Pulse Oximeter APIs
Approved Version 1.0 – 24 Jul 2018

Open Mobile Alliance
OMA-TS-Pulse_Oximeter_APIs-V1_0-20180724-A
Contents
1. SCOPE .................................................................................................................................................. 4
2. REFERENCES ......................................................................................................................................... 5
   2.1 NORMATIVE REFERENCES ............................................................................................................. 5
   2.2 INFORMATIVE REFERENCES ........................................................................................................... 5
3. TERMINOLOGY AND CONVENTIONS .................................................................................................. 6
   3.1 CONVENTIONS .................................................................................................................................. 6
   3.2 ABBREVIATIONS ............................................................................................................................... 7
4. INTRODUCTION ..................................................................................................................................... 8
   4.1 VERSION 1.0 ....................................................................................................................................... 8
5. TECHNICAL SPECIFICATIONS ............................................................................................................. 9
   5.1 THE SERVICE DISCOVERY ON THE GOTAPI-4 INTERFACE .......................................................... 9
   5.2 ONE-SHOT MEASURING API ........................................................................................................... 11
      5.2.1 Request for one-shot measuring on the GotAPI-1 Interface ......................................................... 12
      5.2.2 Request for one-shot measuring on the GotAPI-4 Interface ......................................................... 12
      5.2.3 Response for one-shot measuring on the GotAPI-4 Interface ..................................................... 14
      5.2.4 Response for one-shot measuring on the GotAPI-1 Interface ..................................................... 19
   5.3 ASYNCHRONOUS MESSAGING API ................................................................................................ 22
      5.3.1 Request for asynchronous messaging on the GotAPI-1 Interface ................................................ 24
      5.3.2 Request for asynchronous messaging on the GotAPI-4 Interface .............................................. 24
      5.3.3 Response for asynchronous messaging on the GotAPI-4 Interface ............................................. 26
      5.3.4 Response for asynchronous messaging on the GotAPI-1 Interface ............................................. 29
      5.3.5 Asynchronous message from the Plug-In to the GotAPI Server on the GotAPI-4 Interface ............ 30
      5.3.6 Asynchronous message from the GotAPI Server to the application on the GotAPI-5 Interface .......... 34
      5.3.7 Stop request from the application to the GotAPI Server on the GotAPI-1 Interface ...................... 36
      5.3.8 Stop request from the GotAPI Server to the Plug-In on the GotAPI-4 Interface ............................ 37
      5.3.9 Stop response from the Plug-In to the GotAPI Server on the GotAPI-4 Interface .......................... 38
      5.3.10 Stop response from the GotAPI Server to the application on the GotAPI-1 Interface .................... 39
   APPENDIX A. CHANGE HISTORY (INFORMATIVE) ................................................................................ 40
      A.1 APPROVED VERSION HISTORY .................................................................................................... 40

Figures
Figure 1: Message flow of the Service Discovery ..................................................................................... 9
Figure 2: Message flow of the One-shot measuring API .......................................................................... 11
Figure 3: Message Flow of the Asynchronous messaging API .................................................................. 23

Tables
No table of figures entries found.
1. Scope

Pulse Oximeters provide two vital signs measurements, oxygen saturation and pulse rate.

The GotAPI provides a multi-purpose web-based framework to enable interwork applications and external devices such as pulse oximeters. The GotAPI consists of the GotAPI Server and the Extension Plug-Ins. A smartphone application communicates with a specific Extension Plug-In through the GotAPI Server using Web technologies.

In the GotAPI framework, an Extension Plug-In interacts with the Pulse Oximeters and exposes interfaces to the GotAPI Server. Thanks to Extension Plug-Ins, smartphone applications can interact with many kinds of Pulse Oximeters using consistent APIs specified in this specification.

This is the technical specification part of the Pulse Oximeter Device Web APIs whose requirements and architecture are defined in a separate document [DWAPI-PCH].
2. References

2.1 Normative References

[DWAPI-PCH] Device WebAPI-PCH
OMA-ER-Device_WebAPIs-V1.0-20160419-C URL:http://www.openmobilealliance.org/


2.2 Informative References


3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

Agent A node that collects and transmits personal health data to an associated manager.

API Patterns Design guidelines and requirements for definition of APIs.

Browser Context Web applications executing under a Web browser as Web runtime environment.

Datagram An API providing access to UDP protocol based networking.

Device A physical device implementing either an Agent or manager role.

ECMAScript Use definition from [OMADICT].

Hybrid Native/Web App An application designed to execute under the native OS / middleware environment of a device, and that use native APIs for the execution of web content in addition to native code.

JavaScript Use definition from [OMADICT].

Manager A node receiving data from one or more agent systems. Examples of managers include a cellular phone, health appliance, set top box, or computer system.

Native App An application designed to execute under the native OS / middleware environment of a device.

Personal Health Device A device used in personal health applications.

Pulse Oximeter Plug-In An Extension Plug-In that is defined by the GotAPI specification. It interacts with pulse oximeters and provides Web APIs, exposing the features of pulse oximeters, for applications through the GotAPI Server.

Socket An API providing access to TCP protocol based networking.

Uniform Resource Identifier Use definition from [OMADICT].

User Agent Use definition from [OMADICT].

Web The World Wide Web, a content and application framework based upon hypertext and related technologies, e.g. XML, JavaScript/ECMAScript, CSS, etc.

Web Application An application designed using Web technologies (e.g. HTML, CSS, and Javascript).

Web IDL An IDL language for Web application APIs.

Web Runtime Application A client-side Web application that is executed in Web runtime environments.

Web Runtime Environment Client software that supports the execution of Web applications (e.g. browsers or widget engines).

WebSocket An API providing networking services per the WebSocket standard [WebSocket].

Widget Context Web applications installed and executing under a W3C Widget [W3C-Widgets] engine as Web runtime environment.

Widget Engine Software which supports the execution of Web applications running outside a browser context, e.g. with the same functional capabilities as browsers but without the user interface functions provided by a browser, including window frames, menus, toolbars and scroll bars.
3.2 Abbreviations

API Application Programming Interface
EventSource The EventSource API (Server-Sent Events)
HTTP HyperText Transfer Protocol
IDL Interface Definition Language
JSON JavaScript Object Notation
MIME Multipurpose Internet Mail Extensions
OMA Open Mobile Alliance
REST REpresentational State Transfer
RPC Remote Procedure Call
SCR Static Conformance Requirements
TS Technical Specification
UA User Agent
UE User Equipment
URI Uniform Resource Identifier
URL Uniform Resource Locator
W3C World Wide Web Consortium
WRAPI The OMA Web Runtime API enabler
XML eXtensible Markup Language
XSD XML Schema Definition
4. Introduction

This is the technical specification part of the Pulse Oximeter Device Web APIs whose requirements and architecture are defined in a separate document [DWAPI-PCH].

Pulse oximeters report measurements of body oxygen saturation and pulse rate. They are connected to smartphones via physical media such as Bluetooth to Extension Plug-Ins that expose the pulse oximeter features to applications through the GotAPI 1.1 framework [GotAPI1.1]. Applications are able to use the pulse oximeter features through the GotAPI 1.1 framework as defined by GotAPI 1.1.

Pulse oximeters are typically accessed by streamed data (Asynchronous messaging). Pulse oximeters report measurements in several different manners. Data is often streamed at regular intervals (e.g., once per second) or reported over various ‘sample’ times. The latter cases are referred to as “spot modalities”, and they can be fast, slow, or just ‘spot’. The spot measurements represent a more ‘robust’ estimate of the actual value. The IEEE standards support several types of spot modality measurements. Spot measurements, being episodic, typically have time stamps whereas streaming measurements tend not to have time stamps.

The pulse oximeter Plug-Ins report at least one type of oxygen saturation and pulse rate. It may be further described as modality spot, modality fast spot, or modality slow spot.

Pulse oximeter devices supported in this specification are expected to be able to report the oxygen saturation and pulse rate. The description of the measurements reported by the Plug-In follows the definition of the IEEE 11073-10404 Pulse Oximeter specialization specification. Nonetheless, this does not mean the pulse oximeters themselves must be compliant to the IEEE 11073-10404 specification. The pulse oximeter APIs specified in this document can be used for pulse oximeters that support IEEE 11073-10404 as well as those that do not support IEEE 11073-10404. In the latter case, however, the pulse oximeters must provide the Plug-Ins with the necessary information such that the Plug-Ins can fulfil their reporting requirements as specified in this document.

This document defines API specifications for

- Service Discovery
- One-shot messaging API
- Asynchronous messaging API

The architectural aspects of these APIs are defined in the AD section of [DWAPI-PCH]. This specification must adhere to the GotAPI 1.1 specification.

Due to the nature of the data arising from pulse oximeters, not all the use cases may be supported by one-shot messaging. But there are use cases that can be supported by one-shot messaging. One-shot messaging allows developers simpler and convenient access, while asynchronous messaging enables versatile use cases.

4.1 Version 1.0

Pulse Oximeter APIs version 1.0 includes the functionality:

- Device Web API specifications for DWAPI-PCH, with device classes from IEEE 11073-10404 Pulse Oximeter specialization based on the GotAPI 1.1 framework
- Device Web APIs for Service Discovery, and asynchronous measuring
- Requirements and architecture documents [DWAPI-PCH]
5. Technical Specifications

This specification must adhere to the GotAPI 1.1 specification. This document specifies certain aspect of GotAPI 1.1 as the basis and introduces new elements, which are necessary for pulse oximeters supporting IEEE 11073-10404 Pulse Oximeter specialization.

In order to increase readability, the specification described below uses the same tables as defined in GotAPI 1.1, describing the necessary features including those of the general procedures of any GotAPI 1.1 uses as well as those specific to the Pulse Oximeter APIs. Those specifications that are specific to the Pulse Oximeter APIs are colored in green in the following tables, in order to increase readability, to make identity distinction easily. Those rows that are not colored in green are merely copies from the GotAPI 1.1 specification [GotAPI 1.1]

5.1 The Service Discovery on the GotAPI-4 Interface

Service Discovery API enables applications to discover available services as define in the Section 7.2.1[DWAPI-PCH]. Service Discovery API specification adheres to that of GotAPI 1.1.

Here is the Service Discovery based on what is defined in GotAPI 1.1. After the application obtains authorization for access to GotAPI-based APIs using the GotAPI-2 Interface, the application sends the Service Discovery request to the GotAPI Server. Then the GotAPI Server sends the Service Discovery request to all of the installed Extension Plug-Ins. The message flow of the Service Discovery is shown in Fig. 1.

![Figure 1: Message flow of the Service Discovery](image)

The specific data in the message flows labelled (4) in the figure above are defined as follows. The other message flows SHALL be consistent to what are defined in the GotAPI 1.1 specification:

When the GotAPI Server receives the request of the Service Discovery API from an application, the GotAPI Server sends the Plug-In discovery request to the installed Plug-Ins as defined in the GotAPI specification. When the Pulse Oximeter Plug-In
receives the Plug-In discovery request from the GotAPI Server, the Pulse Oximeter Plug-In SHALL return the message as follows:

**Definition of the data object for the Plug-In discovery response**

<table>
<thead>
<tr>
<th>Name</th>
<th>Sub name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestCode</td>
<td></td>
<td>int</td>
<td>The request code coming from the GotAPI Server.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td></td>
<td>int</td>
<td>If success, the value is 0, otherwise an integer other than 0, which indicates an error code. This specification doesn't define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>services</td>
<td></td>
<td>Array</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>serviceId</td>
<td></td>
<td>String</td>
<td>The service identifier. The id could be &quot;com.example.plugin&quot;.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>name</td>
<td></td>
<td>String</td>
<td>The name of the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>manufacturer</td>
<td></td>
<td>String</td>
<td>The manufacturer of the targeted pulse oximeter.</td>
<td>Optional</td>
</tr>
<tr>
<td>version</td>
<td></td>
<td>String</td>
<td>The version of the targeted pulse oximeter.</td>
<td>Optional</td>
</tr>
<tr>
<td>type</td>
<td></td>
<td>String</td>
<td>This value represents the type of the network used to connect to the pulse oximeter. The value must be any one of &quot;WiFi&quot;, &quot;BLE&quot;, &quot;NFC&quot;, &quot;Bluetooth&quot; or &quot;USB&quot;.</td>
<td>Optional</td>
</tr>
<tr>
<td>online</td>
<td></td>
<td>Boolean</td>
<td>If the service is available, this value SHALL be true. Otherwise (e.g. the Pulse Oximeter Plug-In has not yet detect any pulse oximeters or the Plug-In is not allowed to access to any devices), this value SHALL be false.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>scopes</td>
<td></td>
<td>Array</td>
<td>This value SHALL be an array including a string &quot;oximeter&quot; as an array element (&quot;oximeter&quot; ...).</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

The Pulse Oximeter Plug-In MAY append additional data in the data object as needed.

This data object is sent to the Plug-Ins in an OS specific mechanism, e.g., Intents for Android.

**Requirements for OS-specific response channel and data container**

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the response. The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

**Example of the data object of the Android Explicit Intents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>&quot;org.deviceconnect.action.RESPONSE&quot;</td>
<td>This value is defined by the GotAPI Server application.</td>
</tr>
<tr>
<td>Component</td>
<td>&quot;org.deviceconnect&quot;</td>
<td>This value is the package name of the GotAPI Server application.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Extra</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>requestCode</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
5.2 One-shot measuring API

One-shot API enables applications to receive measured data from pulse oximeters by one HTTP request/response transaction as defined in the Section 7.2.2 [DWAPI-PCH]. One-shot measuring API specification adheres to that of GotAPI 1.1.

As defined by GotAPI 1.1, after the application obtains authorization to access GotAPI-based APIs using the GotAPI-2 Interface and completes the Service Discovery, the application can use the service (so called "One-shot measuring API") provided by the Plug-In through the GotAPI Server.

The One-shot measuring API offers a measurement result reported by the targeted device in response to a request. The message flow of this API is as shown below.

![Message flow of the One-shot measuring API](image)

This section defines the data object for all the message flows described in the figure above.

This value is an example. Note that this is "not" a JSON string. This value must be an Array object whose content is the same as the following JSON example:

```json
[
  {
    "id": "org.example.plugin.12345",
    "name": "Coolest Pulse Oximeter",
    "manufacturer": "ABC Health Care Inc.",
    "version": "3.0",
    "type": "Bluetooth",
    "online": true,
    "scopes": ["oximeter"
  },
  ...
]
```

This name-value pair is an additional data which is not defined by this specification.

This value must be an Array object whose content is the same as the following JSON example:
5.2.1 Request for one-shot measuring on the GotAPI-1 Interface

When the application uses the one-shot measuring it sends a request to the GotAPI Server on the GotAPI-1 Interface as follows:

**Definition of the HTTP request**

<table>
<thead>
<tr>
<th>Method</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP PUT</td>
<td></td>
</tr>
</tbody>
</table>

**Request URL**

- http://127.0.0.1:4035/gotapi/health/oximeter
- https://127.0.0.1:4036/gotapi/health/oximeter

**Definition of the request parameters**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceId</td>
<td>The identifier of the targeted service. This value is available from the Service Discovery API on the GotAPI-1 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>accessToken</td>
<td>The access token obtained from the GotAPI Auth Server through the GotAPI-2 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>nonce</td>
<td>A nonce generated by the application, which is described in the section “7.3.3.3 HMAC server authentication using trusted Application ID for the Server spoofing attack” in the GotAPI specification.</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Example of the request URL**

http://127.0.0.1:4035/gotapi/health/oximeter?serviceId=abcdefg123&accessToken=0987654321&nonce=93b3a219347

5.2.2 Request for one-shot measuring on the GotAPI-4 Interface

When an application sends a request to the GotAPI Server on the GotAPI-1 Interface, the GotAPI Server passes the request to the Plug-In on the GotAPI-4 Interface. The request includes the data object as follows:

**Definition of the data object for request**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>This value SHALL be &quot;GET&quot;.</td>
<td>Mandatory if the OS is not Android. Otherwise, optional. If the OS is Android, the &quot;Action&quot; value SHALL include this information as described below.</td>
</tr>
<tr>
<td>receiver</td>
<td>String</td>
<td>The address of the GotAPI Server application used by Plug-Ins. Generally, it is the application ID recognized by the OS, such as a package name.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>requestCode</td>
<td>int</td>
<td>A request code identifying the request. This value could be any number but must MUST be an integer greater than 0, and unique for each open request, to ensure responses can be correlated.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>serviceId</td>
<td>String</td>
<td>The identifier of the targeted Service. This value is provided by the application over the GotAPI-1 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>api</td>
<td>String</td>
<td>The value must be &quot;gotapi&quot;.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>profile</td>
<td>String</td>
<td>The value must be &quot;health&quot;.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>
attribute | String | The value must be "oximeter" | Mandatory
---|---|---|---
clientId | String | The identifier of the application, which is generated by the Plug-In when the Plug-In Approval procedure defined in the GotAPI specification. | Mandatory
accessToken | String | The access token for the application, which is generated by the Plug-In when the Plug-In Approval procedure defined in the GotAPI specification. | Mandatory

This data object is sent to the Plug-Ins in an OS specific mechanism, e.g., Intents for Android.

Requirements for OS-specific request channel and data container

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request. The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

Example of the data object of the Android Explicit Intents

<table>
<thead>
<tr>
<th>Name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>org.deviceconnect.action.GET</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;GET&quot;.</td>
</tr>
<tr>
<td>Component</td>
<td>org.example.plugin</td>
<td>This value is the package name of the Plug-In application.</td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
</tr>
<tr>
<td>receiver</td>
<td>org.deviceconnect</td>
<td></td>
</tr>
<tr>
<td>requestCode</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>serviceId</td>
<td>dev1.example.org</td>
<td></td>
</tr>
<tr>
<td>api</td>
<td>gotapi</td>
<td></td>
</tr>
<tr>
<td>profile</td>
<td>health</td>
<td></td>
</tr>
<tr>
<td>attribute</td>
<td>oximeter</td>
<td></td>
</tr>
<tr>
<td>clientId</td>
<td>1234567890</td>
<td></td>
</tr>
<tr>
<td>accessToken</td>
<td>0987654321</td>
<td></td>
</tr>
</tbody>
</table>
### 5.2.3 Response for one-shot measuring on the GotAPI-4 Interface

When the Plug-In receives the request, it SHALL respond to the GotAPI Server as follows:

#### Definition of the data object for the response

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>This value SHALL be &quot;RESPONSE&quot;.</td>
<td>Mandatory if the OS is not Android. Otherwise, optional. If the OS is Android, the &quot;Action&quot; value SHALL include this information as described below.</td>
</tr>
<tr>
<td>requestCode</td>
<td>int</td>
<td>The request code coming from the GotAPI Server.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>int</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code. This specification doesn't define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>oximeter</td>
<td></td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>device</td>
<td>Object</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>productName</td>
<td>String</td>
<td>The product name of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, it SHALL create a name for the pulse oximeter using an arbitrary algorithm. The algorithm is up to the Plug-In implementation, and this specification does not define any algorithms.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>manufacturerName</td>
<td>String</td>
<td>The manufacturer name of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>modelNumber</td>
<td>String</td>
<td>The model number of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Mandatory</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>firmwareRevision</td>
<td>String</td>
<td>The firmware revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>serialNumber</td>
<td>String</td>
<td>The serial number of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>softwareRevision</td>
<td>String</td>
<td>The software revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>hardwareRevision</td>
<td>String</td>
<td>The hardware revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>partNumber</td>
<td>String</td>
<td>The part number of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>protocolRevision</td>
<td>String</td>
<td>The protocol revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>systemId</td>
<td>String</td>
<td>The system id of the targeted pulse oximeter. This value SHALL be a 16-character HEX string without a '0x' prefix (e.g. &quot;ABCDEF0123456789&quot;). If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be &quot;0000000000000000&quot; (a string of 16 '0' characters).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Mandatory</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>batteryLevel</td>
<td>Float</td>
<td>The battery level of the targeted pulse oximeter. This value must be a float number in a range from 0.0 to 1.0. The value 0.0 represents that the targeted pulse oximeter is completely out of charge. The value 1.0 represents that the targeted pulse oximeter is fully charged. Even if the targeted pulse oximeter reports this value in percent in a range from 1 to 100, the Plug-In SHALL convert it to a float number in a range from 0.0 to 1.0. If the Plug-In can’t obtain battery level from the targeted pulse oximeter, this value SHALL be 1.0.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>spo2</td>
<td>Object</td>
<td>This value represents the SpO2 measured by the targeted pulse oximeter.</td>
<td>Mandatory if the &quot;pulse&quot; object below does not exist.</td>
</tr>
<tr>
<td>value</td>
<td>Float</td>
<td>This value represents the SpO2 measured by the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>mderFloat</td>
<td>String</td>
<td>This value represents the SpO2 measured by the targeted pulse oximeter, which is a hexadecimal string of a MDER FLOAT, such as “FFFFC8E”</td>
<td>Mandatory</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>This value represents the TYPE attribute as a human readable string and as its 32-bit MDC code such as “Oxygen Saturation”. If the Plug-In can’t obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>typeCode</td>
<td>String</td>
<td>This value represents the TYPE attribute, which is expressed by a code such as “150456” (This code means “Oxygen Saturation”). If the Plug-In can’t obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value represents the Supplemental types attribute as a human readable string and as its 32-bit MDC code such as “Spot (average) measurement”. If the Plug-In can’t obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value represents the Supplemental types attribute, which is expressed by a code such as “150588” (This code means “Spot (average) measurement”). If the Plug-In can’t obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value represents the unit of the reported SpO2, which is expressed by a human readable string such as “%”.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value represents the unit of the reported SpO2, which is expressed by a code such as “262688” (This code means “%”).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Mandatory</td>
</tr>
<tr>
<td>----------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>timeStamp</td>
<td>int</td>
<td>This value represents the measurement time when the measurement was done. If the measurement time is reported from the targeted pulse oximeter, the Plug-In SHALL convert it to a unix time stamp in millisecond. Otherwise, the Plug-In set this value to the unix time when the Plug-In receives the measurement value from the Plug-In based on the clock of the underlying operating system.</td>
<td></td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value represents the same time stamp as &quot;timeStamp&quot;. The format is &quot;YYYYMMDDHHMMSS.sss+-HHMM&quot;, such as &quot;20150504135813.220-0400&quot;.</td>
<td></td>
</tr>
<tr>
<td>pulse</td>
<td>Object</td>
<td>This value represents the SpO2, measured by the targeted pulse oximeter, which is a hexadecimal string of a MDER FLOAT, such as &quot;FFFFEC8E&quot;.</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>Float</td>
<td>This value represents the pulse rate measured by the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>mderFloat</td>
<td>String</td>
<td>This value represents the SpO2, measured by the targeted pulse oximeter, which is a hexadecimal string of a MDER FLOAT, such as &quot;FFFFEC8E&quot;.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>This value represents the TYPE attribute as a human readable string and as its 32-bit MDC code such as &quot;Pulse Rate&quot;. If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>typeCode</td>
<td>String</td>
<td>This value represents the TYPE attribute, which is expressed by a code such as &quot;149530&quot; (This code means &quot;Pulse Rate&quot;). If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value represents the Supplemental类型 attribute as a human readable string and as its 32-bit MDC code such as &quot;Spot (average) measurement&quot;. If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value represents the Supplemental类型 attribute, which is expressed by a code such as &quot;150588&quot; (This code means &quot;Spot (average) measurement&quot;). If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value represents the unit of the reported pulse rate, which is expressed by a human readable string such as &quot;beats per min&quot;.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value represents the unit of the reported pulse rate, which is expressed by a code such as &quot;264864&quot; (This code means &quot;beats per min&quot;).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>timeStamp</td>
<td>int</td>
<td>This value represents the measurement time when the measurement was done. If the measurement time is reported from the targeted pulse oximeter, the Plug-In SHALL convert it to a unix time stamp in millisecond. Otherwise, the Plug-In set this value to the unix time when the Plug-In receives the measurement value from the Plug-In based on the clock of the underlying operating system. Mandatory</td>
<td></td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value represents the same time stamp as &quot;timeStamp&quot;. The format is &quot;YYYYMMDDHHMMSS.sss+/-HHMM&quot;, such as &quot;20150504135813.220-0400&quot; Mandatory</td>
<td></td>
</tr>
</tbody>
</table>

The Plug-In MAY append additional data in the data object as needed.
This data object is sent to the GotAPI Server in an OS specific mechanism, e.g., Intents for Android.

**Requirements for OS-specific response channel and data container**

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request. The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

**Example of the data object of the Android Intents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>org.deviceconnect.action.RESPONSE</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;RESPONSE&quot;.</td>
</tr>
<tr>
<td>Component</td>
<td>org.deviceconnect</td>
<td>This value is the package name of the GotAPI Server application.</td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
</tr>
<tr>
<td>requestCode</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>oximeter device</td>
<td></td>
<td></td>
</tr>
<tr>
<td>productName</td>
<td>ABC Pulse Oximeter Pro</td>
<td></td>
</tr>
<tr>
<td>manufacturerName</td>
<td>ABC Inc.</td>
<td></td>
</tr>
<tr>
<td>modelNumber</td>
<td>TP-001</td>
<td></td>
</tr>
<tr>
<td>firmwareRevision</td>
<td>rev.1.001.003</td>
<td></td>
</tr>
<tr>
<td>serialNumber</td>
<td>01234-5678-9ABCD-EF01</td>
<td></td>
</tr>
<tr>
<td>softwareRevision</td>
<td>rev.2.000.000</td>
<td></td>
</tr>
<tr>
<td>hardwareRevision</td>
<td>rev.1.0</td>
<td></td>
</tr>
<tr>
<td>partNumber</td>
<td>002</td>
<td></td>
</tr>
<tr>
<td>protocolRevision</td>
<td>rev.3.1</td>
<td></td>
</tr>
</tbody>
</table>
5.2.4 Response for one-shot measuring on the GotAPI-1 Interface

When GotAPI Server receives the response from the Plug-In, the GotAPI Server passes it to the application as follows:

**Definition of the HTTP response**

<table>
<thead>
<tr>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIME-Type</strong></td>
</tr>
<tr>
<td><strong>HTTP status</strong></td>
</tr>
</tbody>
</table>

---

**Editor’s note:**

The extra data of Android is just a key-value structure. How should such structured data above be expressed? JSON string?

```java
intent.putExtra("oximeter", "{
  \"deviceProductName\": \"ABC Pulse Oximeter Pro\", ...
}"");
```
### Definition of the data object for the response

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>String</td>
<td>The name of the GotAPI Server (e.g. &quot;ABConnect&quot;)</td>
<td>Mandatory</td>
</tr>
<tr>
<td>version</td>
<td>String</td>
<td>The version of the GotAPI Server (e.g. &quot;1.0&quot;).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>Number</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code. This specification doesn't define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>oximeter</td>
<td></td>
<td><strong>device</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>productName</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>manufacturerName</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>modelNumber</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>firmwareRevision</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>serialNumber</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>softwareRevision</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>hardwareRevision</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>partNumber</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>protocolRevision</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>systemId</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>batteryLevel</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>spo2</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>value</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>mderFloat</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>type</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>typeCode</strong></td>
<td>Mandatory</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Mandatory</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStamp</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>pulse</td>
<td>Object</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>value</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>mderFloat</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>typeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStamp</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>hmac</td>
<td>String</td>
<td>An HMAC generated for the counter measure against the GotAPI Server spoofing attack. If the application includes a key for HMAC calculation in the API request, the GotAPI Server adds this value in the API response. Evaluating whether the HMAC is identical to the result of calculation of HMAC from the key, the application can ensure that the response is genuine.</td>
<td>Mandatory if the application provide a key to the GotAPI Server</td>
</tr>
</tbody>
</table>

The GotAPI Server SHALL serialize the data structure above as a JSON formatted stream (i.e. JSON string).

**Example of the response**

```json
{
  ...
}
```
"product" : "ABCConnect",
"version" : "1.0",
"requestCode" : 10,
"result" : 0,
"oximeter" : {
  "device" : {
    "productName" : "ABC Pulse Oximeter Pro",
    "manufacturerName" : "ABC Inc.",
    "modelNumber" : "TP-001",
    "firmwareRevision" : "rev.1.001.003",
    "serialNumber" : "01234-5678-9ABCD-EF01",
    "softwareRevision" : "rev.2.000.000",
    "hardwareRevision" : "rev.1.0",
    "partNumber" : "002",
    "protocolRevision" : "rev.3.1",
    "systemId" : "ABCDEF0123456789",
    "batteryLevel" : 0.5
  },
  "spo2" : {
    "value" : 98,
    "mderFloat" : "00000062",
    "type" : "Oxygen Saturation",
    "typeCode" : "150456",
    "supType" : "Spot (average) measurement",
    "supTypeCode" : "150588",
    "unit" : "%",
    "unitCode" : "262688",
    "timeStamp" : 1431856940275,
    "timeStampString" : "20150517100220.000-0000"
  },
  "pulse" : {
    "value" : 42,
    "mderFloat" : "0000002A",
    "type" : "Pulse Rate",
    "typeCode" : "149530",
    "supType" : "Spot (average) measurement",
    "supTypeCode" : "150588",
    "unit" : "beats per min",
    "unitCode" : "264864",
    "timeStamp" : 1431856940275,
    "timeStampString" : "20150517100220.000-0000"
  }
},
"hmac" : "0123456789"

5.3 Asynchronous messaging API

Asynchronous messaging API enables applications to receive measured data from pulse oximeters asynchronously using WebSocket as define in the Section 7.2.3 [DWAPI-PCH]. Asynchronous messaging API specification adheres to that of GotAPI 1.1.
As defined by GotAPI 1.1, after the application obtains authorization to access GotAPI-based APIs using the GotAPI-2 Interface and completes the Service Discovery, the application can use the service (so called "Asynchronous messaging API") provided by the Plug-In through the GotAPI Server.

The asynchronous messaging API offers a series of measurement values reported by the targeted device to an application in real time as the measurement values become available. The timing when and the reasons why such measurement values become available is determined by the Plug-Ins and connected devices, and is out of the scope of this specification.

This API uses WebSocket protocol to handle asynchronous event messages. The message flow of this API is shown blow:

![Message Flow of the Asynchronous messaging API](image)

**Figure 3: Message Flow of the Asynchronous messaging API**
This section defines the data object for the message flows labelled from (1) to (4) and from (8) to (13) described in the figure above.

5.3.1 Request for asynchronous messaging on the GotAPI-1 Interface

When the application uses the API in order to receive asynchronous messages, it sends a request to the GotAPI Server on the GotAPI-1 Interface as follows:

Definition of the HTTP request

<table>
<thead>
<tr>
<th>Method</th>
<th>Definition of the HTTP request</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP PUT</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Request URL</th>
<th>Definition of the request parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://127.0.0.1:4035/gotapi/health/oximeter">http://127.0.0.1:4035/gotapi/health/oximeter</a></td>
<td>serviceId: The identifier of the targeted service. This value is available from the Service Discovery API on the GotAPI-1 Interface. Mandatory</td>
</tr>
<tr>
<td><a href="https://127.0.0.1:4036/gotapi/health/oximeter">https://127.0.0.1:4036/gotapi/health/oximeter</a></td>
<td>accessToken: The access token obtained from the GotAPI Auth Server through the GotAPI-2 Interface. Mandatory</td>
</tr>
</tbody>
</table>

Example of the request URL

http://127.0.0.1:4035/gotapi/health/oximeter?serviceId=abcdefg123&accessToken=0987654321&nonce=93b3a219347

5.3.2 Request for asynchronous messaging on the GotAPI-4 Interface

When an application sends a request to the GotAPI Server on the GotAPI-1 Interface, the GotAPI Server passes the request to the Plug-In on the GotAPI-4 Interface. The request includes the data object as follows:

Definition of the data object for request

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
</tr>
</thead>
</table>
| method        | String | This value SHALL be "PUT".
|               |        | Mandatory if the OS is not Android. Otherwise, optional. If the OS is Android, the "Action" value SHALL include this information as described below. |
| receiver      | String | The address of the GotAPI Server application used by Plug-Ins. Generally, it is the application ID recognized by the OS, such as a package name. Mandatory |
| requestCode   | int    | A request code identifying the request. This value could be any number but must MUST be an integer greater than 0, and unique for each open request, to ensure responses can be correlated. Mandatory |
| serviceId     | String | The identifier of the targeted Service. This value is provided by the application over the GotAPI-1 Interface. Mandatory |
### api
- **String**
- The value must be "gotapi".
- Mandatory

### profile
- **String**
- The value must be "health".
- Mandatory

### attribute
- **String**
- The value must be "oximeter"
- Mandatory

### clientId
- **String**
- The identifier of the application, which is generated by the Plug-In when the Plug-In Approval procedure defined in the GotAPI specification.
- Mandatory

### accessToken
- **String**
- The access token for the application, which is generated by the Plug-In when the Plug-In Approval procedure defined in the GotAPI specification.
- Mandatory

This data object is sent to the Plug-Ins in an OS specific mechanism, e.g., Intents for Android.

### Requirements for OS-specific request channel and data container

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request. The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

### Example of the data object of the Android Explicit Intents

<table>
<thead>
<tr>
<th>Name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>org.deviceconnect.action.PUT</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;PUT&quot;.</td>
</tr>
<tr>
<td>Component</td>
<td>org.example.plugin</td>
<td>This value is the package name of the Plug-In application.</td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
</tr>
<tr>
<td>receiver</td>
<td>org.deviceconnect</td>
<td></td>
</tr>
<tr>
<td>requestCode</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>serviceId</td>
<td>dev1.example.org</td>
<td></td>
</tr>
<tr>
<td>api</td>
<td>gotapi</td>
<td></td>
</tr>
<tr>
<td>profile</td>
<td>health</td>
<td></td>
</tr>
<tr>
<td>attribute</td>
<td>oximeter</td>
<td></td>
</tr>
<tr>
<td>clientId</td>
<td>1234567890</td>
<td></td>
</tr>
<tr>
<td>accessToken</td>
<td>0987654321</td>
<td></td>
</tr>
</tbody>
</table>
5.3.3 Response for asynchronous messaging on the GotAPI-4 Interface

When the Plug-In receives the request, it SHALL respond to the GotAPI Server as follows:

**Definition of the data object for the response**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>This value SHALL be &quot;RESPONSE&quot;.</td>
<td>Mandatory if the OS is not Android. Otherwise, optional. If the OS is Android, the &quot;Action&quot; value SHALL include this information as described below.</td>
</tr>
<tr>
<td>requestCode</td>
<td>Number</td>
<td>The request code coming from the GotAPI Server.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>Number</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code. This specification doesn't define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>oximeter</td>
<td></td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>device</td>
<td>Object</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>productName</td>
<td>String</td>
<td>The product name of the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>manufacturerName</td>
<td>String</td>
<td>The manufacturer name of the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>modelNumber</td>
<td>String</td>
<td>The model number of the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>firmwareRevision</td>
<td>String</td>
<td>The firmware revision of the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>serialNum</td>
<td>String</td>
<td>The serial number of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>softwareRev</td>
<td>String</td>
<td>The software revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>hardwareRev</td>
<td>String</td>
<td>The hardware revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>partNumber</td>
<td>String</td>
<td>The part number of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>protocolRev</td>
<td>String</td>
<td>The protocol revision of the targeted pulse oximeter. If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be an empty string.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>systemId</td>
<td>String</td>
<td>The system id of the targeted pulse oximeter. This value SHALL be a 16-character HEX string without a '0x' prefix (e.g. &quot;ABCDEF0123456789&quot;). If the Plug-In cannot obtain this information from the targeted pulse oximeter, this value SHALL be &quot;0000000000000000&quot; (a string of 16 '0' characters).</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

The Plug-In MAY append additional data in the data object as needed.

This data object is sent to the GotAPI Server in an OS specific mechanism, e.g., Intents for Android.
Requirements for OS-specific response channel and data container

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request.</td>
</tr>
<tr>
<td></td>
<td>The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

Example of the data object of the Android Intents

<table>
<thead>
<tr>
<th>Name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>org.deviceconnect.action.RESPONSE</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;RESPONSE&quot;.</td>
</tr>
<tr>
<td>Component</td>
<td>org.deviceconnect</td>
<td>This value is the package name of the GotAPI Server application.</td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
</tr>
<tr>
<td>requestCode</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>oximeter</td>
<td>device</td>
<td></td>
</tr>
<tr>
<td></td>
<td>productName</td>
<td>ABC Pulse Oximeter Pro</td>
</tr>
<tr>
<td></td>
<td>manufacturerName</td>
<td>ABC Inc.</td>
</tr>
<tr>
<td></td>
<td>modelNumber</td>
<td>TP-001</td>
</tr>
<tr>
<td></td>
<td>firmwareRevision</td>
<td>rev.1.001.003</td>
</tr>
<tr>
<td></td>
<td>serialNumber</td>
<td>01234-5678-9ABCD-EF01</td>
</tr>
<tr>
<td></td>
<td>softwareRevision</td>
<td>rev.2.000.000</td>
</tr>
<tr>
<td></td>
<td>hardwareRevision</td>
<td>rev.1.0</td>
</tr>
<tr>
<td></td>
<td>partNumber</td>
<td>002</td>
</tr>
<tr>
<td></td>
<td>protocolRevision</td>
<td>rev.3.1</td>
</tr>
<tr>
<td></td>
<td>systemId</td>
<td>ABCDEF0123456789</td>
</tr>
</tbody>
</table>

Editor's note:
The extra data of Android is just a key-value structure. How should such structured data above be expressed? JSON string?

```java
intent.putExtra("oximeter", "{"deviceProductName":"ABC Pulse Oximeter Pro", ...}"");
```
5.3.4 Response for asynchronous messaging on the GotAPI-1 Interface

When GotAPI Server receives the response from the Plug-In, the GotAPI Server passes it to the application as follows:

Definition of the HTTP response

<table>
<thead>
<tr>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIME-Type</td>
</tr>
<tr>
<td>HTTP status</td>
</tr>
</tbody>
</table>

Definition of the data object for the response

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>String</td>
<td>The name of the GotAPI Server (e.g. &quot;ABConnect&quot;)</td>
<td>Mandatory</td>
</tr>
<tr>
<td>version</td>
<td>String</td>
<td>The version of the GotAPI Server (e.g. &quot;1.0&quot;).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>Number</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code. This specification doesn't define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>oximeter</td>
<td>Object</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>device</td>
<td>Object</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>productName</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>manufacturerName</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>modelNumber</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>firmwareRevision</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>serialNumber</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>softwareRevision</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>hardwareRevision</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>partNumber</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>protocolRevision</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>systemId</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>hmac</td>
<td>String</td>
<td>An HMAC generated for the counter measure against the GotAPI Server spoofing attack. If the application includes a key for HMAC calculation in the API request, the GotAPI Server adds this value in the API response. Evaluating whether the HMAC is identical to</td>
<td>Mandatory if the application provide a key to the GotAPI Server</td>
</tr>
</tbody>
</table>
the result of calculation of HMAC from the key, the application can ensure that the response is genuine.

The GotAPI Server SHALL serialize the data structure above as a JSON formatted stream (i.e. JSON string).

**Example of the response**

```json
{
    "product" : "ABCConnect",
    "version" : "1.0",
    "requestCode" : 10,
    "result" : 0,
    "oximeter" : {
        "device": {
            "productName" : "ABC Pulse Oximeter Pro",
            "manufacturerName" : "ABC Inc.",
            "modelNumber" : "TP-001",
            "firmwareRevision" : "rev.1.001.003",
            "serialNumber" : "01234-5678-9ABCD-EF01",
            "softwareRevision" : "rev.2.000.000",
            "hardwareRevision" : "rev.1.0",
            "partNumber" : "002",
            "protocolRevision" : "rev.3.1",
            "systemId" : "ABCDEF0123456789"
        }
    },
    "hmac" : "0123456789"
}
```

### 5.3.5 Asynchronous message from the Plug-In to the GotAPI Server on the GotAPI-4 Interface

The Plug-In sends an asynchronous message as follows:

**Definition of the data object for request**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>This value SHALL be &quot;EVENT&quot;.</td>
<td>Mandatory if the OS is not Android. Otherwise, optional. If the OS is Android, the &quot;Action&quot; value SHALL include this information as described below.</td>
</tr>
<tr>
<td>requestCode</td>
<td>int</td>
<td>The request code coming from the GotAPI Server.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>Number</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code. This specification doesn't define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>oximeter</td>
<td>Object</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>device</td>
<td>Object</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>batteryLevel</td>
<td>Float</td>
<td>The battery level of the targeted oximeter. This value must be a float number in a range from 0.0 to 1.0. The value 0.0 represents that the targeted oximeter is completely out of charge. The value 1.0 represents that the targeted oximeter is fully charged. Even if the targeted oximeter reports this value in percent in a range from 1 to 100, the Plug-In SHALL convert it to a float number in a range from 0.0 to 1.0. If the Plug-In can't obtain battery level from the targeted oximeter, this value SHALL be 1.0.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>spo2</td>
<td>Object</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In. Mandatory if the &quot;pulse&quot; object above does not exist.</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>mderFloat</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>typeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStamp</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>pulse</td>
<td>Object</td>
<td>This value represents the pulse rate measured by the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>value</td>
<td>Float</td>
<td>This value represents the pulse rate measured by the targeted pulse oximeter.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>mderFloat</td>
<td>String</td>
<td>This value represents the SpO₂ measured by the targeted pulse oximeter, which is a hexadecimal string of a MDER FLOAT, such as “FFFFFF8E”.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Mandatory</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>This value represents the TYPE attribute as a human readable string and as its 32-bit MDC code such as &quot;Pulse Rate&quot;. If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td></td>
</tr>
<tr>
<td>typeCode</td>
<td>String</td>
<td>This value represents the TYPE attribute, which is expressed by a code such as &quot;149530&quot; (This code means &quot;Pulse Rate&quot;). If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td></td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value represents the Supplemental types attribute as a human readable string and as its 32-bit MDC code such as &quot;Spot (average) measurement&quot;. If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td></td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value represents the Supplemental types attribute, which is expressed by a code such as &quot;150588&quot; (This code means &quot;Spot (average) measurement &quot;). If the Plug-In can't obtain the type, this value SHALL be an empty string.</td>
<td></td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value represents the unit of the reported pulse rate, which is expressed by a human readable string such as &quot;beats per min&quot;.</td>
<td></td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value represents the unit of the reported pulse rate, which is expressed by a code such as &quot;264864&quot; (This code means &quot;beats per min&quot;).</td>
<td></td>
</tr>
<tr>
<td>timeStamp</td>
<td>int</td>
<td>This value represents the measurement time when the measurement was done. If the measurement time is reported from the targeted pulse oximeter, the Plug-In SHALL convert it to a unix time stamp in millisecond. Otherwise, the Plug-In set this value to the unix time when the Plug-In receives the measurement value from the Plug-In based on the clock of the underlying operating system.</td>
<td></td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value represents the same time stamp as &quot;timeStamp&quot;. The format is &quot;YYYYMMDDHHMMSS.sss+/-HHMM&quot;, such as &quot;20150504135813.220-0400&quot;</td>
<td></td>
</tr>
</tbody>
</table>

The Plug-In MAY append additional data in the data object as needed.

This data object is sent to the Plug-Ins in an OS specific mechanism, e.g., Intents for Android.

**Requirements for OS-specific request channel and data container**

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request. The data object must be mapped to the Extra directly.</td>
<td></td>
</tr>
</tbody>
</table>

**Example of the data object of the Android Explicit Intents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Extra key name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>org.deviceconnect.action.EVENT</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;EVENT&quot;.</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>org.example.plugin</td>
<td>This value is the package name of the Plug-In application.</td>
<td></td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>requestCode</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>result</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oximeter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>device</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>deviceBatteryLevel</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spo2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mderFloat</td>
<td>00000062</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Oxygen Saturation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>typeCode</td>
<td>150456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>supType</td>
<td>Spot (average) measurement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>supTypeCode</td>
<td>150588</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unit</td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unitCode</td>
<td>262688</td>
<td></td>
<td></td>
</tr>
<tr>
<td>timeStamp</td>
<td>1431856940275</td>
<td></td>
<td></td>
</tr>
<tr>
<td>timeStampString</td>
<td>20150517100220.000-0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pulse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mderFloat</td>
<td>0000002A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Pulse Rate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>typeCode</td>
<td>149530</td>
<td></td>
<td></td>
</tr>
<tr>
<td>supType</td>
<td>Spot (average) measurement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>supTypeCode</td>
<td>150588</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unit</td>
<td>beats per min</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unitCode</td>
<td>264864</td>
<td></td>
<td></td>
</tr>
<tr>
<td>timeStamp</td>
<td>1431856940275</td>
<td></td>
<td></td>
</tr>
<tr>
<td>timeStampString</td>
<td>20150517100220.000-0000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 5.3.6 Asynchronous message from the GotAPI Server to the application on the GotAPI-5 Interface

When the GotAPI Server receives an asynchronous message from the Plug-In, the GotAPI Server passes it to the application on the GotAPI-5 Interface. The format of the data is a JSON string as follows:

**Definition of the data object**

<table>
<thead>
<tr>
<th>Name</th>
<th>Sub name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceId</td>
<td></td>
<td>String</td>
<td>The identifier of the targeted Service. This value is provided by the application when the application send the originated API request on the GotAPI-1 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>oximeter</td>
<td></td>
<td>Object</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>device</td>
<td></td>
<td>Object</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>batteryLevel</td>
<td></td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>spo2</td>
<td></td>
<td>Object</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory if the &quot;pulse&quot; object below does not exist.</td>
</tr>
<tr>
<td>value</td>
<td></td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>mderFloat</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>type</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>typeCode</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supType</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>supTypeCode</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unit</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>unitCode</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeStamp</td>
<td></td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>timeString</td>
<td></td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

**Editor's note:**
The extra data of Android is just a key-value structure. How should such structured data above be expressed? JSON string?

```java
intent.putExtra("oximeter", "{"deviceProductName":"ABC Pulse Oximeter Pro", ...");
```
<table>
<thead>
<tr>
<th>pulse</th>
<th>Object</th>
<th>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mandatory if the &quot;spo2&quot; object above does not exist.</td>
</tr>
<tr>
<td>value</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>mderFloat</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>typeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>supType</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>supTypeCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>unitCode</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>timeStamp</td>
<td>Number</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
<tr>
<td>timeStampString</td>
<td>String</td>
<td>This value SHALL be the same as that which the GotAPI Server received from the Plug-In.</td>
</tr>
</tbody>
</table>

| hmac | String | An HMAC generated for the counter measure against the GotAPI Server spoofing attack. |
|      |        | If the application includes a key for HMAC calculation in the API request, the GotAPI Server adds this value in the API response. Evaluating whether the HMAC is identical to the result of calculation of HMAC from the key, the application can ensure that the response is genuine. |
|      |        | Mandatory if the application provide a key to the GotAPI Server |

Example of the JSON string
```
{
  "serviceId" : 0,
  "oximeter" : {
    "device": {
      "batteryLevel" : 0.5
    },
  "spo2" : {
    "value" : 98,
    "mderFloat" : "00000062",
    "type" : "Oxygen Saturation",
    "typeCode" : "150456",
    "supType" : "Spot (average) measurement",
    "supTypeCode" : "150588",
    "unit" : "%",
    "unitCode" : "262688",
    "timeStamp" : 1431856940275,
    "timeStampString" : "20150517100220.000-0000"
  }
}
```
5.3.7 Stop request from the application to the GotAPI Server on the GotAPI-1 Interface

When the application wants to stop receiving asynchronous messages, it sends a request to the GotAPI Server on the GotAPI-1 Interface as follows:

Definition of the HTTP request

<table>
<thead>
<tr>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>Request URL</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Definition of the request parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceId</td>
<td>The identifier of the targeted service. This value is available from the Service Discovery API on the GotAPI-1 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>accessToken</td>
<td>The access token obtained from the GotAPI Auth Server through the GotAPI-2 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>nonce</td>
<td>A nonce generated by the application, which is described in the section &quot;7.3.3.3 HMAC server authentication using trusted Application ID for the Server spoofing attack&quot; in the GotAPI specification.</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Example of the request URL

http://127.0.0.1:4035/gotapi/health/oximeter?serviceId=abcdefg123&accessToken=0987654321&nonce=93b3a219347
5.3.8 Stop request from the GotAPI Server to the Plug-In on the GotAPI-4 Interface

When the GotAPI Server receives a stop request from the application on the GotAPI-1 Interface, the GotAPI Server sends a stop request to the Plug-in on the GotAPI-4 Interface. The request includes the data object as follows:

**Definition of the data object for request**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>This value SHALL be &quot;DELETE&quot;.</td>
<td>Mandatory if the OS is not Android. Otherwise, optional.</td>
</tr>
<tr>
<td>receiver</td>
<td>String</td>
<td>The address of the GotAPI Server application used by Plug-Ins. Generally, it is the application ID recognized by the OS, such as a package name.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>requestCode</td>
<td>int</td>
<td>A request code identifying the request. This value could be any number but must MUST be an integer greater than 0, and unique for each open request, to ensure responses can be correlated.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>serviceId</td>
<td>String</td>
<td>The identifier of the targeted Service. This value is provided by the application over the GotAPI-1 Interface.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>api</td>
<td>String</td>
<td>The value must be &quot;gotapi&quot;.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>profile</td>
<td>String</td>
<td>The value must be &quot;health&quot;.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>attribute</td>
<td>String</td>
<td>The value must be &quot;oximeter&quot;</td>
<td>Mandatory</td>
</tr>
<tr>
<td>clientId</td>
<td>String</td>
<td>The identifier of the application, which is generated by the Plug-In when the Plug-In Approval procedure defined in the GotAPI specification.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>accessToken</td>
<td>String</td>
<td>The access token for the application, which is generated by the Plug-In when the Plug-In Approval procedure defined in the GotAPI specification.</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

This data object is sent to the Plug-Ins in an OS specific mechanism, e.g., Intents for Android.

**Requirements for OS-specific request channel and data container**

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request.</td>
</tr>
<tr>
<td></td>
<td>The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

**Example of the data object of the Android Explicit Intents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>org.deviceconnect.action.DELETE</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;DELETE&quot;.</td>
</tr>
<tr>
<td>Component</td>
<td>org.example.plugin</td>
<td>This value is the package name of the Plug-In application.</td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.3.9 **Stop response from the Plug-In to the GotAPI Server on the GotAPI-4 Interface**

When the Plug-In receives the stop request, it SHALL respond as follows:

**Definition of the data object for the response**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>String</td>
<td>This value SHALL be &quot;RESPONSE&quot;.</td>
<td>Mandatory if the OS is not Android. Otherwise, optional.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the OS is Android, the &quot;Action&quot; value SHALL include this information as described below.</td>
<td></td>
</tr>
<tr>
<td>requestCode</td>
<td>Number</td>
<td>The request code coming from the GotAPI Server.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>Number</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code.</td>
<td>Mandatory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This specification doesn't define error codes.</td>
<td></td>
</tr>
</tbody>
</table>

The Plug-In MAY append additional data in the data object as needed.

This data object is sent to the GotAPI Server in an OS specific mechanism, e.g., Intents for Android.

**Requirements for OS-specific response channel and data container**

<table>
<thead>
<tr>
<th>OS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>The GotAPI Server must use Explicit Intents for the request.</td>
</tr>
<tr>
<td></td>
<td>The data object must be mapped to the Extra directly.</td>
</tr>
</tbody>
</table>

**Example of the data object of the Android Intents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Sub name</th>
<th>Example of value</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td></td>
<td>org.deviceconnect.action.RESPONSE</td>
<td>This value is defined by the GotAPI Server application. But the last part SHALL be &quot;RESPONSE&quot;.</td>
</tr>
<tr>
<td>Component</td>
<td></td>
<td>org.deviceconnect</td>
<td>This value is the package name of the GotAPI Server application.</td>
</tr>
<tr>
<td>Extra</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>requestCode</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
5.3.10 Stop response from the GotAPI Server to the application on the GotAPI-1 Interface

When the GotAPI Server receives the stop response, the GotAPI Server passes the response to the application follows:

Definition of the HTTP response

<table>
<thead>
<tr>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIME-Type: application/json</td>
</tr>
<tr>
<td>HTTP status: 200 OK</td>
</tr>
</tbody>
</table>

Definition of the data object for the response

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Definition of value</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>String</td>
<td>The name of the GotAPI Server (e.g. &quot;ABConnect&quot;).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>version</td>
<td>String</td>
<td>The version of the GotAPI Server (e.g. &quot;1.0&quot;).</td>
<td>Mandatory</td>
</tr>
<tr>
<td>result</td>
<td>Number</td>
<td>If success, the value is 0, otherwise an integer greater than 0, which indicates an error code. This specification doesn’t define error codes.</td>
<td>Mandatory</td>
</tr>
<tr>
<td>hmac</td>
<td>String</td>
<td>An HMAC generated for the counter measure against the GotAPI Server spoofing attack. If the application includes a key for HMAC calculation in the API request, the GotAPI Server adds this value in the API response. Evaluating whether the HMAC is identical to the result of calculation of HMAC from the key, the application can ensure that the response is genuine.</td>
<td>Mandatory if the application provide a key to the GotAPI Server</td>
</tr>
</tbody>
</table>

The GotAPI Server SHALL serialize the data structure above as a JSON formatted stream (i.e. JSON string), then send it to the originating application on the GotAPI-5 (WebSocket connection).

Example of the response

```
{
    "product": "ABCConnect",
    "version": "1.0",
    "result" : 0,
    "hmac"   : "0123456789"
}
```
Appendix A. Change History (Informative)

A.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-TS-Pulse_Oximeter_APIs-V1_0-20180724-A</td>
<td>24 Jul 2018</td>
<td>Status changed to Approved by CD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Doc Ref # OMA-CD-2018-0005-INP_DWAPI_V1_0_ERP_for_final_Approval</td>
</tr>
</tbody>
</table>

© 2018 Open Mobile Alliance All Rights Reserved.
Used with the permission of the Open Mobile Alliance under the terms as stated in this document.