

JANET Controller REST API

0.8.3

OAS3

REST API for interfacing with a JANET Controller node in the JANET Home Simulation System

Servers

TBD - TBD

Sublocations API



PUT

/sublocation/{subloc_id} Add a new sublocation



Adds a new empty sublocation in the controller's location

Parameters

Try it out

Name

Description

subloc_id * required

integer

(path)

The sublocation ID within the location (int > 0, because the default sublocation {loc_id,0} cannot be added)

4

Request body

application/json



- **name** → The sublocation name (string, optional)

Example Value | Schema

```
{
  "name": "kitchen"
}
```

Responses

| Code | Description | Links |
|------|--|----------|
| 201 | CREATED (sublocation successfully added) | No links |
| 400 | BAD REQUEST <ul style="list-style-type: none"> Parameter subloc_id is not an integer Parameter subloc_id is < 1 <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Parameter "subloc_id" is not an integer ("ciao")</div> | No links |
| 405 | METHOD NOT ALLOWED (unsupported HTTP method) <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Method "GET" is not allowed in the specified path</div> | No links |
| 409 | CONFLICT (sublocation {loc_id,subloc_id} already exists) <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> A sublocation with such "sub_id" ({1,2}) already exists</div> | No links |
| 415 | UNSUPPORTED MEDIA TYPE (request body could not be interpreted as JSON) <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Request body could not be interpreted in JSON format</div> | No links |

| Code | Description | Links |
|------|--|----------|
| 500 | <div>INTERNAL SERVER ERROR</div> <div><ul style="list-style-type: none">Unknown JSONE parameter mappingUnhandled server error</div> <div>Media type</div> <div><div>text/plain</div><div>▼</div></div> <div>Example Value Schema</div> <div><UNKNOWN SERVER ERROR> Unknown error: {errorReason}</div> | No links |

| | | |
|--|---|------------------|
| DELETE | /sublocation/{subloc_id} Delete a sublocation | ^ |
| <div>Deletes an existing sublocation from the controller's location, also moving all its devices into the default sublocation {loc_id,0}</div> | | |
| Parameters | | Try it out |
| <div>Name</div> | <div>Description</div> | |
| <div>subloc_id * required</div> <div>integer</div> <div>(path)</div> | <div>The sublocation ID within the location (int > 0, because the default sublocation {loc_id,0} cannot be deleted)</div> <div>4</div> | |
| Responses | | |
| <div>Code</div> | <div>Description</div> | <div>Links</div> |
| 204 | NO CONTENT (sublocation successfully deleted) | No links |

| Code | Description | Links |
|------|---|----------|
| 400 | <p>BAD REQUEST</p> <ul style="list-style-type: none"> Parameter subloc_id is not an integer Parameter subloc_id is < 1 <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> Parameter "subloc_id" is not an integer ("ciao")</pre> | No links |
| 404 | <p>NOT FOUND (sublocation {loc_id,subloc_id} was not found in the JANET Simulator)</p> <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> A sublocation with such "sub_id" ({95,32}) does not exist</pre> | No links |
| 405 | <p>METHOD NOT ALLOWED (unsupported HTTP method)</p> <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> Method "GET" is not allowed in the specified path</pre> | No links |
| 500 | <p>INTERNAL SERVER ERROR</p> <ul style="list-style-type: none"> Unhandled server error <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><UNKNOWN SERVER ERROR> Unknown error: {errorReason}</pre> | No links |

Devices API



PUT

/device/{dev_id} Add a new device



Adds a new device with a default initial configuration in a given sublocation of the controller's location, and starts its JANET Device node

Parameters

Try it out

Name

Description

dev_id * required

The device ID (unique int > 0)

integer

(path)

5

Request body

application/json



- **subloc_id** → The ID of the sublocation in the location where to put the device (int ≥ 0)
- **name** → The device name (string, optional)
- **type** → The device type ("Fan" | "Light" | "Door" | "Thermostat" | "Conditioner")
- **hostname** → The hostname where to deploy the device node (string, required)

Example Value | Schema

```
{
  "subloc_id": 4,
  "name": "Pinguino Cool&Quiet",
  "type": "Conditioner",
  "hostname": "www.somePredefinedHost.com"
}
```

Responses

Code

Description

Links

| Code | Description | Links |
|------|--|----------|
| 200 | <p>OK (device was successfully added)</p> <p>The dev_id and the complete initial state of the device are returned in the response body, where the full list of device states can be found in the <i>devstate.*</i> schemas</p> <p>Media type</p> <div>application/json</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "dev_id": 5, "state": { "onOff": "off", "fanSpeed": 50 } }</pre> | No links |
| 400 | <p>BAD REQUEST</p> <ul style="list-style-type: none"> Parameter dev_id is not an integer Parameter dev_id is < 1 Parameter subloc_id is missing Parameter subloc_id is not an integer Parameter subloc_id is < 0 Parameter type is missing Parameter type is not a valid device type <p>Media type</p> <div>text/plain</div> <p>Example Value Schema</p> <pre><ERROR> The specified device "type" (fAn) is invalid</pre> | No links |
| 404 | <p>NOT FOUND (sublocation {loc_id,subloc_id} was not found in the JANET Simulator)</p> <p>Media type</p> <div>text/plain</div> <p>Example Value Schema</p> <pre><ERROR> A sublocation with such "sub_id" ({95,32}) does not exist</pre> | No links |

| Code | Description | Links |
|------|--|----------|
| 405 | <p>METHOD NOT ALLOWED (unsupported HTTP method)</p> <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> Method "GET" is not allowed in the specified path</pre> | No links |
| 406 | <p>NOT ACCEPTABLE (The hostname does not belong to the list of allowed hosts)</p> <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> The specified "hostname" (pippo.com) does not belong to the list of allowed hosts where JANET nodes can be deployed in</pre> | No links |
| 409 | <p>CONFLICT (device dev_id already exists)</p> <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> A device with such "dev_id" (3) already exists</pre> | No links |
| 415 | <p>UNSUPPORTED MEDIA TYPE (request body could not be interpreted as JSON)</p> <p>Media type</p> <div> text/plain </div> <p>Example Value Schema</p> <pre><ERROR> Request body could not be interpreted in JSON format</pre> | No links |

| Code | Description | Links |
|------|---|----------|
| 500 | <div>INTERNAL SERVER ERROR</div> <div><ul style="list-style-type: none">The device was added, but an internal error occurred in starting its node {error:NodeStartError}Unknown JSONE parameter mappingUnhandled server error</div> <div>Media type<div>text/plain</div></div> <div>Example Value Schema<div><SERVER ERROR> The device of "dev_id" 5 was added, but an internal error occurred in starting its node: {error,NodeStartError}</div></div> | No links |

POST

/device/{dev_id} Update a device's sublocation

Moves an existing device in another existing sublocation in the controller's location

Parameters

Try it out

| Name | Description |
|---|---|
| <div>dev_id * required</div> <div>integer</div> <div>(path)</div> | <div>The device ID (unique int > 0)</div> <div>5</div> |

Request body required

application/json

- subloc_id → The ID of the sublocation where to move the device in (int ≥ 0)

Example Value | Schema

```
{  "subloc_id": 4}
```

Responses

| Code | Description | Links |
|------|---|----------|
| 204 | NO CONTENT (device sublocation successfully updated) | No links |
| 400 | BAD REQUEST <ul style="list-style-type: none"> Parameter dev_id is not an integer Parameter dev_id is < 1 Parameter subloc_id is missing Parameter subloc_id is not an integer Parameter subloc_id is < 0 <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Parameter "subloc_id" is not an integer ("ciao")</div> | No links |
| 404 | NOT FOUND <ul style="list-style-type: none"> Sublocation {loc_id,subloc_id} was not found in the JANET Simulator Device dev_id was not found in the controller's location <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> A sublocation with such "sub_id" ({95,32}) does not exist</div> | No links |
| 405 | METHOD NOT ALLOWED (unsupported HTTP method) <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Method "GET" is not allowed in the specified path</div> | No links |

| Code | Description | Links |
|------|--|----------|
| 415 | UNSUPPORTED MEDIA TYPE (request body could not be interpreted as JSON) Media type <div>text/plain</div> Example Value Schema <div><ERROR> Request body could not be interpreted in JSON format</div> | No links |
| 500 | INTERNAL SERVER ERROR <ul style="list-style-type: none">Unknown JSONE parameter mappingUnhandled server error Media type <div>text/plain</div> Example Value Schema <div><UNKNOWN SERVER ERROR> Unknown error: {errorReason}</div> | No links |

| | | |
|--|--|------------|
| DELETE | /device/{dev_id} Delete a device | ^ |
| Deletes an existing device from the controller's location, also stopping its device node | | |
| Parameters | | Try it out |
| Name | Description | |
| dev_id * required integer (path) | The device ID (unique int > 0) <div>5</div> | |
| Responses | | |
| | | |

| Code | Description | Links |
|------|--|----------|
| 204 | NO CONTENT (device successfully deleted) | No links |
| 400 | BAD REQUEST <ul style="list-style-type: none"> Parameter dev_id is not an integer Parameter dev_id is < 1 <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Parameter "dev_id" is not an integer ("ciao")</div> | No links |
| 404 | NOT FOUND (device dev_id was not found in the controller's location) <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> A device with such "dev_id" (165) does not exist</div> | No links |
| 405 | METHOD NOT ALLOWED (unsupported HTTP method) <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><ERROR> Method "GET" is not allowed in the specified path</div> | No links |
| 500 | INTERNAL SERVER ERROR <ul style="list-style-type: none"> The device was deleted, but an internal error occurred in stopping its node Unhandled server error <div>Media type</div> <div>text/plain ▼</div> <div>Example Value Schema</div> <div><SERVER ERROR> The device "dev_id" was deleted, but an internal error occurred in stopping its node: {error,Reason}</div> | No links |

Devices Commands API



PATCH

/devcommands Issue device commands



Issue one or more commands to the devices in the controller's location

Request Body

The request body must consist in a list of "N" **Device Commands (DevCommands)** structured as follows:

```
[
  // Device Command 1
  {
    "dev_id": 5, // Device the command refers to
    "actions":  // Set of device-specific actions
    {          // (in this example, for a fan)
      "onOff": "on",
      "fanSpeed": 50
    }
  },

  // Device Command 2
  {
    "dev_id": 6,
    "actions":  // Note that not every device-specific action
    {          // is required in the "actions" object
      "tempTarget": 25
    }
  },

  :
  // Device Command N
  { ... }
]
```

Where only one command towards the same "dev_id" can be issued, and the list of device-specific actions with their allowed values can be found in the *actions.** schemas.

HTTP Response Format

Response Body

The response body consists in a list of "N" **Device Responses (DevResponses)**, each associated with a device command (but not necessarily in the same order) and that can be **Successful** or **Unsuccessful**:

Successful Device Response

```
{
  "dev_id": 5,    // Device (and so command) the response refers to
  "status": 200,  // 200 -> Successful response
  "updatedState": // Device traits that have changed or, possibly
  {              // the entire device state (see below)
    "onOff": "on",
    "tempTarget": 25,
    "tempCurrent": 32
  },
  "timestamp": "2021-10-06T02:49:35" // Device timestamp
}
```

Where the "updatedState" object may consist in:

- If the command was issued **within a predefined maximum time from the last update** the controller received from the device, in **the traits that have changed from such update** (possibly *none*, leading to an empty object "updatedState":{})
- If the command was issued **after a predefined maximum time from the last update** the controller received from the device, in **the complete device state** (which in this case cannot be empty)

Unsuccessful Device Response

```
{
  "dev_id": 6,    // Device (and so command) the response refers to
  "status": 406,  // not 200 -> Unsuccessful response (307-504)
  "errorReason": "The command was rejected by the device for it is invalid in
  its current state"
  // A textual description of why the command was unsuccessful
}
```

Where the schema of the "state" object is analogous to the one of the "actions" objects for the device commands, and the full list of errors that may be returned can be found in the *errorReason.** schemas.

Response Status Code

Apart from corner cases where the request could not be interpreted correctly, the HTTP status code of the response is set to:

- 200 (OK) -> ALL device commands were successful (all device responses are successful)
- 202 (ACCEPTED) -> There are both successful and unsuccessful commands
- *maxErr* -> The maximum status code among all unsuccessful device responses (if no command was successful)

Parameters

[Try it out](#)

No parameters

Request body required

application/json

Example Value | Schema

```
[
  {
    "dev_id": 5,
    "actions": {
      "onOff": "on",
      "fanSpeed": 30
    }
  }
]
```

Responses

Code

Description

Links

200

OK (ALL device commands were successful)

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "dev_id": 5,
    "status": 200,
    "devstate": {
      "onOff": "off",
      "fanSpeed": 50
    },
    "timestamp": "2021-10-06T02:49:35"
  }
]
```

Code**Description****Links**

202

ACCEPTED (There are both successful and unsuccessful responses)

No links

Media type

application/json



Example Value | Schema

```
[
  {
    "dev_id": 5,
    "status": 200,
    "devstate": {
      "onOff": "off",
      "fanSpeed": 50
    },
    "timestamp": "2021-10-06T02:49:35"
  },
  {
    "dev_id": 5,
    "status": 400,
    "errorReason": "A valid command towards such device was already issued"
  }
]
```

400

BAD REQUEST

No links

- The request body could not be interpreted as a JSON list
- The request body is empty

Media type

text/plain



Example Value | Schema

```
<ERROR> The request body could not be interpreted as a JSON list
```

405

METHOD NOT ALLOWED (unsupported HTTP method)

No links

Media type

text/plain



Example Value | Schema

```
<ERROR> Method "GET" is not allowed in the specified path
```

| Code | Description | Links |
|------|--|----------|
| 415 | <p>UNSUPPORTED MEDIA TYPE (request body could not be interpreted as JSON)</p> <p>Media type</p> <div>text/plain ▼</div> <p>Example Value Schema</p> <pre><ERROR> Request body could not be interpreted in JSON format</pre> | No links |
| 500 | <p>INTERNAL SERVER ERROR</p> <ul style="list-style-type: none"> The body of the HTTP response could not be encoded in JSON Unknown JSONE parameter mapping Unhandled server error <p>Media type</p> <div>text/plain ▼</div> <p>Example Value Schema</p> <pre><SERVER ERROR> The device responses could not be encoded in JSON</pre> | No links |
| 3XX | <p>(all commands were unsuccessful, the one with the highest code is "3XX")</p> <p>Media type</p> <div>application/json ▼</div> <p>Example Value Schema</p> <pre>[{ "dev_id": 5, "status": 400, "errorReason": "A valid command towards such device was already issued" }]</pre> | No links |

| Code | Description | Links |
|------|--|----------|
| 4XX | <p>(all commands were unsuccessful, the one with the highest code is "4XX")</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>[{ "dev_id": 5, "status": 400, "errorReason": "A valid command towards such device was already issued" }]</pre> | No links |
| 5XX | <p>(all commands were unsuccessful, the one with the highest code is "5XX")</p> <p>Media type</p> <div>application/json</div> <p>Example Value Schema</p> <pre>[{ "dev_id": 5, "status": 400, "errorReason": "A valid command towards such device was already issued" }]</pre> | No links |

| Schemas | |
|--|--|
| <div><div>dev_id integer</div><div>minimum: 1</div><div>example: 5</div></div> | |

devcommand ▾ {

dev_id

dev_id integer

minimum: 1

example: 5

actions

actions > {...}

example: OrderedMap { "onOff": "on", "fanSpeed": 30 }

}

actions ▾ {

oneOf ->

actions.fan > {...}

minProperties: 1

maxProperties: 2

actions.light > {...}

minProperties: 1

maxProperties: 3

actions.door > {...}

minProperties: 1

maxProperties: 2

actions.thermostat > {...}

minProperties: 1

maxProperties: 2

actions.conditioner > {...}

minProperties: 1

maxProperties: 3

}

example: OrderedMap { "onOff": "on", "fanSpeed": 30 }

actions.fan ▾ {

anyOf ->

trait.onOff string

example: off

Enum:

▾ [on, off]

trait.fanSpeed integer

minimum: 1

maximum: 100

example: 50

}

minProperties: 1

maxProperties: 2

actions.light ∨ {



```
anyOf ->          trait.onOff string
                   example: off
                   Enum:
                     ∨ [ on, off ]
                   trait.brightness integer
                   minimum: 1
                   maximum: 100
                   example: 50
                   trait.color string
                   example: #FFFFFF
}
minProperties: 1
maxProperties: 3
```

actions.door ∨ {



```
anyOf ->          trait.openClose string
                   example: close
                   Enum:
                     ∨ [ open, close ]
                   trait.lockUnlock string
                   example: unlock
                   Enum:
                     ∨ [ lock, unlock ]
}
minProperties: 1
maxProperties: 2
```

actions.thermostat ∨ {



```
anyOf ->          trait.onOff string
                   example: off
                   Enum:
                     ∨ [ on, off ]
                   trait.tempTarget integer
                   minimum: 0
                   maximum: 50
                   example: 21
}
minProperties: 1
maxProperties: 2
```

actions.conditioner ▾ {

anyOf -> trait.onOff **string**
example: off
Enum:
 ▾ [on, off]
 trait.tempTarget **integer**
 minimum: 0
 maximum: 50
 example: 21
 trait.fanSpeed **integer**
 minimum: 1
 maximum: 100
 example: 50

}

minProperties: 1
maxProperties: 3

action.onOff **string**

example: off
Enum:
 ▾ [on, off]

action.fanSpeed **integer**

minimum: 1
maximum: 100
example: 50

action.brightness **integer**

minimum: 1
maximum: 100
example: 50

action.color **string**

example: #FFFFFF

action.openClose **string**

example: close
Enum:
 ▾ [open, close]

action.lockUnlock string

example: unlock

Enum:

▼ [lock, unlock]



action.tempTarget integer

minimum: 0

maximum: 50

example: 21



devresponse_successful ▼ {

| | |
|------------------|---|
| dev_id | dev_id integer <i>minimum: 1</i> <i>example: 5</i> |
| status | integer <i>minimum: 200</i> <i>maximum: 200</i> <i>example: 200</i> |
| devstate | devstate > {...} |
| timestamp | string <i>example: 2021-10-06T02:49:35</i> |

}



devresponse_unsuccessful ▼ {

| | |
|--------------------|--|
| dev_id | dev_id integer <i>minimum: 1</i> <i>example: 5</i> |
| status | integer <i>minimum: 307</i> <i>maximum: 504</i> <i>example: 400</i> |
| errorReason | errorReason > {...} <i>example: A valid command towards such device was already issued</i> |

}



devstate ▾ {

oneOf ->

devstate.fan > {...}
devstate.light > {...}
devstate.door > {...}
devstate.thermostat > {...}
devstate.conditioner > {...}

}

devstate.fan ▾ {

onOff*

trait.onOff **string**
example: off
Enum:

▾ [on, off]

fanSpeed*

trait.fanSpeed **integer**
minimum: 1
maximum: 100
example: 50

}

devstate.light ▾ {

onOff*

trait.onOff **string**
example: off
Enum:

▾ [on, off]

brightness*

trait.brightness **integer**
minimum: 1
maximum: 100
example: 50

color

trait.color **string**
example: #FFFFFF

}

devstate.door ▾ {

openClose* trait.openClose **string**
 example: close
 Enum:
 ▾ [open, close]

lockUnlock* trait.lockUnlock **string**
 example: unlock
 Enum:
 ▾ [lock, unlock]

}

devstate.thermostat ▾ {

onOff* trait.onOff **string**
 example: off
 Enum:
 ▾ [on, off]

tempTarget* trait.tempTarget **integer**
 minimum: 0
 maximum: 50
 example: 21

tempCurrent* trait.tempCurrent **integer**
 example: 21

}

devstate.conditioner ▾ {

onOff* trait.onOff **string**
 example: off
 Enum:
 ▾ [on, off]

tempTarget* trait.tempTarget **integer**
 minimum: 0
 maximum: 50
 example: 21

tempCurrent* trait.tempCurrent **integer**
 example: 21

fanSpeed* trait.fanSpeed **integer**
 minimum: 1
 maximum: 100
 example: 50

}

errorReason ▾ {

oneOf -> errorReason.missingparam_devid **string**
 Enum:
 ▾ [Required parameter 'dev_id' is missing]

```

errorReason.notaninteger_dev_id  string
Enum:

    ✓ [ Parameter 'dev_id' is not an integer ]
errorReason.outofrange_dev_id  string
Enum:

    ✓ [ Invalid value of parameter 'dev_id' (0) ]
errorReason.duplicate_dev_id  string
Enum:

    ✓ [ A valid command towards such device was already issued ]
errorReason.missingparam_actions  string
Enum:

    ✓ [ The 'actions' object is missing ]
errorReason.notamap_actions  string
Enum:

    ✓ [ The 'actions' parameter could not be interpreted as a
JSON object ]
errorReason.empty_actions  string
Enum:

    ✓ [ The 'actions' object is empty ]
errorReason.device_not_exists  string
Enum:

    ✓ [ A device with such 'dev_id' does not exist ]
errorReason.device_offline  string
Enum:

    ✓ [ The device is currently offline ]
errorReason.invalid_action  string
Enum:

    ✓ [ The 'actions' object contains invalid actions for the
associated device type (fan) ]
errorReason.invalid_state_door  string
Enum:

    ✓ [ The state {'open','lock'} is invalid for device type
'door' ]
errorReason.unknown_jsone_cast  string
Enum:

    ✓ [ Invalid value of trait 'fanSpeed' was cast to an unknown
term() by the JSONE library ]
errorReason.invalid_db_devtype  string
Enum:

    ✓ [ The device type in the controller database (fAn) is
invalid ]
errorReason.unhandled_parsing  string
Enum:

    ✓ [ Unhandler server error while parsing the command:
{error,Reason} ]
errorReason.cmdclient_crash  string
Enum:

    ✓ [ Server error in issuing the device commands:
{cmdclient_crash} ]
errorReason.unhandled_issuing_general  string
Enum:

    ✓ [ Server error in issuing the device commands:
{error,Reason} ]
errorReason.invalid_devconfig  string
Enum:

```



```

    ✓ [ The command was rejected by the device for it is invalid
in its current state ]
errorReason.dev_timeout string
Enum:

    ✓ [ The device node is not responding ]
errorReason.statem_timeout string
Enum:

    ✓ [ The device state machine is not responding ]
errorReason.unexpected_response string
Enum:

    ✓ [ An unexpected device command response was received:
{UnexpectedResponse} ]
errorReason.missing_response string
Enum:

    ✓ [ No command response was returned from the device ]
errorReason.invalid_devtype string
Enum:

    ✓ [ The updated state returned by the device is malformed ]
errorReason.unhandled_issuing string
Enum:

    ✓ [ Unhandled server error in parsing the command response:
{error,Reason} ]
}

```

example: A valid command towards such device was already issued

errorReason.missingparam_devid string

Enum:

- ✓ [Required parameter 'dev_id' is missing]

errorReason.notaninteger_devid string

Enum:

- ✓ [Parameter 'dev_id' is not an integer]

errorReason.outofrange_devid string

Enum:

- ✓ [Invalid value of parameter 'dev_id' (0)]

errorReason.duplicate_dev_id string

Enum:

- ✓ [A valid command towards such device was already issued]

errorReason.missingparam_actions `string`



Enum:

- ▼ [The 'actions' object is missing]

errorReason.notamap_actions `string`



Enum:

- ▼ [The 'actions' parameter could not be interpreted as a JSON object]

errorReason.empty_actions `string`



Enum:

- ▼ [The 'actions' object is empty]

errorReason.device_not_exists `string`



Enum:

- ▼ [A device with such 'dev_id' does not exist]

errorReason.device_offline `string`



Enum:

- ▼ [The device is currently offline]

errorReason.invalid_action `string`



Enum:

- ▼ [The 'actions' object contains invalid actions for the associated device type (fan)]

errorReason.invalid_state_door `string`



Enum:

- ▼ [The state {'open','lock'} is invalid for device type 'door']

errorReason.unknown_jsone_cast `string`



Enum:

- ▼ [Invalid value of trait 'fanSpeed' was cast to an unknown term() by the JSONE library]

errorReason.invalid_db_devtype `string`



Enum:

- ▼ [The device type in the controller database (fAn) is invalid]

errorReason.unhandled_parsing `string`



Enum:

- ▼ [Unhandler server error while parsing the command: {error,Reason}]

errorReason.cmdclient_crash `string`



Enum:

- ▼ [Server error in issuing the device commands: {cmdclient_crash}]

errorReason.unhandled_issuing_general `string`



Enum:

- ▼ [Server error in issuing the device commands: {error,Reason}]

errorReason.invalid_devconfig `string`



Enum:

- ▼ [The command was rejected by the device for it is invalid in its current state]

errorReason.dev_timeout `string`



Enum:

- ▼ [The device node is not responding]

errorReason.statem_timeout `string`



Enum:

- ▼ [The device state machine is not responding]

errorReason.unexpected_response `string`



Enum:

- ▼ [An unexpected device command response was received: {UnexpectedResponse}]

errorReason.missing_response `string`



Enum:

- ▼ [No command response was returned from the device]

errorReason.invalid_devtype `string`



Enum:

- ▼ [The updated state returned by the device is malformed]

errorReason.unhandled_issuing `string`



Enum:

- ▼ [Unhandled server error in parsing the command response: {error,Reason}]

trait.onOff `string`



example: off

Enum:

- ▼ [on, off]

trait.fanSpeed `integer`



minimum: 1

maximum: 100

example: 50

trait.brightness `integer`



minimum: 1

maximum: 100

example: 50

trait.color `string`



example: #FFFFFF

trait.openClose `string`



example: close

Enum:

- ▼ [open, close]

trait.lockUnlock `string`



example: unlock

Enum:

▼ [lock, unlock]

trait.tempTarget `integer`



minimum: 0

maximum: 50

example: 21

trait.tempCurrent `integer`



example: 21