ETSI GS MEC 009 V3.1.1 (2021-06)

Multi-access Edge Computing (MEC); General principles, patterns and common aspects of MEC Service APIs

Disclaimer

The present document has been produced and approved by the Multi-access Edge Computing (MEC) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGS/MEC-0009v311ApiPrinciples

Keywords
API, MEC

ETSI

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

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

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

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

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Multi-access Edge Computing (MEC).

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.

1 Scope

The present document defines design principles for RESTful MEC service APIs, provides guidelines and templates for the documentation of these, and defines patterns of how MEC service APIs use RESTful principles.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

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

NOTE: Available at https://tools.ietf.org/html/rfc7231.

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

NOTE: Available at https://tools.ietf.org/html/rfc7232.

[3] IETF RFC 5789: "PATCH Method for HTTP".

NOTE: Available at https://tools.ietf.org/html/rfc5789.

[4] IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".

NOTE: Available at https://tools.ietf.org/html/rfc6901.

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

NOTE: Available at https://tools.ietf.org/html/rfc7396.

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

NOTE: Available at https://tools.ietf.org/html/rfc6902.

[7] IETF RFC 5261: "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors".

NOTE: Available at https://tools.ietf.org/html/rfc5261.

[8] IETF RFC 6585: "Additional HTTP Status Codes".

NOTE: Available at https://tools.ietf.org/html/rfc6585.

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

NOTE: Available at https://tools.ietf.org/html/rfc63986.

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

NOTE: Available at https://tools.ietf.org/html/rfc8259.

[11] W3C Recommendation 16 August 2006: "Extensible Markup Language (XML) 1.1" (Second Edition).

NOTE: Available at https://www.w3.org/TR/2006/REC-xml11-20060816/.

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

NOTE: Available at https://tools.ietf.org/html/rfc8288.

[13] Void.

[14] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

NOTE: Available at https://tools.ietf.org/html/rfc5246.

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

NOTE: Available at https://tools.ietf.org/html/rfc7807.

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

NOTE: Available at https://tools.ietf.org/html/rfc6749.

[17] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

NOTE: Available at https://tools.ietf.org/html/rfc6750.

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

NOTE: Available at https://tools.ietf.org/html/rfc7540.

[19] Void.

[20] IETF RFC 3339: "Date and Time on the Internet: Timestamps".

NOTE: Available at https://tools.ietf.org/html/rfc3339.

[21] IETF RFC 4918: "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)".

NOTE: Available at https://tools.ietf.org/html/rfc4918.

[22] IETF RFC 7233: " Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

NOTE: Available at https://tools.ietf.org/html/rfc7233.

[23] IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

NOTE: Available at https://tools.ietf.org/html/rfc7235.

[24] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

NOTE: Available at https://tools.ietf.org/html/rfc8446.

[25] IETF RFC 6455: "The WebSocket Protocol".

NOTE: Available at https://tools.ietf.org/html/rfc6455.

[26] ETSI TS 129 122: "Universal Mobile Telecommunications System (UMTS); LTE; 5G; T8 reference point for Northbound APIs" (3GPP TS 29.122).

[27] ETSI TS 133 210: "Universal Mobile Telecommunications System (UMTS); LTE; 3G Security; Network Domain Security (NDS); IP network layer security (3GPP TS 33.210)".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS MEC 001: "Multi-access Edge Computing (MEC); Terminology".

[i.2] William Durand: "Please. Don't Patch Like An Idiot".

NOTE: Available at http://williamdurand.fr/2014/02/14/please-do-not-patch-like-an-idiot/.

[i.3] Martin Fowler: "Richardson Maturity Model: steps toward the glory of REST".

NOTE: Available at http://martinfowler.com/articles/richardsonMaturityModel.html.

[i.4] JSON Schema, Draft Specification 2020-12, December 8, 2020.

NOTE: Referenced version available as Internet Draft (work in progress) at https://tools.ietf.org/html/draft-bhutton-json-schema-00. All versions are available at http://json-schema.org/specification.html.

[i.5] W3C® Recommendation: "XML Schema Part 0: Primer Second Edition".

NOTE: Available at https://www.w3.org/TR/xmlschema-0/.

[i.6] ETSI GS MEC 011: "Multi-access Edge Computing (MEC); Edge Platform Application Enablement".

[i.7] ETSI GS MEC 012: "Multi-access Edge Computing (MEC); Radio Network Information API".

[i.8] IANA: "Hypertext Transfer Protocol (HTTP) Status Code Registry".

NOTE: Available at http://www.iana.org/assignments/http-status-codes.

[i.9] MQTT Version 3.1.1 Plus Errata 01, OASIS™ Standard, 10 December 2015.

NOTE: Available at http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[i.10] Apache Kafka®.

NOTE: Available at https://kafka.apache.org/.

[i.11] gRPC®.

NOTE: Available at http://www.grpc.io/.

[i.12] Protocol buffers.

NOTE: Available at https://developers.google.com/protocol-buffers/.

[i.13] IETF RFC 7519: "JSON Web Token (JWT)".

NOTE: Available at https://tools.ietf.org/html/rfc7519.

[i.14] OpenAPI™ Specification.

NOTE: Available at https://github.com/OAI/OpenAPI-Specification.

[i.15] ETSI TS 129 222 (V16.8.0): "Universal Mobile Telecommunications System (UMTS); LTE; 5G; T8 reference point for Northbound APIs (3GPP TS 29.122 version 16.8.0 Release 16)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS MEC 001 [i.1] and the following apply:

resource: object with a type, associated data, a set of methods that operate on it, and, if applicable, relationships to other resources

NOTE: A resource is a fundamental concept in a RESTful API. Resources are acted upon by the RESTful API using the Methods (e.g. POST, GET, PUT, DELETE, etc.). Operations on Resources affect the state of the corresponding managed entities.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS MEC 001 [i.1] and the following apply:

AA	Authentication and Authorization
API	Application Programming Interface
BYOT	Bring Your Own Transport
CRUD	Create, Read, Update, Delete
DDoS	Distributed Denial of Service
DN	Distinguished Name
GS	Group Specification
HATEOAS	Hypermedia As The Engine Of Application State
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
ISG	Industry Specification Group
JSON	JavaScript Object Notation
JWT	JSON Web Token
MEC	Multi-access Edge Computing
PKI	Public Key Infrastructure
POX	Plain Old XML
REST	Representational State Transfer
RFC	Request For Comments
RPC	Remote Procedure Call
SCS/AS	Services Capability Server/Application Server
SCEF	Service Capability Exposure Function
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Indicator
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

4 Design principles for developing RESTful MEC service APIs

4.1 REST implementation levels

The Richardson Maturity Model as defined in [i.3] breaks down the principal elements of a REST approach into three steps.

All RESTful MEC service APIs shall implement at least Level 2 of the Richardson Maturity Model explained in annex C.

It is recommended to implement Level 3 when applicable.

4.2 General principles

RESTful MEC service APIs are not technology implementation dependent.

RESTful MEC service APIs embrace all aspects of HTTP/1.1 (IETF RFC 7231 [1]) including its request methods, response codes, and HTTP headers. Support for PATCH (IETF RFC 5789 [3]) is optional.

For each RESTful MEC service API specification, the following information should be included:

- Purpose of the API.
- URIs of resources including version number.
- HTTP methods (IETF RFC 7231 [1]) supported.
- Representations supported: JSON and, if applicable, XML.
- Response schema(s).
- Request schema(s) when PUT, POST, or PATCH are supported.
- Links supported (Optional in Level 2 APIs).
- Response status codes supported.

Since the release of HTTP/1.1, major revisions have been introduced through HTTP/2 (IETF RFC 7540 [18]) including binary serialization in place of textual, single TCP connection, full multiplexing, header compression and server push. MEC system deployments may utilize HTTP/2. However, this is transparent to the RESTful MEC service APIs, as the main semantic of HTTP has been retained in HTTP/2 thereby providing backwards compatibility. If HTTP/2 (IETF RFC 7540 [18]) is supported, its use shall be negotiated as specified in section 3 of IETF RFC 7540 [18].

4.3 Entry point of a RESTful MEC service API

Entry point for a RESTful MEC service API:

- Needs to have one and exactly one entry point. The URI of the entry point needs to be communicated to API clients so that they can find the API.
- It is common for the entry point to contain some or all of the following information:
  - Information on API version, supported features, etc.
  - A list of top-level collections.
  - A list of singleton resources.

  - Any other information that the API designer deemed useful, for example a small summary of operating status, statistics, etc.

- It can be made known via different means:
  - Client discovers automatically the entry point and meaning of the API.
  - Client developer knows about the API at time of client development.

4.4 API security and privacy considerations

To allow proactive protection of the APIs against the known security and privacy issues, e.g. DDoS, frequency attack, unintended or accidental information disclosure, etc. the design for a secure API should consider at least the following aspects:

- Ability to control the frequency of the API calls (calls/min), e.g. by supporting the definition of a validity time or expiration time for a service response.
- Anonymization of the real identities.
- Authorization of the applications based on the sensitivity of the information exposed through the API.

5 Documenting RESTful MEC service APIs

5.1 RESTful MEC service API template

Annex D defines a template for the documentation of RESTful MEC service APIs. Examples how to use this template can for instance be found in ETSI GS MEC 011 [i.6] and ETSI GS MEC 012 [i.7].

5.2 Conventions for names

5.2.1 Case conventions

The following case conventions for names and strings are used in the RESTful MEC service APIs:

1) UPPER_WITH_UNDERSCORE

   All letters of a string are capital letters. Digits are allowed but not at the first position. Word boundaries are represented by the underscore "_" character. No other characters are allowed.

EXAMPLE 1:

   a) ETSI_MEC_MANAGEMENT;

   b) MULTI_ACCESS_EDGE_COMPUTING.

2) lower_with_underscore

   All letters of a string are lowercase letters. Digits are allowed but not at the first position. Word boundaries are represented by the underscore "_" character. No other characters are allowed.

EXAMPLE 2:

   a) etsi_mec_management;

   b) multi_access_edge_computing.

3) UpperCamel

A string is formed by concatenating words. Each word starts with an uppercase letter (this implies that the string starts with an uppercase letter). All other letters are lowercase letters. Digits are allowed but not at the first position. No other characters are allowed. Abbreviations follow the same scheme (i.e. first letter uppercase, all other letters lowercase).

EXAMPLE 3:

a) EtsiMecManagement;

b) MultiAccessEdgeComputing.

4) lowerCamel

As UpperCamel, but with the following change: The first letter of a string shall be lowercase (i.e. the first word starts with a lowercase letter).

EXAMPLE 4:

a) etsiMecManagement;

b) multiAccessEdgeComputing.

5.2.2 Conventions for URI parts

5.2.2.1 Introduction

Based on IETF RFC 3986 [9], the parts of the URI syntax that are relevant in the context of the RESTful MEC service APIs are as follows:

- *Path*, consisting of *segments*, separated by "/" (e.g. segment1/segment2/segment3).
- *Query*, consisting of pairs of parameter name and value (e.g. ?org=etsi&isg=mec, where two pairs are presented).

5.2.2.2 Path segment naming conventions

a) Each path segment of a resource URI which represents a constant string shall use lower_with_underscore. Only letters, digits and underscore "_" are allowed.

EXAMPLE 1: etsi_mec_management

b) If a resource represents a collection of entities, and the last path segment of that resource's URI is a string constant, the last path segment shall be plural.

EXAMPLE 2: …/prefix/api/v1/user**s**

c) If a resource is not a task resource and the last path segment of that resource's URI is a string constant, the last path segment should be a (composite) noun.

EXAMPLE 3: …/prefix/api/v1/users

d) For resources that are task resources, the last path segment of the resource URI should be a verb, or at least start with a verb.

EXAMPLE 4:

…/app_instances/{appInstanceId}/instantiate

…/app_instances/{appInstanceId}/do_something_else

e) A name that represents a URI path segment or multiple URI path segments in the API documentation but serves as a placeholder for an actual value created at runtime (URI path variable) shall use lowerCamel, and shall be surrounded by curly brackets.

EXAMPLE 5: {appInstanceId}

f) Once a variable is replaced at runtime by an actual string, the string shall follow the rules for a path segment or sequence of path segments (depending on whether the variable represents a single path segment or a sequence thereof) defined in IETF RFC 3986 [9]. IETF RFC 3986 [9] disallows certain characters from use in a path segment. Each actual RESTful MEC service API specification shall define this restriction to be followed when generating values for path segment variables, or propose a suitable encoding (such as percent-encoding according to IETF RFC 3986 [9]), to escape such characters if they can appear in input strings intended to be substituted for a path segment variable.

5.2.2.3 Query naming conventions

a) Parameter names in queries shall use lower_with_underscore.

EXAMPLE 1: ?isg_name=MEC

b) Variables that represent actual parameter values in queries shall use lowerCamel and shall be surrounded by curly brackets.

EXAMPLE 2: ?isg_name={chooseAName}

c) Once a variable is replaced at runtime by an actual string, the convention defined in clause 5.2.2.2 item f) applies to that string.

5.2.3 Conventions for names in data structures

The following syntax conventions apply when defining the names for attributes and parameters in the RESTful MEC service API data structures:

a) Names of attributes/parameters shall be represented using lowerCamel.

EXAMPLE 1: appName.

b) Names of arrays and maps (i.e. those with cardinality 1..N or 0..N) shall be plural rather than singular.

EXAMPLE 2: users, mecApps.

c) The identifier of a data structure via which this data structure can be referenced externally should be named "id".

d) Each value of an enumeration types shall be represented using UPPER_WITH_UNDERSCORE.

EXAMPLE 3: NOT_INSTATIATED.

e) The names of data types shall be represented using UpperCamel.

EXAMPLE 4: ResourceHandle, AppInstance.

5.3 Provision of an OpenAPI definition

An ETSI ISG MEC GS defining a RESTful MEC service API should provide a supplementary description file (or supplementary description files) compliant to the OpenAPI specification [i.14], which inherently include(s) a definition of the data structures of the API in JSON schema or YAML format. A description file is machine readable facilitating content validation and autocreation of stubs for both the service client and server. A link to the specific repository containing the file(s) shall be provided. All API repositories can be accessed from https://forge.etsi.org. The file (or files) shall be informative. In case of a discrepancy between supplementary description file(s) and the underlying specification, the underlying specification shall take precedence.

5.4 Documentation of the API data model

5.4.1 Overview

Clause 5.4 and its clauses specify provisions for API data model documentation for ETSI ISG MEC GSs defining RESTful MEC service APIs. Clause 5 in annex D provides a related data model template.

The data model shall be defined using a tabular format as described in the following clauses. The name of the data type shall be documented appropriately in the heading of the clause and in the caption of the table, preferably as defined in clause 5.2.2 and in annex D.

5.4.2 Structured data types

Structured data types shall be documented in tabular format, as in table 5.4.2-1 (one table per named data type).

Table 5.4.2-1: Template for a table defining a named structured data type

Attribute name	Data type	Cardinality	Description

The following provisions apply to the content of the table:

1) "Attribute name" shall provide the name of the attribute in lowerCamel.

2) "Data type" shall provide one of the following:

   a) The name of a **named data type** (structured, simple or enum) that is defined elsewhere in the document where the data type is specified, or in a referenced document. In case of a referenced type from another document, a reference to the defining document should be included in the "Description" column unless included in a global clause.

   b) An indication of the definition of an **inlined** nested **structure**. In case of inlining a structure, the "Data type" column shall contain the string "Structure (inlined)", and all attributes of the inlined structure shall be prefixed with one or more closing angular brackets ">", where the number of brackets represents the level of nesting.

   c) An indication of the definition of an **inlined enumeration type**. In case of inlining an enumeration type, the "Data type" column shall contain the string "Enum (inlined)", and the "Description" column shall contain the allowed values and (optionally) their meanings. There are two possible ways to define enums (see clause 5.4.4): just to define the valid enum values or to define the valid values and their mapping to integers. It is good practice not to mix the two patterns in the same data structure.

3) If the maximum cardinality is greater than one, "Data type" may indicate the format of the list of values. If it is an array, the format of that list should be indicated by using the key word "array(<type>)". If it is a map, the format shall be indicated by using the key word "map(<type>)". In both cases, <type> indicates the data type of the individual list entries. In case neither "map" nor "array" is given and the maximum cardinality is greater than one, "array" is assumed as default. The presence or absence of the indication of "array" should be consistent between all data types in the scope of an API.

4) "Cardinality" defines the allowed number of occurrences, either as a single value, or as two values indicating lower bound and upper bound, separated by "..". A value shall be either a non-negative integer number or an uppercase letter that serves as a placeholder for a variable number (e.g. N).

5) "Description" describes the meaning and use of the attribute and may contain normative statements. In case of an inlined enumeration type, the "Description" column shall define the allowed values and (optionally) their meanings, as follows: "Permitted values:" on one line, followed by one paragraph of the following format for each value: "- <VAL>: <Meaning of the value>".

Table 5.4.2-2 provides an example.