type: string

description: String providing an URI formatted according to RFC 3986

UriRm:

type: string

nullable: true

description: String providing an URI formatted according to RFC 3986 with the OpenAPI "nullable: true" property.

VarUeId:

type: string

pattern: '^(imsi-[0-9]{5,15}|nai-.+|msisdn-[0-9]{5,15}|extid-[^@]+@[^@]+|gci-.+|gli-.+|.+)$'

description: String represents the SUPI or GPSI

VarUeIdRm:

type: string

pattern: '^(imsi-[0-9]{5,15}|nai-.+|msisdn-[0-9]{5,15}|extid-[^@]+@[^@]+|gci-.+|gli-.+|.+)$'

nullable: true

description: String represents the SUPI or GPSI with the OpenAPI "nullable: true" property.

TimeZone:

type: string

example: '-08:00+1'

description: String with format "<time-numoffset>" optionally appended by "<daylightSavingTime>", where - <time-numoffset> shall represent the time zone adjusted for daylight saving time and be encoded as time-numoffset as defined in clause 5.6 of IETF RFC 3339; - <daylightSavingTime> shall represent the adjustment that has been made and shall be encoded as "+1" or "+2" for a +1 or +2 hours adjustment. The example is for 8 hours behind UTC, +1 hour adjustment for Daylight Saving Time.

TimeZoneRm:

type: string

nullable: true

description: >

String with format "<time-numoffset>" optionally appended by "<daylightSavingTime>", where

- <time-numoffset> shall represent the time zone adjusted for daylight saving time and be encoded as time-numoffset as defined in clause 5.6 of IETF RFC 3339;

- <daylightSavingTime> shall represent the adjustment that has been made and shall be encoded as "+1" or "+2" for a +1 or +2 hours adjustment.

But with the OpenAPI "nullable: true" property.

StnSr:

type: string

description: String representing the STN-SR as defined in clause 18.6 of 3GPP TS 23.003.

StnSrRm:

type: string

nullable: true

description: String representing the STN-SR as defined in clause 18.6 of 3GPP TS 23.003 with the OpenAPI "nullable: true" property.

CMsisdn:

type: string

pattern: '^[0-9]{5,15}$'

description: String representing the C-MSISDN as defined in clause 18.7 of 3GPP TS 23.003.

CMsisdnRm:

type: string

pattern: '^[0-9]{5,15}$'

nullable: true

description: String representing the C-MSISDN as defined in clause 18.7 of 3GPP TS 23.003 with the OpenAPI "nullable: true" property.

DayOfWeek:

type: integer

minimum: 1

maximum: 7

description: integer between and including 1 and 7 denoting a weekday. 1 shall indicate Monday, and the subsequent weekdays shall be indicated with the next higher numbers. 7 shall indicate Sunday.

TimeOfDay:

type: string

description: String with format partial-time or full-time as defined in clause 5.6 of IETF RFC 3339. Examples, 20:15:00, 20:15:00-08:00 (for 8 hours behind UTC).

EmptyObject:

description: Empty JSON object { }, it is defined with the keyword additionalProperties false

type: object

additionalProperties: false

Fqdn:

description: Fully Qualified Domain Name

type: string

pattern: '^([0-9A-Za-z]([-0-9A-Za-z]{0,61}[0-9A-Za-z])?\.)+[A-Za-z]{2,63}\.?$'

minLength: 4

maxLength: 253

FqdnRm:

description: Fully Qualified Domain Name, but it also allows the null value

anyOf:

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

- $ref: '#/components/schemas/NullValue'

*(... text not shown for clarity ...)*

#

# COMMON STRUCTURED DATA TYPES

#

ProblemDetails:

description: Provides additional information in an error response.

type: object

properties:

type:

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

title:

type: string

status:

type: integer

detail:

type: string

description: A human-readable explanation specific to this occurrence of the problem.

instance:

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

cause:

type: string

description: A machine-readable application error cause specific to this occurrence of the problem. This IE should be present and provide application-related error information, if available.

invalidParams:

type: array

items:

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

minItems: 1

supportedFeatures:

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

accessTokenError:

$ref: 'TS29510\_Nnrf\_AccessToken.yaml#/components/schemas/AccessTokenErr'

accessTokenRequest:

$ref: 'TS29510\_Nnrf\_AccessToken.yaml#/components/schemas/AccessTokenReq'

nrfId:

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

*(... text not shown for clarity ...)*

#

# Data Types related to 5G Network as defined in clause 5.4

#

#

# SIMPLE DATA TYPES

#

ApplicationId:

type: string

description: String providing an application identifier.

ApplicationIdRm:

type: string

nullable: true

description: String providing an application identifier with the OpenAPI "nullable: true" property.

PduSessionId:

type: integer

minimum: 0

maximum: 255

description: Unsigned integer identifying a PDU session, within the range 0 to 255, as specified in clause 11.2.3.1b, bits 1 to 8, of 3GPP TS 24.007. If the PDU Session ID is allocated by the Core Network for UEs not supporting N1 mode, reserved range 64 to 95 is used. PDU Session ID within the reserved range is only visible in the Core Network.

Mcc:

type: string

pattern: '^\d{3}$'

description: Mobile Country Code part of the PLMN, comprising 3 digits, as defined in clause 9.3.3.5 of 3GPP TS 38.413.

MccRm:

type: string

pattern: '^\d{3}$'

nullable: true

description: Mobile Country Code part of the PLMN, comprising 3 digits, as defined in clause 9.3.3.5 of 3GPP TS 38.413 with the OpenAPI "nullable: true" property.

Mnc:

type: string

pattern: '^\d{2,3}$'

description: Mobile Network Code part of the PLMN, comprising 2 or 3 digits, as defined in clause 9.3.3.5 of 3GPP TS 38.413.

MncRm:

type: string

pattern: '^\d{2,3}$'

nullable: true

description: Mobile Network Code part of the PLMN, comprising 2 or 3 digits, as defined in clause 9.3.3.5 of 3GPP TS 38.413 with the OpenAPI "nullable: true" property.

Tac:

type: string

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

description: 2 or 3-octet string identifying a tracking area code as specified in clause 9.3.3.10 of 3GPP TS 38.413, in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC shall appear last in the string.

TacRm:

type: string

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

nullable: true

description: This data type is defined in the same way as the "Tac" data type, but with the OpenAPI "nullable: true" property.

EutraCellId:

type: string

pattern: '^[A-Fa-f0-9]{7}$'

description: 28-bit string identifying an E-UTRA Cell Id as specified in clause 9.3.1.9 of 3GPP TS 38.413, in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the Cell Id shall appear first in the string, and the character representing the 4 least significant bit of the Cell Id shall appear last in the string.

EutraCellIdRm:

type: string

pattern: '^[A-Fa-f0-9]{7}$'

nullable: true

description: This data type is defined in the same way as the "EutraCellId" data type, but with the OpenAPI "nullable: true" property.

NrCellId:

type: string

pattern: '^[A-Fa-f0-9]{9}$'

description: 36-bit string identifying an NR Cell Id as specified in clause 9.3.1.7 of 3GPP TS 38.413, in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the Cell Id shall appear first in the string, and the character representing the 4 least significant bit of the Cell Id shall appear last in the string.

NrCellIdRm:

type: string

pattern: '^[A-Fa-f0-9]{9}$'

nullable: true

description: This data type is defined in the same way as the "NrCellId" data type, but with the OpenAPI "nullable: true" property.

Dnai:

type: string

description: DNAI (Data network access identifier), see clause 5.6.7 of 3GPP TS 23.501.

DnaiRm:

type: string

nullable: true

description: This data type is defined in the same way as the "Dnai" data type, but with the OpenAPI "nullable: true" property.

5GMmCause:

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

AmfName:

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

AreaCode:

type: string

description: Values are operator specific.

AreaCodeRm:

type: string

nullable: true

description: This data type is defined in the same way as the "AreaCode" data type, but with the OpenAPI "nullable: true" property.

N3IwfId:

type: string

pattern: '^[A-Fa-f0-9]+$'

description: This represents the identifier of the N3IWF ID as specified in clause 9.3.1.57 of 3GPP TS 38.413 in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the N3IWF ID shall appear first in the string, and the character representing the 4 least significant bit of the N3IWF ID shall appear last in the string.

WAgfId:

type: string

pattern: '^[A-Fa-f0-9]+$'

description: This represents the identifier of the W-AGF ID as specified in clause 9.3.1.162 of 3GPP TS 38.413 in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the W-AGF ID shall appear first in the string, and the character representing the 4 least significant bit of the W-AGF ID shall appear last in the string.

TngfId:

type: string

pattern: '^[A-Fa-f0-9]+$'

description: This represents the identifier of the TNGF ID as specified in clause 9.3.1.161 of 3GPP TS 38.413 in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the TNGF ID shall appear first in the string, and the character representing the 4 least significant bit of the TNGF ID shall appear last in the string.

NgeNbId:

type: string

pattern: '^(MacroNGeNB-[A-Fa-f0-9]{5}|LMacroNGeNB-[A-Fa-f0-9]{6}|SMacroNGeNB-[A-Fa-f0-9]{5})$'

description: This represents the identifier of the ng-eNB ID as specified in clause 9.3.1.8 of 3GPP TS 38.413. The value of the ng-eNB ID shall be encoded in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The padding 0 shall be added to make multiple nibbles, so the most significant character representing the padding 0 if required together with the 4 most significant bits of the ng-eNB ID shall appear first in the string, and the character representing the 4 least significant bit of the ng-eNB ID (to form a nibble) shall appear last in the string.

example: SMacroNGeNB-34B89

Nid:

type: string

pattern: '^[A-Fa-f0-9]{11}$'

description: This represents the Network Identifier, which together with a PLMN ID is used to identify an SNPN (see 3GPP TS 23.003 and 3GPP TS 23.501 clause 5.30.2.1).

NidRm:

type: string

pattern: '^[A-Fa-f0-9]{11}$'

nullable: true

description: This data type is defined in the same way as the "Nid" data type, but with the OpenAPI "nullable: true" property.

NfSetId:

type: string

description: NF Set Identifier (see clause 28.12 of 3GPP TS 23.003), formatted as the following string " set<Set ID>.<nftype>set.5gc.mnc<MNC>.mcc<MCC>", or "set<SetID>. <NFType>set.5gc.nid<NID>.mnc<MNC>.mcc<MCC>" with <MCC> encoded as defined in clause 5.4.2 ("Mcc" data type definition) <MNC> encoded as defined in clause 5.4.2 ("Mnc" data type definition) <NFType> encoded as a value defined in Table 6.1.6.3.3-1 of 3GPP TS 29.510 but with lower case characters <Set ID> encoded as a string of characters consisting of alphabetic characters (A-Z and a-z), digits (0-9) and/or the hyphen (-) and that shall end with either an alphabetic character or a digit.

NfServiceSetId:

type: string

description: NF Service Set Identifier (see clause 28.12 of 3GPP TS 23.003) formatted as the following string " set<Set ID>.sn<Service Name>.nfi<NF Instance ID>.5gc.mnc<MNC>.mcc<MCC>">", or "set<SetID>.sn<ServiceName>.nfi<NFInstanceID>.5gc.nid<NID>.mnc<MNC>.mcc<MCC>" with <MCC> encoded as defined in clause 5.4.2 ("Mcc" data type definition) <MNC> encoded as defined in clause 5.4.2 ("Mnc" data type definition) <NID> encoded as defined in clause 5.4.2 ("Nid" data type definition) <NFInstanceId> encoded as defined in clause 5.3.2 <ServiceName> encoded as defined in 3GPP TS 29.510 <Set ID> encoded as a string of characters consisting of alphabetic characters (A-Z and a-z), digits (0-9) and/or the hyphen (-) and that shall end with either an alphabetic character or a digit.

PlmnAssiUeRadioCapId:

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

ManAssiUeRadioCapId:

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

TypeAllocationCode:

type: string

pattern: '^[0-9]{8}$'

description: Type Allocation Code (TAC) of the UE, comprising the initial eight-digit portion of the 15-digit IMEI and 16-digit IMEISV codes. See clause 6.2 of 3GPP TS 23.003.

HfcNId:

type: string

maxLength: 6

description: This IE represents the identifier of the HFC node Id as specified in CableLabs WR-TR-5WWC-ARCH. It is provisioned by the wireline operator as part of wireline operations and may contain up to six characters.

HfcNIdRm:

type: string

maxLength: 6

nullable: true

description: This data type is defined in the same way as the "HfcNId" data type, but with the OpenAPI "nullable: true" property.

ENbId:

type: string

pattern: '^(MacroeNB-[A-Fa-f0-9]{5}|LMacroeNB-[A-Fa-f0-9]{6}|SMacroeNB-[A-Fa-f0-9]{5}|HomeeNB-[A-Fa-f0-9]{7})$'

description: This represents the identifier of the eNB ID as specified in clause 9.2.1.37 of 3GPP TS 36.413. The string shall be formatted with the following pattern '^('MacroeNB-[A-Fa-f0-9]{5}|LMacroeNB-[A-Fa-f0-9]{6}|SMacroeNB-[A-Fa-f0-9]{5}|HomeeNB-[A-Fa-f0-9]{7})$' The value of the eNB ID shall be encoded in hexadecimal representation. Each character in the string shall take a value of "0" to "9", "a" to "f" or "A" to "F" and shall represent 4 bits. The padding 0 shall be added to make multiple nibbles, so the most significant character representing the padding 0 if required together with the 4 most significant bits of the eNB ID shall appear first in the string, and the character representing the 4 least significant bit of the eNB ID (to form a nibble) shall appear last in the string.

Gli:

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

Gci:

type: string

description: Global Cable Identifier uniquely identifying the connection between the 5G-CRG or FN-CRG to the 5GS. See clause 28.15.4 of 3GPP TS 23.003. This shall be encoded as a string per clause 28.15.4 of 3GPP TS 23.003, and compliant with the syntax specified in clause 2.2 of IETF RFC 7542 for the username part of a NAI. The GCI value is specified in CableLabs WR-TR-5WWC-ARCH.

NsSrg:

type: string

description: String providing a Network Slice Simultaneous Registration Group. See clause 5.15.12 of 3GPP TS 23.501

NsSrgRm:

type: string

nullable: true

description: String providing a Network Slice Simultaneous Registration Group with the OpenAPI "nullable: true" property. See clause 5.15.12 of 3GPP TS 23.501

*(... text not shown for clarity ...)*

EcsServerAddr:

description: Contains the Edge Configuration Server Address Configuration Information as defined in clause 5.2.3.6.1 of 3GPP TS 23.502.

type: object

properties:

ecsFqdnList:

type: array

items:

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

minItems: 1

ecsIpAddressList:

type: array

items:

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

minItems: 1

ecsProviderId:

type: string

*(... text not shown for clarity ...)*

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