ETSI TS 129 501 V15.6.0 (2020-01)

5G;
5G System;
Principles and Guidelines for Services Definition;
Stage 3
(3GPP TS 29.501 version 15.6.0 Release 15)

Reference
RTS/TSGC-0429501vf60

Keywords
5G

ETSI

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

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

Legal Notice

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

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

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

Modal verbs terminology

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

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

Contents

Foreword

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

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

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

1 Scope

The present document defines design principles and documentation guidelines for 5GC SBI APIs. These principles and guidelines should be followed when drafting the 5G System SBI Stage 3 specifications.

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

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

[2] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

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

[4] OpenAPI: "OpenAPI 3.0.0 Specification", https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md.

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

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

[7] IETF RFC 7396: "JSON Merge Patch".

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

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

[10] IETF RFC 5789: "PATCH Method for HTTP"

[11] IETF RFC 8288: "Web Linking".

[12] IANA: "HTTP Status Code Registry at IANA", http://www.iana.org/assignments/http-status-codes

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

[14] Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.

[15] Erik Wilde, Cesare Pautasso, REST: From Research to Practice, Springer

[16] YAML 1.2: "YAML Ain't Markup Language", http://yaml.org.

[17] Semantic Versioning Specification: https://semver.org

[18] 3GPP TS 29.510: "5G System; Network Function Repository Services; Stage 3".

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

[20] 3GPP TS 29.502: "5G System; Session Management Services; Stage 3".

[21] 3GPP TS 29.509: "Authentication Server Services; Stage 3".

[22] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

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

[24] 3GPP TS 29.573: "5G System; Public Land Mobile Network (PLMN) Interconnection;Stage 3".

[25] 3GPP TR 21.900: "Technical Specification Group working methods".

3 Definitions and abbreviations

3.1 Definitions

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

3.2 Abbreviations

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

5GC	5G Core Network
CNF	Conjunctive Normal Form
DNF	Disjunctive Normal Form
HAL	Hypertext Application Language
HATEOAS	Hypermedia as the Engine of Application State
SBI	Service Based Interface
YAML	YAML Ain't Markup Language

4 Design Principles for 5GC SBI APIs

4.1 General Principles

Each 5GC SBI API specification should include the following information for each specified service:

- Purpose of the API;
- URIs of resources;
- Supported HTTP methods for a given resource;
- Supported representations (e.g. JSON, see IETF RFC 8259 [3]);
- Request body schema(s) (where applicable);
- Response body schema(s) (where applicable);
- Supported response status codes;
- Relation types supported if HATEOAS is implemented by the API;
- A reference in the resource description clause to one of the archetypes defined in Annex C if the resource design matches one of them; and
- A list defining identifiers of optional features (see clause 6.6 of 3GPP TS 29.500 [2] for related procedures).

For each specified service a clause to a normative Annex should be provided containing the OpenAPI definitions according to OpenAPI Specification [4] for the service. The specifications should state that content of this normative

annex takes precedence when being discrepant to other parts of the specification with respect to the encoding of information elements and methods.

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

The TS Skeleton Template as provided in Annex A should be used as a starting point when drafting 5GC SBI API specifications.

Common procedures, HTTP extensions and error handling applicable to several 5GC SBI API specifications should be defined in 3GPP TS 29.500 [2] and should be referenced from individual 5GC SBI API specifications.

Common data types applicable to several 5GC SBI API specifications should be defined in 3GPP TS 29.571 [5] and should be referenced from individual 5GC SBI API specifications.

4.2 API Design Style and REST Implementation Levels

4.2.1 General

5GC SBI API specifications should apply a protocol design framework as follows:

a) REST-style service operations should implement the Level 2 of the Richardson maturity model, with standard HTTP methods, whenever it is a good match for the style of interaction to model, e.g. service operations that can naturally map to one of the standard methods (CRUD operations), this should be the preferred modelling attempt;

b) service operations may use custom API operations (RPC-style interaction), when it is seen a better fit for the style of interaction to model, e.g. non-CRUD service operations;

c) it is possible to mix REST-style operations and RPC-style operations in the same API.

NOTE: Level 3 (HATEOAS) of the Richardson maturity model in the 5G Service-Based Architecture can be implemented by an API but is optional. Hypermedia usage guidelines are provided in clause 4.7 of the present specification.

4.2.2 API Design Principles for Query Operation

When designing a query operation API, i.e. the NF service consumer invokes the API aiming to retrieve certain information from the NF service producer, the following principles should be applied:

a) if the query operation does not require any input parameter for the NF service producer, then the REST-style service operation with standard HTTP GET method should be used (see clause 4.6.1.1.2);

b) if

- the query operation requires input parameter(s) for the NF service producer; and
- all the required input parameter(s) are used to identify a particular resource and/or control the content of the result of the query operation;

then the REST-style service operation with standard HTTP GET method should be used (see clause 4.6.1.1.2);

c) standard HTTP GET method shall not be used for non-safe operations and non-idempotent operations.

4.2.3 API Design Principles for Delete Operation

When designing a delete operation API, i.e. the NF service consumer invokes the API aiming to delete certain resource on the NF service producer, the following principles should be applied:

a) if the delete operation does not require any input parameter for the NF service producer, then the REST-style service operation with standard HTTP DELETE method should be used (see clause 4.6.1.1.4);

b) if

- the delete operation requires input parameter(s) for the NF service producer; and
- all the required input parameter(s) are used to identify a particular resource and/or control the content of the result of the delete operation;

then the REST-style service operation with standard HTTP DELETE method should be used (see clause 4.6.1.1.4);

c) standard HTTP DELETE method shall not be used for non-idempotent operations.

4.3 Version Control

4.3.0 General

The version control mechanism in the present clause allows the management of changes to an API and provides a version number that is incremented whenever changes to the API are applied.

NOTE: The version number does not reflect the usage of optional features. A mechanism to negotiate the usage of optional features is defined in clause 6.6 of 3GPP TS 29.500 [2].

4.3.1 Structure of API version numbers

4.3.1.1 API version number format

API version numbers shall consist of at least 3 fields, following a MAJOR.MINOR.PATCH pattern according to the Semantic Versioning Specification [17] with exceptions for 3GPP Releases under development. A fourth DRAFT field is added to denote an OpenAPI version under development i.e., prior to the freeze of the corresponding OpenAPI description for a given 3GPP Release. Optionally, additional fields can be added after those fields based on operator policy.

The 1st field (MAJOR), the 2nd field (MINOR), and the 3rd field (PATCH) shall contain unsigned integer numbers.

During the development of an API (i.e. before the freeze of a given 3GPP Release), the 4th field is called DRAFT, and it shall have the format "alpha-*n*", where "*n*" is an unsigned integer number.

After the freeze of a 3GPP Release, the optional 4th field shall not be considered as DRAFT and it may contain any string, with a format other than "alpha-*n*"; any additional optional field(s), when present, may contain any string.

The fields shall be separated by ".".

EXAMPLE: "1.0.0.alpha-1".

4.3.1.2 Rules for incrementing field values

The first version of a new API under development shall obtain the version number "1.0.0.alpha-1". At the first publication of the 3GPP Technical Specification defining the API after the OpenAPI freeze of the first 3GPP Release that contains the API, the version number of the API shall be set to "1.0.0".

When a new version of the 3GPP TS containing OpenAPI file(s) is published, the fields of the corresponding API version number(s) shall be incremented according to the following rules:

1st Field (MAJOR):

- This numerical field shall be incremented when:

  a)- there are one or more backward incompatible changes to the API after the OpenAPI freeze for a given 3GPP Release; and

b) there are the first backward incompatible change(s) to the existing API with respect to the latest version in the previous 3GPP Release while a 3GPP Release is under development (i.e. prior to the OpenAPI freeze for a given 3GPP Release).

EXAMPLE 1: Assuming that 3GPP Rel-16 under development contains API version "1.1.0.alpha-2", and a backward incompatible change with respect to the latest version in the previous 3GPP Release is applied to that API before the OpenAPI freeze, the new Rel-16 API version is "2.0.0.alpha-1".

NOTE 1: Subsequent changes in a given 3GPP Release under development do not lead to increment of the 1st field (MAJOR) and 2nd field (MINOR).

NOTE 2: Rules for determining backward incompatible changes are provided in Annex B.

NOTE 3: It is recommended to avoid backward incompatible change to the API after the OpenAPI freeze whenever possible, especially after OpenAPI freeze of a succeeding Release. It is preferable to introduce such changes only in the 3GPP Release under development.

- If a backward incompatible change needs to be applied to several 3GPP Releases the following applies:

a) If the 3GPP Releases contain different MAJOR versions of the same API, a new MAJOR API version shall be assigned to each 3GPP Release in the order of those 3GPP Releases in such a manner that the lowest of those 3GPP Releases shall obtain the first unassigned MAJOR version value.

EXAMPLE 2: Assuming that 3GPP Rel-15 contains API version "1.0.0", and Rel-16 contains API version "2.0.0", and that the same backward incompatible change is applied to that API in both Releases, the new Rel-15 API version is "3.0.0" and the new Rel-16 API version is "4.0.0".

b) If the 3GPP Releases contain the same MAJOR version but different MINOR versions of the same API, a single new MAJOR API version value shall be assigned for all those 3GPP Releases, unless other backward incompatible changes only applied to some of those Releases require the creation of separate MAJOR versions.

NOTE 4: For each such Release a new MINOR version is assigned.

EXAMPLE 3: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0", and Rel-17 contains API version "1.2.0", and that the same backward incompatible change is applied to that API in all 3GPP Releases, the new 3GPP Rel-15 and Rel-16 API version is "2.0.0" and the new 3GPP Rel-17 API version is "2.2.0".

c) If the 3GPP Releases contain the same API versions, a single new API version shall be assigned for all those 3GPP Releases, unless other changes only applied to some of those Releases require the creation of separate versions.

EXAMPLE 4: Assuming that 3GPP Rel-15 and 3GPP Rel-16 contain API version "1.0.0", and that only the same backward incompatible change is applied to that API in both 3GPP Releases, the new 3GPP Rel-15 and Rel-16 API version is "2.0.0".

EXAMPLE 5: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0", and that the same backward incompatible change is applied to that API in both Releases and an additional backward compatible change is applied in 3GPP Rel-16, the new 3GPP Rel-15 API version is "2.0.0", and the 3GPP Rel-16 API version is "2.1.0".

EXAMPLE 6: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0", and that the same backward incompatible change is applied to that API in both Releases and an additional backward incompatible change is applied in 3GPP Rel-16, the new 3GPP Rel-15 API version is "2.0.0", and the 3GPP Rel-16 API version is "3.0.0".

2nd Field (MINOR):

- This numerical field shall be incremented when:

a) there are the first one or more backward compatible changes not corresponding to changes to earlier 3GPP Releases (i.e. changes introduced by 3GPP CR with other categories than "mirror") to the same API in a given 3GPP Release without any prior backward incompatible changes in that Release. If the same 1st field (MAJOR) and the 2nd field (MINOR) are assigned to *n* previous 3GPP Releases, a

MINOR version number shall be reserved for each intermediate 3GPP Release for possible subsequent changes in that Release and the MINOR version number shall be incremented by *n*; and

EXAMPLE 7: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0" (because there were no changes to the API in Rel-16), and in Rel-17 the first backward compatible new feature is added before the OpenAPI freeze, the API version "1.2.0.alpha-1" is assigned to Rel-17.

b) there are one or more subsequent backward compatible additions of features not corresponding to changes to previous 3GPP Releases to the API in a frozen 3GPP Release before a higher MINOR number has been allocated for the same MAJOR version (for a subsequent Release).

- This field shall be reset to "0" if the 1st field (MAJOR) is changed, unless a backward incompatible change needs to be applied to several 3GPP Releases that already contain the same MAJOR but different MINOR API versions. In that case a single new major API version is assigned, and for each such 3GPP Release with an own MINOR version, a new MINOR version shall be assigned, starting with MINOR version "0" for the lowest such Release, and reserving a MINOR version number for each intermediate Release without an own MINOR version. (see Example 3)

NOTE 5: In most cases the MINOR version is incremented when new backward compatible features are added in a 3GPP Release. In rare cases, where only backward compatible changes not corresponding to changes to previous 3GPP Releases are applied to a 3GPP Release, the MINOR version is also incremented. It is recommended to avoid such changes in 3GPP Releases without added functionality whenever possible.

NOTE 6: Subsequent backward compatible changes in a given 3GPP Release before OpenAPI freeze do not lead to an increment of the 2nd field (MINOR).

NOTE 7: Changes corresponding to changes in previous 3GPP Releases do not lead to an increment of the 2nd field (MINOR).

NOTE 8: If two 3GPP Releases are under parallel development (because the work on Rel-*X+1* has commenced before the OpenAPI freeze of Rel-*X*), the corresponding APIs will obtain distinct values of the 1st field (MAJOR) or 2nd field (MINOR).

EXAMPLE 8: Assuming that an API was introduced with version "1.0.0" in Rel-15, and that the Rel-16 version is "1.1.0.alpha-5" because the OpenAPI is not yet frozen in Rel-16, and that a new backward compatible Rel-17 feature is added, the Rel-17 API version is "1.2.0.alpha-1".

3rd Field (PATCH):

- This numerical field shall be incremented:

a) if the changes are only one or more backward-compatible corrections (but no changes requiring an update of the 1st field (MAJOR) or of the 2nd field (MINOR) are made to the API after the OpenAPI freeze of a 3GPP Release; and

b) if one or more backward compatible additions of features, but no changes requiring an update of the 1st field (MAJOR) or of the 2nd field (MINOR), are made to the API after the OpenAPI freeze of a 3GPP Release and after the assignment of a MINOR version to a higher 3GPP Release.

- This field shall be reset to "0" if the 1st field (MAJOR) or 2nd field (MINOR) is changed.

NOTE 9: Before the OpenAPI freeze for a given 3GPP Release, the 3rd field will not be incremented.

NOTE 10: If the 1st field (MAJOR) and 2nd field (MINOR) were not incremented between 3GPP Releases (because there were no added features and no backward incompatible changes), and the same backward compatible changes are then applied to those 3GPP Releases, the API files in those 3GPP Releases are identical and will obtain the same API version number.

NOTE 11: In rare cases for which a new backward compatible functionality needs to be added in an older 3GPP Release after the OpenAPI freeze and work on that API already started in a later Release, the new functionality is exceptionally introduced as a PATCH correction and a new supported feature could be defined accordingly.

4th Field:

- Before the OpenAPI freeze of a 3GPP Release, the 4th field (DRAFT) shall be supplied as follows: