

Version: 1.0

Illuminance

The Illuminance component handles the monitoring of device's illuminance sensors. Illuminance components are identified with `illuminance:<id>` in objects containing multiple component payloads.

The Illuminance component uses `Illuminance` as RPC namespace and implements the minimal component interface:

- `Illuminance.GetConfig` to obtain the component's configuration
- `Illuminance.SetConfig` to update the component's configuration
- `Illuminance.GetStatus` to obtain the component's status

Methods

Illuminance.SetConfig

Properties:

Property	Type	Description
<code>id</code>	<code>number</code>	Id of the Illuminance component instance
<code>config</code>	<code>object</code>	Configuration that the method takes

Find more about the config properties in [config section](#)

Illuminance.GetConfig

Property	Type	Description
<code>id</code>	<code>number</code>	Id of the Illuminance component instance

Find the `Illuminance.GetConfig` response properties in [config section](#)

Illuminance.GetStatus

Properties:

Property	Type	Description
<code>id</code>	<code>number</code>	Id of the Illuminance component instance

Find more about the status response properties in [status section](#)

Configuration

The configuration of the Illuminance component allows to adjust the `illumination` thresholds (`dark_thr` / `bright_thr`). To Get/Set the configuration of the Illuminance component its `id` must be specified.

Properties:

Property	Type	Description
<code>id</code>	<code>number</code>	Id of the Illuminance component instance
<code>name</code>	<code>string</code>	Name of the Illuminance instance. <code>name</code> length should not exceed 64 chars
<code>dark_thr</code>	<code>number</code>	Threshold in lux. Accepted range is device-specific

Property	Type	Description
bright_thr	number	bright threshold in lux. Accepted range is device-specific

Status

The status of the Illuminance component represents the measurement of the associated illuminance sensor. To obtain the status of the Illuminance component its `id` must be specified.

Properties:

Property	Type	Description
<code>id</code>	number	Id of the Illuminance component instance
<code>lux</code>	number or null	Illuminance in lux (<code>null</code> if valid value could not be obtained) (if applicable)
<code>illumination</code>	string or null	Illuminance level interpreted according to <code>dark_thr/bright_thr</code> (<code>null</code> if valid value could not be obtained): <code>lux</code> below <code>dark_thr</code> is interpreted as <code>dark</code> , <code>lux</code> between <code>dark_thr</code> and <code>bright_thr</code> is interpreted as <code>twilight</code> , <code>lux</code> above <code>bright_thr</code> is interpreted as <code>bright</code>
<code>errors</code>	array of type string	Shown only if at least one error is present. May contain <code>out_of_range</code> , <code>read</code> when there is problem reading sensor

Webhook Events

There is one event related to the Illuminance component that can trigger webhooks:

- `illuminance.change` - produced when `illumination` (dark/twilight/bright) has changed between two measurements

`illuminance.change` support two *attributes*, that can be used to compose conditional [webhooks](#):

- `lux`: number, new illuminance in lux (if applicable)
- `illumination`: string, new illuminance level interpreted according to `dark_thr/bright_thr`

Examples

Illuminance.SetConfig example

Illuminance.SetConfig HTTP

GET Request

Illuminance.SetConfig

Curl Request

Illuminance.SetConfig

Mos Request

```
http://192.168.33.1/rpc/Illuminance.SetConfig?id=0&config=
{"name": "Illuminance0"}
```

Illuminance.GetConfig example

Illuminance.GetConfig HTTP

GET Request

Illuminance.GetConfig

Curl Request

Illuminance.GetConfig

Mos Request

```
http://192.168.33.1/rpc/Illuminance.GetConfig?id=0
```

Response

```
{
  "id": 0,
  "name": null,
  "dark_thr": 200,
  "bright_thr": 300
}
```

Illuminance.GetStatus example

[Illuminance.GetStatus HTTP](#)

[GET Request](#)

[Illuminance.GetStatus](#)

[Curl Request](#)

[Illuminance.GetStatus](#)

[Mos Request](#)

```
http://192.168.33.1/rpc/Illuminance.GetStatus?id=0
```

Response

```
{  
  "id": 0,  
  "lux": 350,  
  "illumination": "bright",  
  "errors": [  
    "out_of_range"  
  ]  
}
```