ETSI GS MEC 009 V2.1.1 (2019-01)

Multi-access Edge Computing (MEC); General principles for MEC Service APIs

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fb1e962-2267-4f0e-9352-e4ccd6884cbd/etsi-gs-mec-009-v2.1.1-2019-01

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-0009v211ApiPrinciples

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 - 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

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2019.
All rights reserved.

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/4f8062-22e7-4be-35-etsi-gs-mec-009-v2.1.1-2019-01

Contents

Intellectual Property Rights 6
Foreword 6
Modal verbs terminology 6
1 Scope 7
2 References 7
2.1 Normative references 7
2.2 Informative references 8
3 Definition of terms, symbols and abbreviations 9
3.1 Terms 9
3.2 Symbols 9
3.3 Abbreviations 10
4 Design principles for developing RESTful MEC service APIs 10
4.1 REST implementation levels 10
4.2 General principles 10
4.3 Entry point of a RESTful MEC service API 11
4.4 API security and privacy considerations 11
5 Documenting RESTful MEC service APIs 11
5.1 RESTful MEC service API template 11
5.2 Conventions for names 12
5.2.1 Case conventions 12
5.2.2 Conventions for URI parts 12
5.2.2.1 Introduction 12
5.2.2.2 Path segment naming conventions 13
5.2.2.3 Query naming conventions 13
5.2.3 Conventions for names in data structures 13
5.3 Provision of an OpenAPI definition 14
6 Patterns of RESTful MEC service APIs 14
6.1 Introduction 14
6.2 void 14
6.3 Pattern: Resource identification 14
6.3.1 Description 14
6.3.2 Resource definition(s) and HTTP methods 14
6.4 Pattern: Resource representations and content format negotiation 15
6.4.1 Description 15
6.4.2 Resource definition(s) and HTTP methods 15
6.4.3 Resource representation(s) 15
6.4.4 HTTP headers 15
6.4.5 Response codes and error handling 16
6.5 Pattern: Creating a resource (POST) 16
6.5.1 Description 16
6.5.2 Resource definition(s) and HTTP methods 16
6.5.3 Resource representation(s) 17
6.5.4 HTTP headers 17
6.5.5 Response codes and error handling 17
6.5a Pattern: Creating a resource (PUT) 17
6.5a.1 Description 17
6.5a.2 Resource definition(s) and HTTP methods 18
6.5a.3 Resource representation(s) 18
6.5a.4 HTTP headers 18
6.5a.5 Response codes and error handling 18
6.6 Pattern: Reading a resource 18
6.6.1 Description 18
6.6.2 Resource definition(s) and HTTP methods 19

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fbc8062-2267-4f0e-9352-e4ccd6884cbd/etsi-gs-mec-009-v2.1.1-2019-01

6.6.3 Resource representation(s) ... 19
6.6.4 HTTP headers ... 19
6.6.5 Response codes and error handling ... 19
6.7 Pattern: Queries on a resource ... 19
6.7.1 Description ... 19
6.7.2 Resource definition(s) and HTTP methods ... 20
6.7.3 Resource representation(s) ... 20
6.7.4 HTTP headers ... 20
6.7.5 Response codes and error handling ... 20
6.8 Pattern: Updating a resource (PUT) ... 20
6.8.1 Description ... 20
6.8.2 Resource definition(s) and HTTP methods ... 21
6.8.3 Resource representation(s) ... 21
6.8.4 HTTP headers ... 21
6.8.5 Response codes and error handling ... 22
6.9 Pattern: Updating a resource (PATCH) ... 22
6.9.1 Description ... 22
6.9.2 Resource definition(s) and HTTP methods ... 23
6.9.3 Resource representation(s) ... 23
6.9.4 HTTP headers ... 24
6.9.5 Response codes and error handling ... 24
6.10 Pattern: Deleting a resource ... 24
6.10.1 Description ... 24
6.10.2 Resource definition(s) and HTTP methods ... 25
6.10.3 Resource representation(s) ... 25
6.10.4 HTTP headers ... 25
6.10.5 Response codes and error handling ... 25
6.11 Pattern: Task resources ... 26
6.11.1 Description ... 26
6.11.2 Resource definition(s) and HTTP methods ... 26
6.11.3 Resource representation(s) ... 26
6.11.4 HTTP headers ... 26
6.11.5 Response codes and error handling ... 26
6.12 Pattern: REST-based subscribe/notify ... 27
6.12.1 Description ... 27
6.12.2 Resource definition(s) and HTTP methods ... 28
6.12.3 Resource representation(s) ... 29
6.12.4 HTTP headers ... 29
6.12.5 Response codes and error handling ... 29
6.13 Pattern: Asynchronous operations ... 29
6.13.1 Description ... 29
6.13.2 Resource definition(s) and HTTP methods ... 31
6.13.3 Resource representation(s) ... 31
6.13.4 HTTP headers ... 32
6.13.5 Response codes and error handling ... 32
6.14 Pattern: Links (HATEOAS) ... 32
6.14.1 Description ... 32
6.14.2 Resource definition(s) and HTTP methods ... 32
6.14.3 Resource representation(s) ... 32
6.14.4 HTTP headers ... 34
6.14.5 Response codes and error handling ... 34
6.15 Pattern: Error responses ... 34
6.15.1 Description ... 34
6.15.2 Resource definition(s) and HTTP methods ... 34
6.15.3 Resource representation(s) ... 34
6.15.4 HTTP headers ... 35
6.15.5 Response codes and error handling ... 35
6.16 Pattern: Authorization of access to a RESTful MEC service API using OAuth 2.0 ... 35
6.16.1 Description ... 35
6.16.2 Resource definition(s) and HTTP methods ... 38
6.16.3 Resource representation(s) ... 39
6.16.4 HTTP headers ... 39

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fbc8062-2267-4fe-9352-e4ed6884cbd/etsi-gs-mec-009-v2.1.1-2019-01

6.16.5 Response codes and error handling 39
6.16.6 Discovery of the parameters needed for exchanges with the token endpoint 39
6.16.7 Scope values 39
6.17 Pattern: Representation of lists in JSON 39
6.17.1 Description 39
6.17.2 Representation as arrays 39
6.17.3 Representation as maps 40
6.18 Pattern: Attribute selectors 40
6.18.1 Description 40
6.18.2 Resource definition(s) and HTTP methods 40
6.18.3 Resource representation(s) 41
6.18.4 HTTP headers 42
6.18.5 Response codes and error handling 42
6.19 Pattern: Attribute-based filtering 42
6.19.1 Description 42
6.19.2 Resource definition(s) and HTTP methods 43
6.19.3 Resource representation(s) 44
6.19.4 HTTP headers 44
6.19.5 Response codes and error handling 45

7 Alternative transport mechanisms 45
7.1 Description 45
7.2 Relationship of topics, subscriptions and access rights 46
7.3 Serializers 47
7.4 Authorization of access to a service over alternative transports using TLS credentials 48

Annex A (informative): REST methods 51

Annex B (normative): HTTP response status codes 52

Annex C (informative): Richardson maturity model of REST APIs 54

Annex D (informative): RESTful MEC service API template 55

History 64

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fac8062-2267-4f0e-9352-e4ccd6884cbd/etsi-gs-mec-009-v2.1.1-2019-01

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.

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/6fbc8062-2267-4f0e-9332-e4ccd6e84cbd/etsi-gs-mec-009-v2.1.1-2019-01

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fbc8063-2267-4f0e-9257-e4e6168853bd/etsi-gs-mec-009-v2.1.1-2019-01

[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 5988: "Web Linking".

NOTE: Available at https://tools.ietf.org/html/rfc5988.

[13] IETF RFC 2818: "HTTP Over TLS".

NOTE: Available at https://tools.ietf.org/html/rfc2818.

[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] IETF RFC 7301: "Transport Layer Security (TLS): Application-Layer Protocol Negotiation Extension".

NOTE: Available at https://tools.ietf.org/html/rfc7301.

[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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/6fbc8062-2267-4f0e-9352-4ccd8884cbd/etsi-gs-mec-009-v2.1.1-2019-01

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 v7, November 19, 2017.

NOTE: Available at http://json-schema.org/specification.html. Also available as Internet Draft (work in progress) from https://tools.ietf.org/html/draft-handrews-json-schema-00.

[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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fbc8062-2267-4f0e-9352-04ccd6884c6d/etsi-gs-mec009-v2.1.1-2019-01

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:

API	Application Programming Interface
BYOT	Bring Your Own Transport
CRUD	Create, Read, Update, Delete
DDoS	Distributed Denial of Service
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
MEC	Multi-access Edge Computing
POX	Plain Old XML
REST	Representational State Transfer
RFC	Request For Comments
RPC	Remote Procedure Call
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Indicator
XML	eXtensible Markup Language

4 Design principles for developing RESTful MEC service APIs

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fbc8062-2267-4f0e-9352-44ccd0f84cbd/etsi-gs-mec-009-v2.1.1-2019-01

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0bc8062-2267-
360-93527cccd6884ed/etsi-gs-mec-009-v2.1.1-2019-01

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).

iTeh STANDARD PREVIEW
(standards.iteh.ai)
Full standard:
https://standards.iteh.ai/catalog/standards/sist/0fbc8062-2267-4f0e-9352-e4ccd6884cbd/etsi-gs-mec-009-v2.1.1-2019-01

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ETSI GS MEC 009 V2.1.1 (2019-01)
https://standards.iteh.ai/catalog/standards/sist/0bc6b2-2267-4f8-0352-e4cc65/etsi-gs-mec-009-v2.1.1-2019-01

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.