

cloud_devices module

cloud_devices module description

Module provides an access to a cloud device's functionality.

cloud_devices module API (`require "cloud_devices"`)

`subscribe(script_name[, events_filter])`

Subscribe a script for cloud_devices **events**, optionally specifying **event filter rules**. After subscribing the script will be launched for each event that happens on a cloud_devices add-on and information about this event will be passed as a parameter.

call: `cloud_devices.subscribe()`

params: script name (string), events_filter (table)

return: subscriber_id (string).

example:

```
require "cloud_devices"
```

```
cloud_devices.subscribe( "HUB:cloud_devices/scripts/events_handling" )
```

`set_subscription_filters(subscriber_id, filter_rules)`

Change events filter rules for a subscription `subscriber_id`. (**set_subscription_filters**)

params:

subscriber_id: string

filter_rules: table (array)

return:

nil

example:

```
require "cloud_devices"
```

```
cloud_devices.subscribe("HUB:cloud_devices/scripts/events_handling")
cloud_devices.set_subscription_filters("HUB:cloud_devices/scripts/events_handling", { { event =
"node_added" } } )
```

unsubscribe()

Unsubscribe a script from cloud_devices events.

call: cloud_devices.unsubscribe()

params: subscriber_id (string).

return: none.

example:

```
require "cloud_devices"
```

```
cloud_devices.unsubscribe( "HUB:cloud_devices/scripts/events_handling" )
```

get_device()

Get cloud device by its device id. (device id is a part of data coming to a script subscribed for cloud devices events)

call: cloud_devices.get_device()

params: device_id (string).

return: cloud device (table). Look at the objects page to get device table format.

example:

```
require "cloud_devices"
```

```
local device = cloud_devices.get_node( device_id )
if device then do
  print( "device.reachable: " .. device.reachable )
end
```

send_command()

Send command to a specified cloud device capability.

call: `cloud_devices.send_command()`

params: `command_id` (string), `cloud_device_id` (string), `capability` (string), `command` (string), `command_params` (table).

return: none.

example:

```
require "cloud_devices"
```

```
local status, error_msg = pcall(cloud_devices.send_command, command_id, cloud_device_id,  
"text_command", "send", { text = "turn on the light" })  
if not status then  
    print("Error sending command: " .. error_msg)  
end
```

Capability	Supported commands	Supported parameters
text_command	send	text (string)