



Version: 1.0

# Smoke

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

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

- `Smoke.GetConfig` to obtain the component's [configuration](#)
- `Smoke.SetConfig` to update the component's [configuration](#)
- `Smoke.GetStatus` to obtain the component's [status](#)
- `Smoke.Mute` to mute alarm of the associated smoke sensor

## Methods

### Smoke.SetConfig

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

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

### Smoke.GetConfig

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

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

## Smoke.GetStatus

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

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

## Smoke.Mute

This method mute alarm of the associated smoke sensor.

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

## Configuration

To Get/Set the configuration of the Smoke component its `id` must be specified.

Properties:

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

Property	Type	Description
<code>name</code>	<i>string or null</i>	Name of the Smoke instance. <code>name</code> length should not exceed 64 chars

## Status

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

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the Smoke component instance
<code>alarm</code>	<i>boolean</i>	Alarm state
<code>mute</code>	<i>boolean</i>	Mute state

## Webhook Events

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

- `smoke.alarm` - produced when smoke alarm is triggered
- `smoke.alarm_off` - produced when smoke alarm goes off
- `smoke.alarm_test` - produced when alarm test is invoked

## Examples

### Smoke.SetConfig example

### Smoke.SetConfig HTTP GET Request

---

### Smoke.SetConfig Curl Request

### Smoke.SetConfig Mos Request

```
http://192.168.33.1/rpc/Smoke.SetConfig?id=0&config={"name":"Smoke0"}
```

#### Response

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

## Smoke.GetConfig example

### Smoke.GetConfig HTTP GET Request

---

### Smoke.GetConfig Curl Request

### Smoke.GetConfig Mos Request

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

#### Response

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

## Smoke.GetStatus example

### Smoke.GetStatus HTTP GET Request

---

### Smoke.GetStatus Curl Request

### Smoke.GetStatus Mos Request

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

#### Response

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

## Smoke.Mute example

### Smoke.Mute HTTP GET Request

---

### Smoke.Mute Curl Request

### Smoke.Mute Mos Request

```
http://192.168.33.1/rpc/Smoke.Mute?id=0
```

### Response

```
null
```