

# hub.info.get

Description: Return hub info (mode, firmware, hardware, ...)

## parameters:

No parameters required.

## return result fields:

Field	Type	Required	Description
<b>model</b>	string	+	Production name
<b>architecture</b>	string	+	SOC architecture
<b>firmware</b>	string	+	Firmware version
<b>kernel</b>	string	+	Kernel version (for atom32 ESP-SDK version used)
<b>hardware</b>	string	+	Hardware version
<b>serial</b>	string	+	Serial number
<b>uuid</b>	string	+	Controller's UUID
<b>offlineAnonymousAccess</b>	bool	+	Is anonymous access (without authentication via hub.offline.login.ui for requests via WS API or HTTP basic authentication for requests via HTTP API) allowed for local connections
<b>offlineInsecureAccess</b>	bool	+	Is insecure access (WS and HTTP as opposed to secure WSS and HTTPS) allowed for local connections
<b>location</b>	JsonObj	+	Location information
<b>location.latitude</b>	float32	+	Latitude value
<b>location.longitude</b>	float32	+	Longitude value

<b>location.timezone</b>	string	+	Time zone name. Can be empty on internal error. Check logs on the hub for the error message (/var/log/firmware/ha-uid.log for H2)
<b>location.state</b>	string	+	Can contain one of possible values: default, customAll, customTimezone, customCoordinates
<b>build</b>	JsonObj	+	Build info
<b>build.time</b>	string	+	Time and date when the build was made (ISO 8601: YYYY-MM-DDThh:mm:ss±hhmm)
<b>build.builder</b>	string	+	Where the firmware was built (user@host)
<b>build.branch</b>	string	+	Branch name
<b>build.commit</b>	string	+	Commit hash
<b>battery</b>	object	-	If controller have no battery support whole section is not returned
<b>battery.status</b>	enum	-	initializing, missing, charging, full, discharging
<b>battery.stateOfCharge</b>	int	-	Percent of available battery capacity
<b>battery.remainingTime</b>	int	-	Remaining time in minutes to empty if discharging, or to full if charging
<b>battery.health</b>	int	-	Percent of current full capacity relative to original full capacity
<b>uptime</b>	string	+	Hub uptime
<b>localtime</b>	string	+	Current time on hub (ISO 8601: YYYY-MM-DDThh:mm:ss±hhmm)

Here is it an example of usage:

**call:**

```
{
  "method": "hub.info.get",
  "id": "_ID_",
  "params": {}
}
```

**reply:**

```
{
  "error": null,
  "id": "_ID_",
  "result": {
    "model": "ATOM32",
    "architecture": "esp32",
    "firmware": "0.9.2",
    "kernel": "v3.3-dev-239-g18118a6d5",
    "hardware": "rev1",
    "serial": "7000183",
    "uuid": "18980500-bf7b-11ea-8603-853e1e71d301",
    "location": {
      "latitude": 46.4828401,
      "longitude": 30.7595042,
      "timezone": "Europe/Kiev",
      "state": "custom"
    },
    "build": {
      "time": "2004-05-23T14:25:10+0200",
      "builder": "jenkins@builder1",
      "branch": "qa",
      "commit": "cc2b9921c8572147d762674eacb9b02974ece302"
    },
    "battery": {
      "stateOfCharge": 87,
      "remainingTime": 236,
      "status": "discharging",
      "health": 91
    },
    "uptime": "5d 22h 18m 11s",
    "localtime": "2004-05-23T14:25:10+0200"
  }
}
```

## hub.reboot

Starts hub to reboot

**call:**

```
{
  "api" : "1.0"
  "id": "_ID_",
  "method": "hub.reboot",
  "params": {}
}
```

**reply:** Reply by json-RPC protocol means that command successfully received by hub

```
{
  "error": null,
  "id": "_ID_",
  "result": {}
}
```

# hub.data.list

The command allows to request data from multiple collections from a hub in one request, allows data filtering by id and field names.

## Request

Default request:

```
{
  "api": "1.0",
  "method": "hub.data.list",
  "id": "<REQUEST_ID>",
  "params": {
    "devices":null,
    "items":null,
    "rooms": null,
    "scenes":null
  }
}
```

Default request, with all fields filled:

```
{
  "api": "1.0",
  "method": "hub.data.list",
  "id": "<REQUEST_ID>",
  "params": {
    "devices":{
      "ids":[], "fields": { "include":[] }
    },
    "items":{
      "ids":[], "fields": { "include":[] }
    },
    "rooms":{
      "ids":[], "fields": { "include":[] }
    },
    "scenes":{
      "ids":[], "fields": { "include":[] }
    }
  }
}
```

## Parameters:

devices, items, rooms, scenes

All of them are optional, but at least one is required.

Name	Required	Description
devices	-	If this field is present: a response with the <i>devices</i> will be sent anyway. If it has null value then all <i>devices</i> will be sent.

devices.ids	-	Array of IDs of requested <i>devices</i> . Only requested <i>devices</i> will be in the answer. If the controller doesn't have a requested <i>device</i> then null will be sent instead of <i>device</i> data.
devices.fields	-	Description of fields filtering. If this field is missed then all fields will be present in answer.
devices.fields.include	-	Name of fields which should be in answer. Possible values: <i>parentDeviceId, category, subcategory, deviceTypeId, gatewayId, name, type, batteryPowered, reachable, armed, roomId, persistent, info, serviceNotification, security, ready, status</i>
items	-	If this field is present: a response with the <i>items</i> will be sent anyway. If it has null value then all <i>items</i> will be sent.
items.ids	-	Array of IDs of requested <i>items</i> . Only requested <i>items</i> will be in the answer. If the controller doesn't have requested <i>item</i> then null will be sent instead of <i>item</i> data.
items.fields	-	Description of fields filtering. If this field is missed then all fields will be present in answer.
items.fields.include	-	Name of fields which should be in answer. Possible values: <i>deviceId, enum, hasGetter, hasSetter, name, show, scale, valueType, valueFormatted, value, minValue, maxValue, elementsMaxNumber, userCodeRestriction, elementsMaxNumberPerArray, oneWeekDayCost, oneShiftedWeekDayCost</i>
rooms	-	If this field is present: a response with the <i>rooms</i> will be sent anyway. If it has null value then all <i>rooms</i> will be sent.
rooms.ids	-	Array of IDs of requested <i>rooms</i> . Only requested <i>rooms</i> will be in the answer. If the controller doesn't have requested <i>room</i> then null will be sent instead of <i>room</i> data.
rooms.fields	-	Description of fields filtering. If this field is missed then all fields will be present in answer.
rooms.fields.include	-	Name of fields which should be in answer. Possible values: <i>enabled, group_id, is_group, name, parent_id, then, when</i>
scenes	-	If this field is present: a response with the <i>scenes</i> will be sent anyway. If it has null value then all <i>scenes</i> will be sent.

scenes.ids	-	Array of IDs of requested <i>scenes</i> . Only requested <i>scenes</i> will be in the answer. If the controller doesn't have requested <i>room</i> then null will be sent instead of <i>room</i> data.
scenes.fields	-	Description of fields filtering. If this field is missed then all fields will be present in the answer.
scenes.fields.include	-	Name of fields which should be in answer. Possible values: <i>enabled</i> , <i>group_id</i> , <i>is_group</i> , <i>name</i> , <i>parent_id</i> , <i>then</i> , <i>when</i>

**Errors:**

Type of error	Error code	Reason	Data Field
Invalid parameters	-32602	Empty params	rpc.params.invalid
Wrong format of request fields	-32602	Wrong params	rpc.params.invalid.[NAME]

**Answers:** Request will send a few answers with the same request and sender info in each answer. Each type of controller sends in a separate answer.

```
{
  "api": "1.0",
  "error": null,
  "method": "hub.data.list",
  "id": "<REQUEST_ID>",
  "result": {
    "devices": {
      "5e6b361746abd40dfb1494c1":{
        "name": "Smart Plug",
        "reachable": true,
        "armed": false,
        "serviceNotification": false,
        "roomId": "",
        "status": "idle"
      }
    }
  }
}

{
  "api": "1.0",
  "error": null,
  "method": "hub.data.list",
  "id": "<REQUEST_ID>",
  "result": {
    "items": {
      "5e6b361746abd40dfb1494c2":{
        "value": "idle"
      },
      "5e6b361746abd40dfb1494c3":null
    }
  }
}

{
  "api": "1.0",
  "error": null,
  "method": "hub.data.list",
  "id": "<REQUEST_ID>",
```

```
"result": {
  "rooms": {
    "8e6b361746abd40dfb1494c4":{
      "name": "Living Room"
    },
    "8e6b361746abd40dfb149417":{
      "name": "Hall"
    }
  }
}
}
{
  "api": "1.0",
  "error": null,
  "method": "hub.data.list",
  "id": "<REQUEST_ID>",
  "result": {
    "scenes": {
      "5c7ff48b7f00002a07a408e3":{
        "name": "Scene1"
      },
      "5c6ec961cc01eb07f86f9dd9":{
        "name": "Test Scene"
      }
    }
  }
}
}
```