Multi-access Edge Computing (MEC);
IoT API

Disclaimer: This DRAFT is a working document of ETSI ISG MEC. It is provided for information only and is still under development within ETSI ISG MEC. ETSI and its Members accept no liability for any further use/implementation of this Specification.

Non-published MEC drafts stored in the "Open Area" are working documents, these may be updated, replaced, or removed at any time

Do not use as reference material.

Do not cite this document other than as "work in progress".

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.

Approved and published Specifications and reports for implementation of the MEC system shall be obtained via the ETSI Standards search page at: http://www.etsi.org/standards-search
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intellectual Property Rights</td>
<td>5</td>
</tr>
<tr>
<td>Foreword</td>
<td>5</td>
</tr>
<tr>
<td>Modal verbs terminology</td>
<td>5</td>
</tr>
<tr>
<td>Scope</td>
<td>6</td>
</tr>
<tr>
<td>References</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Normative references</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Informative references</td>
<td>6</td>
</tr>
<tr>
<td>3 Definition of terms, symbols and abbreviations</td>
<td>7</td>
</tr>
<tr>
<td>3.1 Terms</td>
<td>7</td>
</tr>
<tr>
<td>3.2 Symbols</td>
<td>7</td>
</tr>
<tr>
<td>3.3 Abbreviations</td>
<td>7</td>
</tr>
<tr>
<td>4 Overview</td>
<td>7</td>
</tr>
<tr>
<td>5 Description of the service (informative)</td>
<td>8</td>
</tr>
<tr>
<td>5.1 IoT service introduction</td>
<td>8</td>
</tr>
<tr>
<td>5.2 Device provisioning</td>
<td>9</td>
</tr>
<tr>
<td>5.2.1 General</td>
<td>9</td>
</tr>
<tr>
<td>5.2.2 Registered devices query</td>
<td>9</td>
</tr>
<tr>
<td>5.2.3 Device registration</td>
<td>10</td>
</tr>
<tr>
<td>5.2.4 Device registration query</td>
<td>11</td>
</tr>
<tr>
<td>5.2.5 Device registration update</td>
<td>11</td>
</tr>
<tr>
<td>5.2.6 Device deregistration</td>
<td>12</td>
</tr>
<tr>
<td>5.3 IoT service discovery</td>
<td>12</td>
</tr>
<tr>
<td>5.3.1 General</td>
<td>12</td>
</tr>
<tr>
<td>5.3.2 Registered IoT service platforms query</td>
<td>12</td>
</tr>
<tr>
<td>5.3.3 IoT service platform discovery</td>
<td>13</td>
</tr>
<tr>
<td>5.4 Transport configuration</td>
<td>13</td>
</tr>
<tr>
<td>5.4.1 General</td>
<td>13</td>
</tr>
<tr>
<td>5.4.2 User transport query</td>
<td>14</td>
</tr>
<tr>
<td>5.4.3 User transport assignment</td>
<td>14</td>
</tr>
<tr>
<td>6 Data Model</td>
<td>14</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>6.2 Resource data types</td>
<td>15</td>
</tr>
<tr>
<td>6.2.1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>6.2.2 Type: DeviceInfo</td>
<td>15</td>
</tr>
<tr>
<td>6.3 Subscription data types</td>
<td>15</td>
</tr>
<tr>
<td>6.3.1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>6.3.y Type: &lt;TypeNameY&gt;</td>
<td>15</td>
</tr>
<tr>
<td>6.4 Notifications data types</td>
<td>15</td>
</tr>
<tr>
<td>6.4.1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>6.4.z Type: &lt;TypeNameZ&gt;</td>
<td>15</td>
</tr>
<tr>
<td>6.5 Referenced structured data types</td>
<td>16</td>
</tr>
<tr>
<td>6.5.1 Introduction</td>
<td>16</td>
</tr>
<tr>
<td>6.5.xx Type: &lt;TypeNameXX&gt;</td>
<td>16</td>
</tr>
<tr>
<td>6.6 Referenced simple data types and enumerations</td>
<td>16</td>
</tr>
<tr>
<td>6.6.1 Introduction</td>
<td>16</td>
</tr>
<tr>
<td>6.6.yy Type: &lt;TypeNameyy&gt;</td>
<td>16</td>
</tr>
<tr>
<td>7 API definition</td>
<td>16</td>
</tr>
<tr>
<td>7.1 Introduction</td>
<td>16</td>
</tr>
<tr>
<td>7.2 Global definitions and resource structure</td>
<td>16</td>
</tr>
<tr>
<td>7.3 Resource: &lt;foo_bar&gt;</td>
<td>18</td>
</tr>
<tr>
<td>7.3.1 Description</td>
<td>18</td>
</tr>
<tr>
<td>7.3.2 Resource definition</td>
<td>18</td>
</tr>
</tbody>
</table>
7.3.3 Resource Methods ......................................................................................................................... 18
7.3.3.1 GET .................................................................................................................................................. 18
7.3.3.2 PUT .................................................................................................................................................. 19
7.3.3.3 PATCH ............................................................................................................................................. 19
7.3.3.4 POST ............................................................................................................................................... 19
7.3.3.5 DELETE .......................................................................................................................................... 19

Annex A (informative): Complementary material for API utilisation ...................................................... 20

History .......................................................................................................................................................... 21
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/](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 the IoT API to assist the deployment and usage of devices that require additional support in a MEC environment, e.g. due to security constraints, limited power, compute and communication capabilities, such as IoT and MTC devices. The API enables the device provisioning and configuration of the associated components and applications requiring connection to these devices.

The present document describes the information flows and the required information. It also specifies the RESTful binding with the data model.

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 http://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] ETSI GS MEC 001: "Multi-access Edge Computing (MEC); Terminology".


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 009: "Multi-access Edge Computing (MEC); General principles for Edge Service APIs".

[i.2] ETSI GS MEC 011: "Multi-access Edge Computing (MEC); Edge Platform Application Enablement".
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 [1] and the following apply:

Definition format

<defined term>: <definition>

example 1: text used to clarify abstract rules by applying them literally

NOTE:  
This may contain additional information.

Editor's note: definition template above to be cleared once the first definition will be incorporated

3.2 Symbols
Void.

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

Abbreviation format

<ACRONYM1> <Explanation>
<ACRONYM2> <Explanation>
<ACRONYM3> <Explanation>

Editor's note: abbreviation template above to be cleared once the first abbreviation will be incorporated

4 Overview

The present document specifies the IoT API to support the deployment in a MEC environment of an IoT service based on components that may or may not support other ETSI MEC functions and interfaces.

Clause 5 introduces how the IoT API may be used by an IoT service administrator to execute configuration, provisioning and enablement tasks for the IoT devices and for the other IoT components in order to properly run in a MEC environment.

The information that can be exchanged over the IoT API is described in clause 6 which provides detailed description on all necessary information elements.

Clause 7 defines the actual IoT API providing detailed information how information elements are mapped into a RESTful API design.
5 Description of the service (informative)

5.1 IoT service introduction

The IoT ecosystem comprises a large number of proprietary and standard architectures, aiming at creating an appropriate service layer for the IoT deployment. The service layer comprises a feature-rich set of functions to enable the communication between the IoT devices and the applications, such functions focusing on device management, security aspects, interconnection of applications, etc.

Edge computing is deemed a key added value for IoT deployment, but the architecture of an IoT service, despite being arbitrarily complex, may not take into account MEC-related aspects.

The IoT service provides means to incorporate heterogeneous IoT frameworks in MEC, and exposes APIs for the MEC platform configuration to facilitate the device provisioning and the configuration of the IoT components running as MEC applications. In addition, the IoT service provides capabilities to discover additional IoT service platforms that will be managed by the MEC platform.

The IoT API aims at facilitating the interworking between a MEC platform and IoT service platforms, providing a means for IoT service administrators to easily configure the MEC platform and the IoT service platform, allowing for, e.g.:

- the provisioning of IoT devices into the MEC system;
- the routing of communications between the devices and the intended IoT service platform;
- the discovery of additional IoT service platforms;
- the enablement of discovery and usability of the IoT service platform's native APIs.

Figure 5.1-1 illustrates a usage example of the IoT API exposed by the MEC platform. On the one hand, the IoT API is used by a MEC platform frontend to receive inputs directly from an IoT service administrator. On the other hand, the IoT API is used by a MEC application to perform automated operations; such MEC application can be i) the end IoT application, ii) the IoT service platform, or iii) a dedicated MEC application, e.g. to complement the functionalities of existing not MEC-compliant components.

![Figure 5.1-1: Usage example of IoT API](image-url)

The IoT API offers the following services to clients:

- IoT service platform discovery.
- Device provisioning.
5.2 Device provisioning

5.2.1 General

The Device provisioning interface enables an IoT Service administrator to manage the association of a device to a particular traffic rule. The device is referred to by using one or a combination of its identifiers, let them be human-defined names or network layer identifiers, e.g. serial number, MAC address, IMEI, etc., whereas the traffic rule can be the one defined in MEC 010-2 or an extension.

This service enables a constrained device to send (receive) a message to (from) an application even if a protocol translation is necessary (provided that the MEC platform supports a similar traffic rule). As an example, this service permits a NB-IoT device in Non-IP Data Delivery (NIDD) mode to exchange messages with a MEC application that operates as an MQTT client.

Clauses 5.2.2 to 5.2.6 describe the operations available for Device provisioning. The related sequence diagrams are presented.

5.2.2 Registered devices query

This operation allows a service consumer to retrieve the information of all the devices currently registered in all or target MEC platforms with a valid traffic rule association. The call flow for this operation is depicted in Figure 5.2.2-1.
Registered devices query consists of the following steps:

1) The service consumer requests the IoT Service to send the list of and the information associated to the registered devices in all MEC platforms with a valid traffic rule association. The request may optionally contain a list of target MEC Platforms to reduce the scope of the query;

2) The IoT Service replies with the list of registered devices (if any) and their associated information.

### 5.2.3 Device registration

This operation allows a service consumer to register a new device, i.e. to create an association between a device and a traffic rule. The call flow for this operation is depicted in Figure 5.2.3-1.

Device registration consists of the following steps:

1) The service consumer requests the IoT Service to register the device, i.e. to create an association between a device and a traffic rule. If known to the service consumer, the request may optionally contain a list of target MEC platforms to which registration should be applied;

2) The IoT Service acknowledges the operation. The response may contain the list of MEC platforms where the device is registered.
5.2.4 Device registration query

This operation allows a service consumer to obtain information about a registered device. The call flow for this operation is depicted in Figure 5.2.4-1.

![Figure 5.2.4-1: Flow of Device registration query](image)

Device registration query consists of the following steps:

1) The service consumer requests the IoT service to send information about a registered device;
2) The IoT Service replies with the information of the registered device (if it exists).

5.2.5 Device registration update

This operation allows a service consumer to update the information about a registered device. The call flow for this operation is depicted in Figure 5.2.5-1.

![Figure 5.2.5-1: Flow of Device registration update](image)

Device registration update consists of the following steps:

1) The service consumer requests the IoT Service to update the information about a registered device. The request may optionally contain a list of target MEC Platforms where the update should be applied;
2) The IoT Service replies with the updated information of the registered device (if it exists).
5.2.6 Device deregistration

This operation allows a service consumer to deregister a device, i.e. to delete an existing association between a device and a traffic rule. The call flow for this operation is depicted in Figure 5.2.6-1.

![Flow of Device deregistration](Image)

Device deregistration consists of the following steps:

1) The service consumer requests the IoT Service to deregister the device, i.e. to delete an existing association between a device and a traffic rule. The request may optionally contain a list of target MEC Platform where the deletion should be applied;

2) The IoT Service acknowledges the request.

5.3 IoT service discovery

5.3.1 General

The IoT service discovery enables the discovery process of IoT service platforms. The discovery process can be executed to obtain the identifier of an IoT service platform. This service enables a MEC application to obtain the IoT service platform reference so as to use IoT service platform's native APIs. As an example, this service allows to discover additional IoT gateways and get information on exposed native APIs.

5.3.2 Registered IoT service platforms query

This operation allows a service consumer to retrieve the information of all the IoT service platforms currently registered in all or target MEC platforms. The call flow for this operation is depicted in Figure 5.3.2-1.

![Flow of Registered IoT service platforms query](Image)

Registered IoT service platforms query consists of the following steps:

1) The service consumer requests the IoT Service to send the list of all registered devices in all MEC platforms. The request may optionally contain a list of target MEC Platforms to reduce the scope of the query.

2) The IoT Service replies with the list of registered IoT service platforms (if any) and their associated references.

Editor's note: A Service consumer must be authorized to discover the service. Details enabling this are TBD.
5.3.3 IoT service platform discovery

This operation allows a service consumer to discover native services running on the IoT service platform. The call flow for this operation is depicted in Figure 5.3.3-1.

![Diagram of IoT service platform discovery](image)

**Figure 5.3.3-1: Flow of IoT service platform discovery**

IoT service platform discovery consists of the following steps:

1) The service consumer requests the IoT Service to send information about a registered IoT Service platform.

The IoT Service replies with the information of the native APIs available at the selected IoT Service platform (if any).

5.4 Transport configuration

5.4.1 General

The Transport configuration interface enables the routing of communications from an IoT device to the intended IoT service platform.

If the discovered IoT service platform is a service-producing MEC app itself, BYOT is leveraged to register the provided user transport on the MEC platform. Otherwise, the registration of the user transport provided by the discovered IoT service platform is performed through the IoT API via the IoT service platform registration procedure. In both cases, the registered transport shall be properly labeled in the MEC platform registry, in order to identify it as the user transport provided by the discovered IoT service platform instance.

As a result, the IoT API is able to route packets from the provisioned IoT devices to the correct user transport provided by the intended IoT service platform.

Clauses 5.4.2 to 5.4.4 describe the operations available for Transport configuration. The related sequence diagrams are presented.
5.4.2 User transport query

This operation permits to obtain information about user transports provided by IoT service platforms which were discovered in the target MEC platform. The call flow for this operation is depicted in Figure 5.4.2-1.

![Figure 5.4.2-1: Flow of User transport query](image)

User transport query consists of the following steps:

1) The service consumer requests the IoT Service the list of registered user transports offered by discovered IoT service platforms;

2) The IoT Service replies with the updated information of the registered user transports (if at least one exists).

5.4.3 User transport assignment

This operation permits to assign a IoT device to a given user transport provided by a discovered IoT service platform. The call flow for this operation is depicted in Figure 5.4.3-1.

![Figure 5.4.3-1: Flow of User transport assignment](image)

User transport assignment consists of the following steps:

1) The service consumer requests the IoT Service to assign a device to a given user transport offered by a discovered IoT service platform;

2) The IoT Service acknowledges the request.

6 Data Model

6.1 Introduction

The following clauses specify the data types that are used to implement the IoT API, for which the relevant sequence diagrams are described in clauses 5.2.2 to 5.2.6.
6.2 Resource data types

6.2.1 Introduction
This clause defines data structures that shall be used in resource representations.

6.2.2 Type: DeviceInfo
This type represents the information associated to an IoT device.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Data type</th>
<th>Cardinality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gpsi</td>
<td>String</td>
<td>0..1</td>
<td>GPSI of the IoT device (see note)</td>
</tr>
<tr>
<td>imei</td>
<td>String</td>
<td>0..1</td>
<td>IMEI of the IoT device (see note)</td>
</tr>
<tr>
<td>imsi</td>
<td>String</td>
<td>0..1</td>
<td>IMSI of the IoT device (see note)</td>
</tr>
<tr>
<td>iccid</td>
<td>String</td>
<td>0..1</td>
<td>ICCID of the IoT device (see note)</td>
</tr>
<tr>
<td>deviceId</td>
<td>String</td>
<td>1</td>
<td>Human-friendly identifier of the IoT device</td>
</tr>
<tr>
<td>transportConfig</td>
<td>Structure (inlined)</td>
<td>0..1</td>
<td>Transport to be used by the IoT device. Editor's note: FFS, upon definition of transport configuration sequence diagrams.</td>
</tr>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>1</td>
<td>Indication whether the IoT device is active (TRUE) or not (FALSE).</td>
</tr>
</tbody>
</table>

NOTE: At least one attribute among gpsi, imei, imsi, and iccid should be provided. Sufficient security measures shall be put in place when IMEI and/or IMSI is disclosed over the API.

6.3 Subscription data types

6.3.1 Introduction
This clause defines data structures for subscriptions.

6.3.y Type: <TypeNameY>

<Template note: Same structure as in 6.2.x>

Editor's note: resource template to be replaced with actual Content

6.4 Notifications data types

6.4.1 Introduction
This clause defines data structures that define notifications.

6.4.z Type: <TypeNameZ>

<Template note: Same structure as in 6.2.x>

Editor's note: resource template to be replaced with actual Content
6.5 Referenced structured data types

6.5.1 Introduction

This clause defines data structures that are referenced from data structures defined in the previous clauses, but are neither resource representations nor bound to any pub/sub mechanism.

6.5.xx Type: <TypeNameXX>

*Template note: Same structure as in 6.2.x*

Editor's note: resource template to be replaced with actual Content

6.6 Referenced simple data types and enumerations

6.6.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

6.6.yy Type: <TypeNameyy>

*Template note: Same structure as in 6.2.x*

Editor's note: whole subsection to be removed if necessary or replaced with actual Content

7 API definition

7.1 Introduction

This clause defines the resources and operations of the IoT API (IoT API).

7.2 Global definitions and resource structure

All resource URLs of this API shall have the following root:

```
{apiRoot}/{apiName}/{apiVersion}/
```

"apiRoot" and "apiName" are discovered using the service registry. It includes the scheme ("http" or "https"), host and optional port, and an optional prefix string. The "apiName" shall be set to "iot" and "apiVersion" shall be set to "v1" for the current version of the specification. The API shall support HTTP over TLS (also known as HTTPS defined in IETF RFC 2818 [2]). TLS version 1.2 as defined by IETF RFC 5246 [3] shall be supported. HTTP without TLS is not recommended. All resource URIs in the clauses below are defined relative to the above root URI.

The content format of JSON shall be supported.

The JSON format is signalled by the content type "application/json".

This API shall require the use of the OAuth 2.0 client credentials grant type according to IETF RFC 6749 [4] with bearer tokens according to IETF RFC 6750 [5]. See ETSI GS MEC 009 [i.1], clause 7.16 for more information. The token endpoint can be discovered as part of the service availability query procedure defined in ETSI GS MEC 011 [i.2]. How the client credentials are provisioned into the MEC application is out of scope of the present document.

This API supports additional application-related error information to be provided in the HTTP response when an error occurs. See clause 7.15 of ETSI GS MEC 009 [i.1] for more information.

Editor's note. Text above consolidated from previous APIs (012 and 013)
Figure 7.2-1 below illustrates the resource URL structure of this API. Table x.y provides an overview of the resources defined by the present specification, and the applicable HTTP methods.

> Figure 7.2-1: Resource URL structure of the <xyz> API

Table x.y: Resources and methods overview

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Resource Data Types</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Resource Meaning&gt;</td>
<td>&lt;relative URL below root&gt;</td>
<td>&lt;Type 1&gt;</td>
<td>GET &lt;no / short description&gt;, PUT &lt;no / short description&gt;, POST &lt;no / short description&gt;, DELETE &lt;no / short description&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Type 2&gt;</td>
<td></td>
</tr>
</tbody>
</table>

> Figure 7.2-1: Resource URL structure of the <xyz> API

> Table x.y: Resources and methods overview
<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
<th>Resource Data Types</th>
<th>HTTP verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>All foobar sessions</td>
<td>/{userId}/sessions</td>
<td>FoobarSession (for POST)</td>
<td>Retrieve a list of foobar sessions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FoobarSessionList (for GET)</td>
<td>Create a new foobar session</td>
</tr>
<tr>
<td>Individual foobar session</td>
<td>/{userId}/sessions/{sessionId}</td>
<td>FoobarSession</td>
<td>Retrieve a foobar session</td>
</tr>
</tbody>
</table>

**Editor's note:** templates to be replaced with actual content

7.3 Resource: <foo_bar>

7.3.1 Description

This resource is used to …

**Editor's note:** resource description to be replaced with actual content

7.3.2 Resource definition

Resource URI: `{apiRoot}/{apiName}/{apiVersion}/foo_bar`

This resource shall support the resource URI variables defined in table x.y.

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiRoot</td>
<td>See clause 7.2</td>
</tr>
<tr>
<td>apiName</td>
<td>See clause 7.2</td>
</tr>
<tr>
<td>apiVersion</td>
<td>See clause 7.2</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
<td>&lt;definition&gt;</td>
</tr>
</tbody>
</table>

**Editor's note:** table to be replaced with actual content

7.3.3 Resource Methods

7.3.3.1 GET

This method is used to …

**Editor's note:** Start Example

This method is used to retrieve information about a fooBar object.

**Editor's note:** End Example

The valid resource representations are listed in table x.y.
<table>
<thead>
<tr>
<th>Type</th>
<th>In</th>
<th>Out</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>FooBarType</td>
<td></td>
<td>x</td>
<td>only if applicable</td>
</tr>
</tbody>
</table>

In addition to the response codes defined in clause 8.3.3 the codes specified in table x.y are returned by the operation.

**Table x.y: Response codes for the GET method on resource "<Meaning>"**

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
<td>The operation completed successfully.</td>
</tr>
<tr>
<td>&lt;code&gt;</td>
<td>&lt;name from RFC7231&gt;</td>
<td>&lt;description&gt;</td>
</tr>
</tbody>
</table>

Editor's note: method to replaced with actual Content

7.3.3.2 PUT

<same structure as for GET>

Editor's note: method to replaced with actual Content

7.3.3.3 PATCH

<same structure as for GET>

Editor's note: method to replaced with actual Content

7.3.3.4 POST

<same structure as for GET>

Editor's note: method to replaced with actual Content

7.3.3.5 DELETE

<same structure as for GET>

Editor's note: method to replaced with actual Content
History

<table>
<thead>
<tr>
<th>Document history</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0.1  April 2019 Document Skeleton</td>
</tr>
<tr>
<td>2.0.3  December 2020 Integration of contribution MEC(20)000308r1</td>
</tr>
<tr>
<td>2.0.4  April 2021 Integration of contributions MEC(20)000440r1, MEC(20)000444r2</td>
</tr>
<tr>
<td>2.0.5  September 2021 Integration of contributions on transport</td>
</tr>
</tbody>
</table>
| V2.0.5  October 2021 Clean-up done by editHelp!  
E-mail: mailto:edithelp@etsi.org |
| V2.0.6  October 2021 Editorial comments from editHelp were taken into account by the Rapporteur. |