



Version: 1.0

Flood

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

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

- `Flood.GetConfig` to obtain the component's [configuration](#)
- `Flood.SetConfig` to update the component's [configuration](#)
- `Flood.GetStatus` to obtain the component's [status](#)

Methods

Flood.SetConfig

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

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

Flood.GetConfig

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

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

Flood.GetStatus

The status of the Flood component represents the alarm and mute state of the associated flood sensor. To obtain the status of the Flood component its `id` must be specified.

Properties:

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

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

Configuration

The configuration of the Flood component allows to adjust the Flood `alarm_mode`. To Get/Set the configuration of the Flood component its `id` must be specified.

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the Flood component instance
<code>name</code>	<i>string or null</i>	Name of the Flood instance. <code>name</code> length should not exceed 64 chars
<code>alarm_mode</code>	<i>string or null</i>	Configuration of sound made on alarm. Accepted values are <code>disabled</code> , <code>normal</code> , <code>intense</code> and <code>rain</code> . When cable is unplugged value is <code>null</code>
<code>report_holddoff</code>	<i>number</i>	Time waited before reporting flood status. For this period status must be steady. Accepted values are between <code>0</code> - <code>60</code>

Property	Type	Description
		sec with step <code>5 sec</code>

Status

The status of the Flood component represents the alarm and mute state of the associated flood sensor. To obtain the status of the Flood component its `id` must be specified.

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the Flood component instance
<code>alarm</code>	<i>boolean</i>	Alarm state
<code>mute</code>	<i>boolean</i>	Mute state
<code>errors</code>	<i>array of type string</i>	Error conditions occurred. May contain <code>cable_unplugged</code> , (shown if at least one error is present)

Webhook Events

There are two events related to the Flood component that can trigger webhooks:

- `flood.alarm` - produced when flood alarm is triggered
- `flood.alarm_off` - produced when flood alarm goes off
- `flood.cable_unplugged` - produced when cable for flood detection is unplugged

Examples

Flood.SetConfig example

Flood.SetConfig HTTP GET Request

Flood.SetConfig Curl Request

Flood.SetConfig Mos Request

```
http://192.168.33.1/rpc/Flood.SetConfig?id=0&config={"name":"Flood0"}
```

Response

```
{
  "restart_required": false
}
```

Flood.GetConfig example

Flood.GetConfig HTTP GET Request

Flood.GetConfig Curl Request

Flood.GetConfig Mos Request

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

Response

```
{
  "id": 0,
  "name": null,
  "alarm_mode": "normal"
}
```

Flood.GetStatus example

Flood.GetStatus HTTP GET Request

Flood.GetStatus Curl Request

Flood.GetStatus Mos Request

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

Response

```
{  
  "id": 0,  
  "alarm": true,  
  "mute": false  
}
```