RPC methods

Homegear Cloudconnect provides the following RPC methods:

getDeviceDescription

```
(1) Struct getDeviceDescription(Integer peerId)
(2) Struct getDeviceDescription(String interfaceId)
(3) Struct getDeviceDescription()
```

This method returns information about an actual device connected to Homegear `(1)`, a communication interface `(2)` or the system `(3)`.

Parameters

Parameter	Description
`peerId`	The Homegear ID of the peer information should be returned for.
`interfaceId`	The Homegear ID of the communication interface information should be returned for.

Return values

Returns a `Struct` containing information about the requested entity.

Signature `(1)` (device)

The returned `Struct` has the following properties:

Property	Type	Optional	Description
`type`	`String`	no	The name of the product.
`name`	`String`	yes	The name assigned to the device.
`equipmentId`	`String`	yes	The equipment ID of the device (e. g. the German AKS code).

Property	Type	Optional	Description
`parent`	`String`	yes	The parent of this device. This entry is a string that contains the location ID of the edge client and the parent's peer ID in the following form: `<location ID>_<peer ID>`. E. g. `21:15211.2_23`. Careful: `location ID` must be the location ID of the edge client the parent device is connected to and NOT the location ID of the parent device.
`parentDirection`	`Integer`	yes	Only set when `parent` is set. `0` for input or subtraction, `1` for output or addition.
`serialNumber`	`String`	no	The serial number of the device.
`softwareVersion`	`String`	no	The software or firmware version of the device.
`description`	`String`	yes	A short description of the device.
`longDescription`	`String`	yes	A longer description of the device.
`interface`	`String`	no	The interface used to communicate with the device.
`roles`	`Array<String>`	yes	The list of all roles assigned to the device. When empty, the device can be ignored. Not all roles require an implementation. Roles of level 0 (first two digits) and level 1 (second two digits) are placed here.
`tags`	`Array<String>`	yes	A list of tags to further specify what the device is used for. E. g. `heat-pump` for a power meter when the power meter is used to measure the power usage of a heat pump. When the tag is prefixed with `section-` (e. g. `section-primary-facilities`), this device is associated to this section. Multiple sections are allowed.
`dataPoints`	`Struct`	yes	

Property	Type	Optional	Description
			The list of data points that are being exposed to Sensaru Cloud.

`dataPoints` is a `Struct`, which contains the channel numbers as property name. The value of the channel property is a `Struct` with the datapoint's id as property name and a `Struct` with the following properties as value:

Property	Type	Optional	Description
`type`	`String`	no	The type of the datapoint. One of: `boolean`, `string`, `action`, `integer`, `enum`, `double`, `array`, `struct`
`operations`	`Array<String>`	no	Possible elements are: `transmit`, `read` and `write`. Only `read` and `write` are relevant.
`flags`	`Array<String>`	no	Possible elements are: `visible`, `internal`, `transform`, `service` and `sticky`. See the Homegear documentation for a description of these elements.
`minimumValue`	`Double` or `Integer`	yes	The minimum possible value of the datapoint. Only relevant for numerical datapoints.
`maximumValue`	`Double` or `Integer`	yes	The maximum possible value of the datapoint. Only relevant for numerical datapoints.
`special`	`Struct`	yes	Special values with string identifiers. These values can be outside of the scope specified by `minimumValue` and `maximumValue`. The property name of the `Struct` is the string identifier. The value is the special value.
`valueList`	`Array<String>`	yes	Defines the names for a datapoint of type `enum`. The array index of the name is the associated value. If an array element is an empty string, the value is not defined.
`unit`	`String`	yes	The unit of the datapoint (e. g. `°C`).
`priority`	`Integer`	no	A priority between 1 and 5, where `1` means `critical`, `2` `error`, `3` `warning`, `4` `info` and `5` `debug`.

Property	Type	Optional	Description
`au`	`String`	yes	Only set when the location ID of the variable differs from the location ID of the edge client.
`roles`	`Array<Struct>`	yes	The list of roles assigned to the data point. Every `Struct` has the following properties: `id` : The role ID `level` : The role level: 0 (first two digits are not zero, all digits after this are zero), 1 (second two digits are not zero, last two digits are zero) or 2 (last two digits are not zero).
`aggregation`	`String`	`no`	A recommendation on how to aggregate the data point. Possible values are: `none` , `count` (e. g. strings and actions), `count-distinct` (e. g. enumerations and booleans), `average` and `meter` .
`label`	`Struct`	yes	The label of the datapoint. The property name of the `Struct` is the language code and property value the translation. The language code can be either only the ISO 639 language code or the ISO 639 language code and the ISO 3166 country code separated by a dash (`-`).
`description`	`Struct`	yes	The description of the datapoint. The property name of the `Struct` is the language code and property value the translation. The language code can be either only the ISO 639 language code or the ISO 639 language code and the ISO 3166 country code separated by a dash (`-`).

EXAMPLE:


```
{
 "id": 12,
 "jsonrpc": "2.0",
 "result": {
  "dataPoints": {
   "2": {
    "CO2": {
     "flags": [
      "visible"
     ],
     "maximumValue": 670760,
     "minimumValue": 0,
     "operations": [
      "transmit",
```

```
      "read"
    ],
    "priority": 4,
    "type": "double",
    "unit": "ppm"
  },
  "HUMIDITY": {
    "flags": [
      "visible"
    ],
    "maximumValue": 670760,
    "minimumValue": 0,
    "operations": [
      "transmit",
      "read"
    ],
    "priority": 4,
    "roles": [
      {
        "direction": 0,
        "id": 700004,
        "level": 2
      },
      {
        "direction": 0,
        "id": 700000,
        "level": 0
      }
    ],
    "type": "double",
    "unit": "%"
  },
  "TEMPERATURE": {
    "flags": [
      "visible"
    ],
    "maximumValue": 670760,
    "minimumValue": -273,
    "operations": [
      "transmit",
      "read"
    ],
    "priority": 4,
    "roles": [
      {
        "direction": 0,
        "id": 700010,
        "level": 2
      },
      {
        "direction": 0,
        "id": 700000,
        "level": 0
      }
    ],
    "type": "double",
    "unit": "°C"
```

```
      }
     }
    },
    "interface": "KNX",
    "name": "Schlafzimmer - Präsenzmelder",
    "roles": [
     700000
    ],
    "tags": [
     "my-tag-1",
     "my-tag-2"
    ],
    "serialNumber": "KNX000000024",
    "softwareVersion": "1.0",
    "type": "2.2.3"
   }
}
```


Signature (2) (interface)

The returned `Struct` has the following properties:

Property	Type	Optional	Description
`name`	`String`	`no`	The name of the interface.
`serialNumber`	`String`	yes	The serial number of the interface.
`softwareVersion`	`String`	yes	The software of firmware version of the interface.
`associatedDevices`	`Array<Integer>`	no	An array with all peer IDs associated to this interface.
`dataPoints`	`Struct`	`yes`	The list of data points that are being exposed to Sensaru Cloud.

The `dataPoints` `Struct` is the same as in signature `(1)`.

EXAMPLE:


```
{
 "id": 12,
 "jsonrpc": "2.0",
 "result": {
  "name": "KNX interface (KNX)",
  "associatedDevices": [12, 27, 28, 49, 54],
  "dataPoints": {
   "online": {
```

```
      "flags": [
        "visible",
        "service"
      ],
      "operations": [
        "read"
      ],
      "priority": 2,
      "type": "boolean"
    }
  },
  "serialNumber": "ENO14230050002",
  "softwareVersion": "15"
 }
}
```


Signature (3) (system)

The returned `Struct` has the following properties:

Property	Type	Optional	Description
`hasPhysicalHa`	`Boolean`	`no`	`true` when this is a high-availability environment using two physical devices in the same installation. When `true`, there are two devices: A `master` and a `slave` device. Otherwise there is only one physical device.
`hasVirtualHa`	`Boolean`	`no`	`true` when this is a high-availability environment using one physical device and one virtual backup. The virtual device does not have a serial number or firmware version.
`masterSerialNumber`	`String`	`no`	Serial number of master device.
`slaveSerialNumber`	`String`	`yes`	Serial number of slave device.
`masterFirmwareVersion`	`String`	`no`	Firmware version of master device.
`slaveFirmwareVersion`	`String`	`yes`	Firmware version of slave device.
`distribution`	`String`	`no`	The linux distribution installed on the device.

Property	Type	Optional	Description
`codename`	`String`	`no`	The codename of the linux distribution installed on the device.
`architecture`	`String`	`no`	The architecture of the system (e. g. `amd64` or `armhf`).
`version`	`String`	`no`	The Homegear version installed on the system.
`activeHaInstance`	`String`	`yes`	Only available in high-availability environments. Can be either `master` or `slave`.
`associatedInterfaces`	`Array<String>`	no	An array with the IDs of all communication interfaces associated to this edge client.
`dataPoints`	`Struct`	`yes`	The list of data points that are being exposed to Sensaru Cloud.

The `dataPoints` `Struct` is the same as in signature `(1)`.

EXAMPLE:


```
{
 "id": 12,
 "jsonrpc": "2.0",
 "result": {
  "activeHaInstance": "master",
  "architecture": "armhf",
  "associatedInterfaces": [
   "5.virtual",
   "My-Sonos-1234",
   "BK90x0",
   "KNX",
   "ENO14230050002",
   "254.virtual"
  ],
  "codename": "bullseye",
  "distribution": "Raspbian",
  "hasPhysicalHa": true,
  "masterFirmwareVersion": "15",
  "masterSerialNumber": "COR14260430001",
  "slaveFirmwareVersion": "13",
  "slaveSerialNumber": "COR14260430002",
  "version": "0.8.0-3392"
```

```
  }
}
```