# GAP REST API

White Paper

**Date**  2014-Apr-07

**Revision**  V10r01

**Group Prepared By**  Internet Working Group

**Feedback Email**  int-main@bluetooth.org

**Abstract**:
This white paper describes a GAP RESTful API that can be used to administer connections between a gateway and connected *Bluetooth*® low energy wireless technology enabled devices. It is used in combination with the GATT REST API.

*Revision History*

| Revision Number | Date | Comments |
|---|---|---|
| D05r01 | 24 April 2013 | Initial Draft |
| D05r02 | 30 May 2013 | Minor changes based on review comments. |
| D05r03 | 2 October 2013 | After WG F2F review |
| D05r04 | 3 November 2013 | Added that HTTPS controls if security is required or not. |
| D05r05 | 22 November 2013 | Reviewed by the SIG editor. |
| D05r06 | 5 December 2013 | After BARB review. |
| D05r07 | 23 January 2014 | Updated after Smart Home SG review |
| D05r08 | 05 March 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 |

*Contributors*

| Name | Company |
|---|---|
| Mats Andersson | connectBlue |
| Andreas Larsson | connectBlue |
| Robin Heydon | CSR |
| Magnus Olsson | Ericsson |
| Tim Howes | CSR |

**V10r01**

# Contents

V10r01

# 1 Scope

This document describes a RESTful API used to administer connections between devices featuring Bluetooth low energy technology and a Gateway.
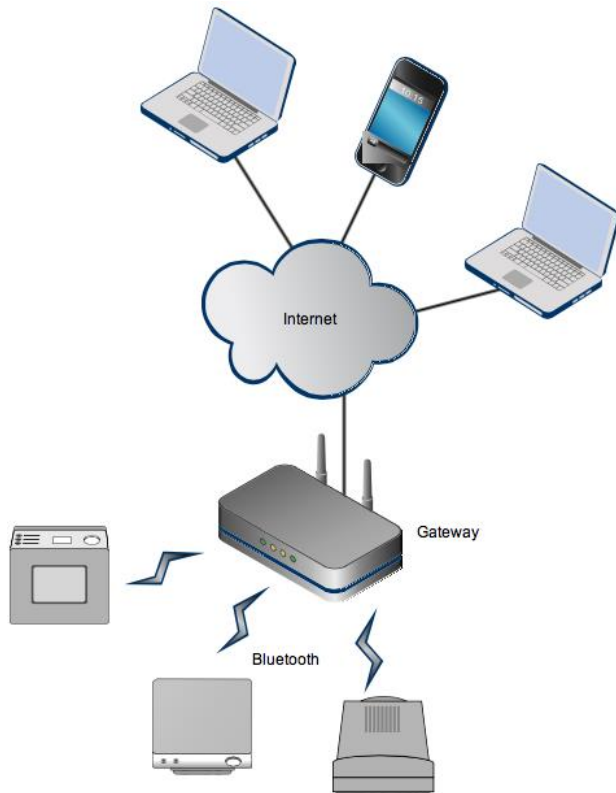
V10r01



***Figure 1****: System*

This API can be used to administrate connections to devices equipped with Bluetooth low energy technology later to be accessed by the GATT REST API (see reference [3]).

# 2 GAP REST API

## 2.1  Introduction

This REST API defines how GAP (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].

<div style="text-align:center">

Nodes

</div>

***Figure 2***  *Resources*

## 2.2  API Remarks

### 2.2.1  Stateless

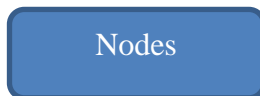A RESTful API is stateless which means that the gateway is not required to memorize any state for its client applications.

This also means that several Web clients may access the same resources in parallel.

### 2.2.2  GAP Role

The REST API shall only handle a gateway operating in the GAP Central and Observer roles (see reference [1]).

### 2.2.3  Security

Detail on how to setup security and bonding of devices are out-of-scope for this specification as a local presence and a local user interface or similar is required.

If Hypertext Transfer Protocol Secure (HTTPS) is used, the GAP API shall cause Bluetooth encryption for products equipped with Bluetooth wireless technology to be used on the Bluetooth link.

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

### 2.2.4  Enabled Nodes

The nodes that are to be connected and reconnected automatically are called *enabled* nodes.

It is assumed that the gateway is responsible for the connections between all enabled nodes and the gateway and will continuously try to establish a connection for all enabled nodes when possible. This includes when a connection is lost between a node and the gateway and when the gateway is powered up.

V10r01

## 2.3 API Definitions for the Resources

### 2.3.1 General

Individual nodes are identified by a handle called <node>.

### 2.3.2 Nodes

**Requests (discover nodes and list enabled nodes):**

GET http://<gateway>/gap/nodes?passive=1

The gateway will perform passive scan for nodes. Used scan parameters are decided by the gateway.

GET http://<gateway>/gap/nodes?active=1

The gateway will perform active scan for nodes. Used scan parameters are decided by the gateway.

GET http://<gateway>/gap/nodes?enable=1

The gateway will return a list of enabled devices (devices that are either connected or will be connected when available and at gateway power-up).

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

The gateway will return data for an enabled node identified by the handle <node>.

**Responses:**

```
200 – OK - application/json

{

"nodes" : [

{

"self" : { "href" = "http://<gateway>/gap/nodes/<node1>" },

"handle"   : "<node1>",             //Handle for the device

"bdaddr"  : "<bdaddr1>",          //Bluetooth address of the node

"AD"       : [

        {

        "ADType " : <type1>,           // See reference [1]

        "ADValue" : "<value1>"          // Value as HEX string

        },

        {
```

V10r01

V10r01

```
            "ADType" : <type2>,

            "ADValue" : "<value2>"

             },
             {

            …

             }

             ]

},

{

"self"  : { "href" = "http://<gateway>/gap/nodes/<node2>" },

"handle"   : "<node2>",

"bdaddr"  : "<bdaddr2>"

"AD"        : [ … ]

},

{

…

}

]

}
```

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. |

**Requests (enabling and disabling of nodes)**

PUT http://<gateway>/gap/nodes/<node>?connect=1(&interval=<interval>&latency=<latency>&enable=1)

Enable and connect to the node identified by <node>. The gateway will try to use the connection interval <interval> and the connection latency <latency>. If left out, default values will be used. The gateway will try to reconnect to the device if the connection is lost or at gateway power-up.

PUT http://<gateway>/gap/nodes/<node>?enable=0

Remove the node <node> from the list of enabled nodes.

**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. |
| 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. |

**Requests (name discovery)**

GET http://<gateway>/gap/nodes/<node>?name=1

Perform a name discovery of the node identified by handle <node>.

**Responses:**

| 200 – OK - application/json |
|---|
| *{* |
| *"self"       : { "href" = "http://<gateway>/gap/nodes/<node>" },* |
| *"name"     : "<name>"* |
| *}* |

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.

### 2.3.3  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 Abbreviations

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

V10r01

# 4 References

[1] Bluetooth Core Specification v4.0 or later

[2] REST API Design Rulebook; Mark Massé; OŔeilly Media; 2012; ISBN: 978-1-449-31050-9

[3] GATT REST API White Paper

V10r01