

GATT REST API

White Paper



Date 2014-Apr-07

Revision V10r01

Group Prepared By Internet WG

Feedback Email int-main@bluetooth.org

Abstract:

This white paper describes a RESTful API that can be used to access devices enabled with *Bluetooth*[®] wireless technology using GATT.

Revision History

Revision Number	Date	Comments
D05r01	31 January 2013	Initial Draft
D05r02	5 February 2013	Group Review
D05r03	14 February 2013	More Group Review
D05r04	18 February 2013	More Group Review. Added support for more GATT procedures. Added a mapping table between REST and GATT.
D05r05	5 March 2013	Use Server Side Events (EventSource) to implement indication and notification “tickers”
D05r06	22 March 2013	Better introduction section, more detailed error return codes + some minor editorial updates
D05r07	24 April 2013	Return code 504 added. Other minor edits.
D05r08	30 May 2013	Changes based on group review.
D05r09	4 November 2013	Added that HTTPS controls if security is required or not.
D05r10	22 November 2013	Reviewed by the SIG editor.
D05r11	3 December 2013	Updated after BARB review.
D05r12	23 January 2014	Updated after Smart Home SG review
D05r13	25 February 2014	Converted to new template and used new disclaimer. Approved by BARB.
V10r00	31 March 2014	Prepared for final publication
V10r01	07 April 2014	Further final publication updates from SIG Legal review

V10r01

Contributors

Name	Company
Mats Andersson	connectBlue
Robin Heydon	CSR
Kim Schulz	Samsung
Magnus Olsson	Ericsson
Andreas Larsson	connectBlue

DISCLAIMER AND COPYRIGHT NOTICE

THIS DOCUMENT IS PROVIDED "AS IS" AND BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES WHATSOEVER, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY, TITLE, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, THAT THE CONTENTS OF THIS DOCUMENT IS FREE FROM ERROR OR ANY WARRANTY OTHERWISE ARISING OUT OF THIS DOCUMENT. BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OF INFORMATION IN THIS DOCUMENT. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL BLUETOOTH SIG, ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUE, PROFITS, DATA OR PROGRAMS, OR BUSINESS INTERRUPTION, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO ANY FURNISHING, PRACTICING, MODIFYING, USE OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS OF THIS DOCUMENT, EVEN IF BLUETOOTH SIG, ITS MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

This document is subject to change without notice.

Copyright © 2001 - 2014. Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.

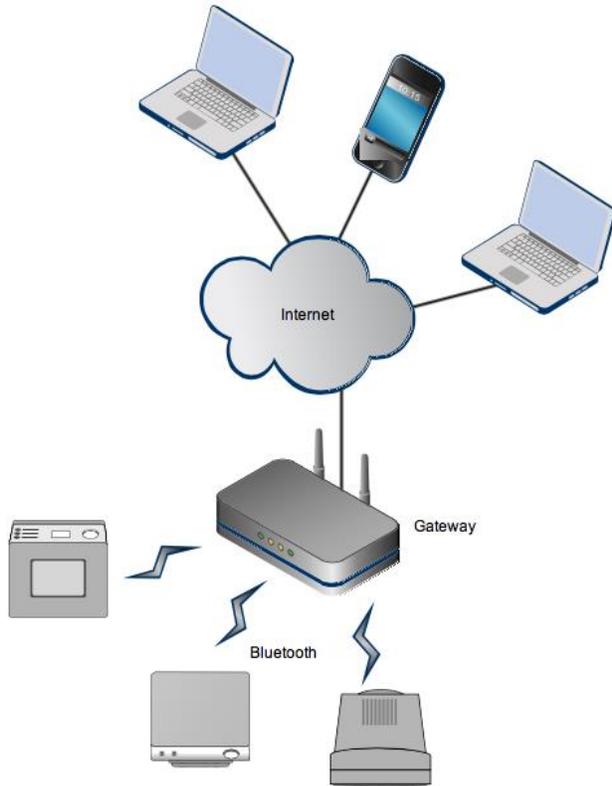
Contents

1	Scope	5
2	GATT REST API.....	6
2.1	Introduction	6
2.2	API Remarks.....	6
2.2.1	Stateless	6
2.2.2	Security.....	6
2.2.3	Indications and Notifications.....	6
2.2.4	The Nodes Resources.....	7
2.3	API Definitions for the Resources.....	7
2.3.1	General	7
2.3.2	Nodes	7
2.3.3	Services	8
2.3.4	Characteristics	10
2.3.5	Characteristic Value	12
2.3.6	Characteristic Descriptors	18
2.3.7	Characteristic Descriptor Value	19
2.3.8	General Error Returns	20
3	Mapping of the REST API and the GATT Procedures	21
3.1	Mapping of GATT Procedures.....	21
4	Abbreviations	23
5	References.....	24

V10r01

1 Scope

This document describes a RESTful API used to access data in devices using the functionality defined in the Bluetooth GATT profile.



V10r01

Figure 1 System

This API assumes a gateway to which the Bluetooth wireless technology enabled device(s) are connected. The gateway contains a "proxy" that serves as the interface between the Internet requests and the Bluetooth GATT profile.

Management of connections, pairing of devices and other parts of the Bluetooth GAP profile is not part of this document. This document assumes this is handled separately and it is GW implementation dependent. For example, handling of connections may be implemented in the following ways:

- The gateway is aware of the type of sensors (e.g., which sensor profile are supported) and the Gateway has all the necessary information to be able to connect to the devices it is supposed to handle. Pairing and other GAP features are handled by the gateway.
- The gateway has a built-in local Web interface or other user interface that is used to administrate which devices it supposed to connect to. The Web interface is also used other GAP functions like pairing of the devices.
- A separate GAP RESTful API is used to administrate the devices to connect to and other GAP functions (see reference [4]).

2 GATT REST API

2.1 Introduction

This REST API defines how GATT (see reference [1]) resources are accessible using the standard HTTP methods (GET=Read and PUT=Write).

The API is using a subset of the REST API design recommendations defined in reference [2].

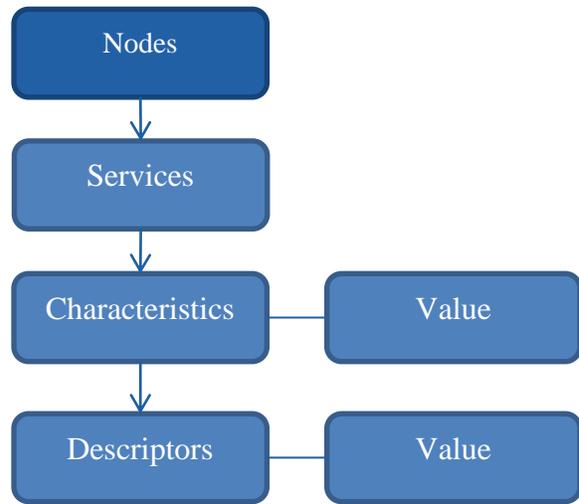


Figure 2 Resources

2.2 API Remarks

2.2.1 Stateless

A RESTful API is stateless, which means that the gateway is not required to memorize any state for its client applications. This also means that several Web clients may access the same resources in parallel.

2.2.2 Security

If Hypertext Transfer Protocol Secure (HTTPS) is used the GATT API shall cause link encryption to be used on the Bluetooth link.

Bluetooth link encryption is not required if (Hypertext Transfer Protocol (HTTP) is used.

2.2.3 Indications and Notifications

If indication or notification is enabled for a Characteristic it is assumed that the latest value indicated or notified are cached in the gateway. This value can be read using the API functions especially defined for reading of cached values.

It is also possible for a Web Service to subscribe to changes using so called Server Side Events (also called Event Source).

2.2.4 The Nodes Resources

How the Nodes resources enumerated by this API have been discovered, paired and connected is not part of this document (more on this in the GAP REST API, see reference [4]).

It is assumed that these Nodes can be reached using the other parts of the GATT REST API for access to their Services and Characteristics.

2.3 API Definitions for the Resources

2.3.1 General

In the API definition are the items named <service>, <characteristic>, and <descriptor>. These handles uniquely identify the corresponding item within a specific node (the node is identified by <node>).

2.3.2 Nodes

Requests:

GET <http://<gateway>/gatt/nodes>

Available nodes

GET <http://<gateway>/gatt/nodes/<node>>

Read data for a specific node identified by the handle <node>,

Responses:

200 – OK - application/json

```
{
  "nodes" : [
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node1>" },
      "handle" : "<node1>",           //Handle for the device
      "bdaddr" : "<bdaddr1>"        //Bluetooth address of the node
    },
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node2>" },
      "handle" : "<node2>",
      "bdaddr" : "<bdaddr2>"
    },
    ...
  ]
}
```

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

2.3.3 Services**Requests:**GET <http://<gateway>/gatt/nodes/<node>/services>

Discover all services in the node identified by <node>.

GET <http://<gateway>/gatt/nodes/<node>/services?primary=1>

Discover all primary services in the node identified by <node>.

GET <http://<gateway>/gatt/nodes/<node>/services?primary=1&uuid=<uuid>>

Discover primary services by UUID in the node identified by <node>.

GET <http://<gateway>/gatt/nodes/<node>/services/<service>>

Read data for a service in the node <node> identified by the handle <service> (discovered by one of the methods/URIs defined earlier in this section).

Responses:

200 – OK - application/json

```
{
  "services" : [
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/services/<service1>" },
      "handle" : "<service1>",
      "uuid"   : "<uuid1>",           //Services UUID
      "primary" : <true/false>       //True if the service is a primary service
    },
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node >/services/<service2>" },
      "handle" : "<service2>",
      "uuid"   : "<uuid2>",
      "primary" : <true/false>
    },
    ...
  ]
}
```

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

2.3.4 Characteristics

Requests (discovery):

GET <http://<gateway>/gatt/nodes/<node>/services/<service>/characteristics>

Discover all characteristics of the service <service> in the node identified by <node>.

GET <http://<gateway>/gatt/nodes/<node>/characteristics?uuid=<uuid>>

Discover characteristics by UUID in the node identified by <node>.

GET <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>>

Get data for the characteristic identified by <characteristics> in the node identified <node> (discovered by one of the methods/URIs defined earlier in this section).

Responses:

200 – OK - application/json

```
{
  "characteristics" : [
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic1>" }
      "handle"   : "<characteristic1>",
      "uuid"     : "<uuid1>",           //Characteristics UUID
      "properties" : <prop1>           //Bit mask describing the properties of the characteristic
                                     //Bit 0: Broadcast
                                     //Bit 1: Readable
                                     //Bit 2: Writable with no response
                                     //Bit 3: Writable
                                     //Bit 4: Notify
                                     //Bit 5: Indicate
                                     //Bit 6: Authenticated signed write
                                     //Bit 7: Extended property available
    },
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic2>" },
      "handle"   : "<characteristic2>",
      "uuid"     : "<uuid2>",
      "properties" : <prop2>
    },
    ...
  ]
}
```

V10r01

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

Requests (enabling and disabling of Indications and Notifications)

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>?indicate=1>

Enable indication for the characteristic <characteristic> in the node identified by <node>.

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>?indicate=0>

Disable indication for the characteristic <characteristic> in the node identified by <node>.

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>?notify=1>

Enable notification for the characteristic <characteristic> in the node identified by <node>.

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>?notify=0>

Disable notification for the characteristic <characteristic> in the node identified by <node>.

Responses:

200 – OK

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

403 – Forbidden

Access denied due to protected resource:
- The Characteristic cannot be indicated or notified.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

V10r01

2.3.5 Characteristic Value

Requests (reading of characteristic values):

GET <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value>

Read the value of the characteristic <characteristic> in the node identified by <node>.

GET

<http://<gateway>/gatt/nodes/<node>/characteristics/value?uuid=<uuid>&start=<handle>&end=<handle>>

Read the value of a characteristic by UUID. First found is returned.

GET <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?long=1>

Read the long value of the characteristic <characteristic> in the node identified by <node>.

Responses:

200 – OK - application/json

```
{
  "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>" },
  "handle" : "<characteristic>",
  "value" : "<value>"           //Value as HEX string
}
```

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

403 – Forbidden

Access denied due to protected resource:

- The characteristic value is not readable.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

412 – Precondition failed

The additional operation on the resource not supported:

- Reading of Characteristic by UUID not supported.
- Reading of long value not supported.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

Requests (reading of multiple characteristic values):GET <http://<gateway>/gatt/nodes/<node>/characteristics/value?multiple=1>

Read the value of multiple characteristics identified by the handles in the body in the node identified by <node>.

Body:

```
{
  "characteristics" : [
    { "handle" : "<characteristic1>" },
    { "handle" : "<characteristic2>" },
    { ... }
  ]
}
```

Responses:

200 – OK - application/json

```
{
  "characteristics" : [
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic1>" }
      "handle" : "<characteristic1>",
      "value" : "<value1>"
    },
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic2>" },
      "handle" : "<characteristic2>",
      "value" : "<value2>"
    },
    ...
  ]
}
```

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

403 – Forbidden

Access denied due to protected resource:
- The characteristic value is not readable.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

412 – Precondition failed

The additional operation on the resource not supported:
- Reading of multiple value not supported.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

Requests (reading of cached indicated or notified value):

GET <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1>

Read the latest cached indicated value of the characteristic <characteristic> in the node identified by <node>. Only valid if the characteristic has indication enabled.

GET <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?notify=1>

Read the latest cached notified value of the characteristic <characteristic> in the node identified by <node>. Only valid if the characteristic has notification enabled.

Responses:

200 – OK - application/json

```
{
  "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>" },
  "value" : "<value>" //Value as HEX string
}
```

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

412 – Precondition failed

The additional operation on the resource not supported:
- Indication or notification not supported or not activated.

Use Server Side Events to subscribe to indications or notifications

URI for the Server Side Events:

<http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1&event=1>

This URI is used to trigger a subscription of indicated values for the characteristic <characteristic> in the node identified by <node>. Only valid if the characteristic has indication enabled.

<http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?notify=1&event=1>

This URI is used to trigger a subscription of notified values for the characteristic <characteristic> in the node identified by <node>. Only valid if the characteristic has notification enabled.

More on Server Side Events in reference [3].

Javascript Example:

```
var source = new EventSource("http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1&event=1");
source.onmessage = function (event) {
  var data = event.data;
  updateindication(data); // The action performed when an indication event is received
};
```

Data Stream:

```
data: <event 1 data>
data: <event 2 data>
...
data: <event n data> }
```

V10r01

Requests (writing of characteristic values):

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>>

Write of the value <value> to the characteristic <characteristic> in the node identified by <node>.

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>?long=1>

Write of the long value <value> to the characteristic <characteristic> in the node identified by <node>.

PUT

<http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>>?noresponse=1

A write without response of the value <value> to the characteristic <characteristic> in the node identified by <node>.

Responses:

200 – OK

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.
--

403 – Forbidden

Access denied due to protected resource: - The characteristic value is not writable.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
--

405 – Method not allowed

Returned if POST is used and Write without Response not is supported.

412 – Precondition failed

The additional operation on the resource not supported: - Writing of long value not supported
--

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.
--

Requests (reliable writes):

PUT <http://<gateway>/gatt/nodes/<node>/characteristics/value?reliable=1>

Perform a reliable write of the characteristics defined in the body and in the node identified by <node>.

Body:

```
{
  "characteristics" : [
    {
      "handle" : "<characteristic1>",
      "value" : "<value1>" //Value as HEX string
    },
    {
      "handle" : "<characteristic2>",
      "value" : "<value2>"
    },
    ...
  ]
}
```

Responses:

200 – OK

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

403 – Forbidden

Access denied due to protected resource:

- One or more characteristic values is not writable.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

412 – Precondition failed

The additional operation on the resource not supported:

- Reliable write not supported.

415 – Unsupported Media Type

The client supplied a media type other than application/json or badly formatted data.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

2.3.6 Characteristic Descriptors

Requests (discovery):

GET <http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/descriptors>

Discover all descriptors of the characteristic <characteristic> in the node identified by <node>.

GET <http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor>>

Read data for the descriptor identified by the handle <descriptor> (discovered by the methods/URIs defined earlier in this section).

Responses:

200 – OK - application/json

```
{
  "descriptors" : [
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor1>" }
      "handle"   : "<descriptor1>",
      "uuid"     : "<uuid1>"           //Descriptor Type UUID
    },
    {
      "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor2>" }
      "handle"   : "<descriptor2>",
      "uuid"     : "<uuid2>"
    },
    ...
  ]
}
```

V10r01

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

2.3.7 Characteristic Descriptor Value

Requests (reading of descriptor values):

GET <http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor>/value>

Read the value of the descriptor <descriptor> in the node identified by <node>.

Responses:**200 – OK - application/json**

```
{
  "self" : { "href" = "http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor>" },
  "value" : "<value>"           //Value as HEX string
}
```

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

403 – Forbidden

Access denied due to protected resource:

- The descriptor value is not readable.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

V10r01

Requests (writing of descriptor values):

PUT <http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor>/value/<value>>

Write of the value <value> to the descriptor <descriptor> in the node identified by <node>.

PUT <http://<gateway>/gatt/nodes/<node>/descriptors/<descriptor>/value/<value>?long=1>

Write the long value <value> to the descriptor <descriptor> in the node identified by <node>.

Responses:

200 – OK

or

400 - Bad Request

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

403 – Forbidden

Access denied due to protected resource:

- The Characteristic value is not writable.

404 - Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

504 – Not able to connect

The server has not been able to connect/reconnect to the requested node.

2.3.8 General Error Returns

These are general error return codes that may be returned for any of the API methods specified above and that not are listed for the individual method.

406 – Not acceptable

The client has requested another media type than the supported application/json.

3 Mapping of the REST API and the GATT Procedures

3.1 Mapping of GATT Procedures

GATT Feature	GATT Sub-Procedure	REST API Method and URI
Server Configuration	Exchange MTU	Not applicable
Primary Service Discovery	Discover All Primary Services	GET <a href="http://<gateway>/gatt/nodes/<node>/services?primary=1">http://<gateway>/gatt/nodes/<node>/services?primary=1
	Discover Primary Services By Service UUID	GET <a href="http://<gateway>/gatt/nodes/<node>/services?primary=1&uuid=<uuid>">http://<gateway>/gatt/nodes/<node>/services?primary=1&uuid=<uuid>
Relationship Discovery	Find Included Services	GET <a href="http://<gateway>/gatt/nodes/<node>/services">http://<gateway>/gatt/nodes/<node>/services
Characteristic Discovery	Discover All Characteristics of a Service	GET <a href="http://<gateway>/gatt/nodes/<node>/services/<service>/characteristics">http://<gateway>/gatt/nodes/<node>/services/<service>/characteristics
	Discover Characteristic by UUID	GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics?uuid=<uuid>">http://<gateway>/gatt/nodes/<node>/characteristics?uuid=<uuid>
Characteristic Descriptor Discovery	Discover All Characteristic Descriptors	GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/descriptors">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/descriptors
Characteristic Value Read	Read Characteristic Value	GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value
	Read Using Characteristic UUID	GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?uuid=<uuid>&start=<handle>&end=<handle>">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?uuid=<uuid>&start=<handle>&end=<handle>
	Read Long Characteristics Value	GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?long=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?long=1
	Read Multiple Characteristics Value	GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/value?multiple=1">http://<gateway>/gatt/nodes/<node>/characteristics/value?multiple=1
Characteristic Value Write	Write Without Response	PUT <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>?noresponse=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>?noresponse=1
	Signed Write Without Response	Not defined
	Write Characteristic Value	PUT <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>

V10r01

GATT Feature	GATT Sub-Procedure	REST API Method and URI
	Write Long Characteristic Value	PUT <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>?long=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value/<value>?long=1
	Characteristic Value Reliable Write	PUT <a href="http://<gateway>/gatt/nodes/<node>/characteristics/value?reliable=1">http://<gateway>/gatt/nodes/<node>/characteristics/value?reliable=1
Characteristic Value Notification	Notification	PUT <a href="http://<gateway>/gatt/nodes/<node>/characteristics?notify=1">http://<gateway>/gatt/nodes/<node>/characteristics?notify=1 GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?notify=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?notify=1 EventSource <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?notify=1&event=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?notify=1&event=1
Characteristic Value Indication	Indication	PUT <a href="http://<gateway>/gatt/nodes/<node>/characteristics?indicate=1">http://<gateway>/gatt/nodes/<node>/characteristics?indicate=1 GET <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1 EventSource <a href="http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1&event=1">http://<gateway>/gatt/nodes/<node>/characteristics/<characteristic>/value?indicate=1&event=1
Characteristic Descriptor Value Read	Read Characteristic Descriptor	GET <a href="http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value">http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value
	Read Long Characteristic Descriptors	GET <a href="http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value?long=1">http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value?long=1
Characteristic Descriptor Value Write	Write Characteristic Descriptor	PUT <a href="http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value/<value>">http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value/<value>
	Write Long Characteristic Descriptors	GET <a href="http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value/<value>?long=1">http://<gateway>/gatt/nodes/<node>/descriptor/<descriptor>/value/<value>?long=1

V10r01

4 Abbreviations

Abbreviation or Acronym	Meaning
REST	Representational State Transfer
GATT	Generic Attribute Profile
GAP	Generic Access Profile

5 References

- [1] Bluetooth Core Specification v4.0 or later
- [2] REST API Design Rulebook; Mark Massé; O'Reilly Media; 2012; ISBN: 978-1-449-31050-9
- [3] <http://dev.w3.org/html5/eventsource/>; Definition of Server Side Event (Event Source)
- [4] GAP REST API; Bluetooth SIG White Paper