

Hardware Monitor v1.1

Copyright © Riverbed Technology Inc. 2024

Created Jan 16, 2024 at 02:01 PM

Resource: chassis

A single chassis

http://{device}/api/npm.hardware_monitor/1.1/chassis/items/{serial_number}

JSON

```
{
  "availability": string,
  "columns": integer,
  "headunit": boolean,
  "health": string,
  "model": string,
  "rows": integer,
  "sensor_status": {
    "cpus": [ cpu ],
    "exhaust_air_sensors": [ exhaust_air_sensor ],
    "intake_air_sensors": [ intake_air_sensor ],
    "power_supplies": [ power_supply ],
    "raid_controllers": [ raid_controller ],
    "storage_unit_controllers": [ storage_unit_controller ],
    "system_fans": [ system_fan ]
  },
  "serial_number": string,
  "unused_slots": [
    [
      integer
    ]
  ]
}
```

| Property Name | Type | Description | Notes |
|--|--------------------------------------|--|---|
| <i>chassis</i> | <object> | A single chassis | |
| <i>availability</i> | <string> | Chassis availability | Read-only; Values: ok, missing, foreign, invalid; |
| <i>chassis.columns</i> | <integer> | The number of columns of disk slots on the chassis. '0' is at the left. | Read-only; Optional; Minimum 0; |
| <i>chassis.headunit</i> | <boolean> | Indicates whether this is the head unit chassis. | Read-only; Optional; |
| <i>health</i> | <string> | Health status | Read-only; Values: ok, notice, failing, failed; |
| <i>chassis.model</i> | <string> | Chassis model | Read-only; Optional; |
| <i>chassis.rows</i> | <integer> | The number of rows of disk slots on the chassis. '0' is at the top. | Read-only; Optional; Minimum 0; |
| <i>chassis.sensor_status</i> | <object> | Status of the sensors on the chassis. Available sensors varies based on chassis model. | Optional; |
| <i>chassis.sensor_status.cpus</i> | <array of <cpu>> | List of CPU sensors | Optional; |
| <i>chassis.sensor_status.cpus[items]</i> | <cpu> | CPU sensor data | |
| <i>chassis.sensor_status.exhaust_air_sensors</i> | <array of <exhaust_air_sensor>> | List of chassis air exhaust sensors | Optional; |
| <i>chassis.sensor_status.exhaust_air_sensors[items]</i> | <exhaust_air_sensor> | Chassis air exhaust sensor data | |
| <i>chassis.sensor_status.intake_air_sensors</i> | <array of <intake_air_sensor>> | List of chassis air intake sensors | Optional; |
| <i>chassis.sensor_status.intake_air_sensors[items]</i> | <intake_air_sensor> | Chassis air intake sensor data | |
| <i>chassis.sensor_status.power_supplies</i> | <array of <power_supply>> | List of power supply sensors | Optional; |
| <i>chassis.sensor_status.power_supplies[items]</i> | <power_supply> | Power supply sensor data | |
| <i>chassis.sensor_status.raid_controllers</i> | <array of <raid_controller>> | List of RAID controller sensors | Optional; |
| <i>chassis.sensor_status.raid_controllers[items]</i> | <raid_controller> | RAID controller sensor data | |
| <i>chassis.sensor_status.storage_unit_controllers</i> | <array of <storage_unit_controller>> | List of storage unit controller sensors | Optional; |
| <i>chassis.sensor_status.storage_unit_controllers[items]</i> | <storage_unit_controller> | Storage unit controller sensor data | |
| <i>chassis.sensor_status.system_fans</i> | <array of <system_fan>> | List of system fan sensors | Optional; |

| | | | |
|---|--|--|------------|
| <code>chassis.sensor_status.system_fans[items]</code> | <code><system_fan></code> | System fan sensor data | |
| <code>serial_number</code> | <code><string></code> | Chassis serial number | Read-only; |
| <code>chassis.unused_slots</code> | <code><array of <array of <integer>>></code> | A list of positions in the disk grid that are unused | Optional; |
| <code>chassis.unused_slots[items]</code> | <code><array of <integer>></code> | A list of [row, column] pairs representing an unused disk location | |
| <code>chassis.unused_slots[items][items]</code> | <code><integer></code> | A row or column index | |

Links

chassis: get

```
GET http://{device}/api/npm.hardware_monitor/1.1/chassis/items/{serial_number}
```

Response Body

Returns a [chassis](#) data object.

Relations

chassis: disks

Related resource

[disks](#)

Variables

| Related var | Data value for replacement |
|-------------|----------------------------|
| chassis | 0/serial_number |

chassis: instances

Related resource

[chassis_list](#)

Resource: chassis_list

All chassis known to the system

```
http://{device}/api/npm.hardware_monitor/1.1/chassis{?availability}
```

JSON

```
{
  "items": [ chassis ]
}
```

| Property Name | Type | Description | Notes |
|--|---|---------------------------------|-----------|
| <code>chassis_list</code> | <code><object></code> | All chassis known to the system | |
| <code>chassis_list.items</code> | <code><array of <chassis>></code> | List of chassis | Optional; |
| <code>chassis_list.items[items]</code> | <code><chassis></code> | A single chassis | |

Links

chassis_list: get

```
GET http://{device}/api/npm.hardware_monitor/1.1/chassis{?availability}
```

Response Body

Returns a [chassis_list](#) data object.

Resource: disk

A single disk

```
http://{device}/api/npm.hardware_monitor/1.1/disks/items/{id}
```

JSON

```
{
  "blink": boolean,
  "chassis": string,
  "device_type": string,
  "disk_type": string,
  "health": string,
  "id": string,
  "location": {
    "column": integer,
    "label": string,
    "row": integer
  },
  "model": string,
  "serial_number": string,
  "size_mb": integer,
  "status": string
}
```

| Property Name | Type | Description | Notes |
|---------------------------|-----------|--|--|
| <i>disk</i> | <object> | A single disk | Required properties: [id, blink]; |
| <i>disk.blink</i> | <boolean> | Set to true to blink the drive for identification | |
| <i>chassis</i> | <string> | Chassis identifier | Read-only; |
| <i>disk.device_type</i> | <string> | The type of device this is | Read-only; Optional; Values: disk; |
| <i>disk.disk_type</i> | <string> | Disk technology | Read-only; Optional; |
| <i>health</i> | <string> | Health status | Read-only; Values: ok, notice, failing, failed; |
| <i>disk.id</i> | <string> | A unique identifier for this disk's slot on this system. | Read-only; |
| <i>location</i> | <object> | The location of a disk in a chassis | Read-only; |
| <i>location.column</i> | <integer> | Disk column | Read-only; Optional; Minimum 0; |
| <i>location.label</i> | <string> | Disk label | Read-only; Optional; |
| <i>location.row</i> | <integer> | Disk row | Read-only; Optional; Minimum 0; |
| <i>disk.model</i> | <string> | Model string of the disk | Read-only; Optional; |
| <i>disk.serial_number</i> | <string> | Unique identifier of the disk | Read-only; Optional; |
| <i>disk.size_mb</i> | <integer> | Size of the disk in mebibytes | Read-only; Optional; |
| <i>status</i> | <string> | Status of a disk | Read-only; Values: ok, failed, failing, rebuilding, missing, invalid, new; |

Links

disk: get

```
GET http://{device}/api/npm.hardware_monitor/1.1/disks/items/{id}
```

Response Body

Returns a [disk](#) data object.

disk: set

```
PUT http://{device}/api/npm.hardware_monitor/1.1/disks/items/{id}
```

Request Body

Provide a [disk](#) data object.

Response Body

Returns a [disk](#) data object.

Relations

disk: chassis

All chassis in the system

Related resource

[chassis](#)

Variables

| Related var | Data value for replacement |
|---------------|----------------------------|
| serial_number | 0/chassis |

disk: instances

All disks in the system

Related resource

[disks](#)

Resource: disks

All disks known to the system

`http://{device}/api/npm.hardware_monitor/1.1/disks{?status,serial_number,chassis,disk_type}`

JSON

```
{
  "items": [ disk ]
}
```

| Property Name | Type | Description | Notes |
|---------------------------------|--|-------------------------------|-----------|
| <code>disks</code> | <code><object></code> | All disks known to the system | |
| <code>disks.items</code> | <code><array of <disk>></code> | List of disks | Optional; |
| <code>disks.items[items]</code> | <code><disk></code> | A single disk | |

Links

disks: get

GET `http://{device}/api/npm.hardware_monitor/1.1/disks{?status,serial_number,chassis,disk_type}`

Response Body

Returns a [disks](#) data object.

Resource: hardware_health

Overall hardware health

`http://{device}/api/npm.hardware_monitor/1.1/status`

JSON

```
{
  "status": string
}
```

| Property Name | Type | Description | Notes |
|------------------------------|-----------------------------|-------------------------|---|
| <code>hardware_health</code> | <code><object></code> | Overall hardware health | |
| <code>status</code> | <code><string></code> | Health status | Read-only; Values: ok, notice, failing, failed; |

Links

hardware_health: get

```
GET http://{device}/api/npm.hardware_monitor/1.1/status
```

Response Body

Returns a [hardware_health](#) data object.

hardware_health: refresh

Force a refresh of hardware status

```
POST http://{device}/api/npm.hardware_monitor/1.1/status/refresh
```

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

Type: chassis_availability

Chassis availability

JSON

```
string
```

| Property Name | Type | Description | Notes |
|-----------------------------|-----------------------|----------------------|--|
| <i>chassis_availability</i> | <i><string></i> | Chassis availability | Values: ok, missing, foreign, invalid; |

Type: cpu

CPU sensor data

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

| Property Name | Type | Description | Notes |
|------------------------|-----------------------------------|-------------------------|---|
| <i>cpu</i> | <i><object></i> | CPU sensor data | Required properties: [status, name, status_msg, temperature]; |
| <i>cpu.name</i> | <i><string></i> | The CPU name | |
| <i>cpu.status</i> | <i><sensor_status></i> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <i>cpu.status_msg</i> | <i><string></i> | The CPU status | |
| <i>cpu.temperature</i> | <i><sensor_temperature></i> | Temperature sensor data | |

Type: device_id

A generic device identifier

JSON

```
string
```

| Property Name | Type | Description | Notes |
|------------------|----------|-----------------------------|-------|
| <i>device_id</i> | <string> | A generic device identifier | |

Type: device_type

The type of a device

JSON

```
string
```

| Property Name | Type | Description | Notes |
|--------------------|----------|----------------------|------------------------|
| <i>device_type</i> | <string> | The type of a device | Values: disk, chassis; |

Type: disk_location

The location of a disk in a chassis

JSON

```
{
  "column": integer,
  "label": string,
  "row": integer
}
```

| Property Name | Type | Description | Notes |
|-----------------------------|-----------|-------------------------------------|----------------------|
| <i>disk_location</i> | <object> | The location of a disk in a chassis | |
| <i>disk_location.column</i> | <integer> | Disk column | Optional; Minimum 0; |
| <i>disk_location.label</i> | <string> | Disk label | Optional; |
| <i>disk_location.row</i> | <integer> | Disk row | Optional; Minimum 0; |

Type: disk_status

Status of a disk

JSON

```
string
```

| Property Name | Type | Description | Notes |
|--------------------|----------|------------------|---|
| <i>disk_status</i> | <string> | Status of a disk | Values: ok, failed, failing, rebuilding, missing, invalid, new; |

Type: exhaust_air_sensor

Chassis air exhaust sensor data

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

| Property Name | Type | Description | Notes |
|--------------------------------|----------|---------------------------------|---|
| <i>exhaust_air_sensor</i> | <object> | Chassis air exhaust sensor data | Required properties: [status, name, status_msg, temperature]; |
| <i>exhaust_air_sensor.name</i> | <string> | Sensor name | |

| | | | |
|---|---|-------------------------|--|
| <code>exhaust_air_sensor.status</code> | <code><sensor_status></code> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <code>exhaust_air_sensor.status_msg</code> | <code><string></code> | Sensor status | |
| <code>exhaust_air_sensor.temperature</code> | <code><sensor_temperature></code> | Temperature sensor data | |

Type: health_status

Health status

JSON

```
string
```

| Property Name | Type | Description | Notes |
|----------------------------|-----------------------------|---------------|--------------------------------------|
| <code>health_status</code> | <code><string></code> | Health status | Values: ok, notice, failing, failed; |

Type: intake_air_sensor

Chassis air intake sensor data

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

| Property Name | Type | Description | Notes |
|--|---|--------------------------------|---|
| <code>intake_air_sensor</code> | <code><object></code> | Chassis air intake sensor data | Required properties: [status, name, status_msg, temperature]; |
| <code>intake_air_sensor.name</code> | <code><string></code> | Sensor name | |
| <code>intake_air_sensor.status</code> | <code><sensor_status></code> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <code>intake_air_sensor.status_msg</code> | <code><string></code> | Sensor status | |
| <code>intake_air_sensor.temperature</code> | <code><sensor_temperature></code> | Temperature sensor data | |

Type: power_supply

Power supply sensor data

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

| Property Name | Type | Description | Notes |
|---------------------------------------|---|--------------------------|---|
| <code>power_supply</code> | <code><object></code> | Power supply sensor data | Required properties: [status, name, status_msg, temperature]; |
| <code>power_supply.name</code> | <code><string></code> | The power supply name | |
| <code>power_supply.status</code> | <code><sensor_status></code> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <code>power_supply.status_msg</code> | <code><string></code> | The power supply status | |
| <code>power_supply.temperature</code> | <code><sensor_temperature></code> | Temperature sensor data | |

Type: raid_controller

RAID controller sensor data

JSON

```
{
  "battery": string,
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

| Property Name | Type | Description | Notes |
|------------------------------------|----------------------|--|--|
| <i>raid_controller</i> | <object> | RAID controller sensor data | Required properties: [status, name, status_msg, battery, temperature]; |
| <i>battery</i> | <string> | The status of the controller's battery | Values: ok, missing, unknown, failing, failed; |
| <i>raid_controller.name</i> | <string> | RAID controller name | |
| <i>raid_controller.status</i> | <sensor_status> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <i>raid_controller.status_msg</i> | <string> | RAID controller status | |
| <i>raid_controller.temperature</i> | <sensor_temperature> | Temperature sensor data | |

Type: rpm

Revolutions per minute

JSON

```
integer
```

| Property Name | Type | Description | Notes |
|---------------|-----------|------------------------|-------|
| <i>rpm</i> | <integer> | Revolutions per minute | |

Type: sensor_status

The status of a sensor

JSON

```
string
```

| Property Name | Type | Description | Notes |
|----------------------|----------|------------------------|--|
| <i>sensor_status</i> | <string> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |

Type: sensor_temperature

Temperature sensor data

JSON

```
{
  "current_temperature": integer,
  "threshold": integer
}
```

| Property Name | Type | Description | Notes |
|---|-----------|----------------------------------|--|
| <i>sensor_temperature</i> | <object> | Temperature sensor data | Required properties: [current_temperature, threshold]; |
| <i>sensor_temperature.current_temperature</i> | <integer> | Current temperature in celsius | |
| <i>sensor_temperature.threshold</i> | <integer> | Threshold temperature in celsius | |

Type: storage_unit_controller

Storage unit controller sensor data

JSON

```
{
  "name": string,
  "status": sensor_status,
  "status_msg": string,
  "temperature": sensor_temperature
}
```

| Property Name | Type | Description | Notes |
|--|-----------------------------------|-------------------------------------|---|
| <i>storage_unit_controller</i> | <i><object></i> | Storage unit controller sensor data | Required properties: [status, name, status_msg, temperature]; |
| <i>storage_unit_controller.name</i> | <i><string></i> | Controller name | |
| <i>storage_unit_controller.status</i> | <i><sensor_status></i> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <i>storage_unit_controller.status_msg</i> | <i><string></i> | Controller status | |
| <i>storage_unit_controller.temperature</i> | <i><sensor_temperature></i> | Temperature sensor data | |

Type: system_fan

System fan sensor data

JSON

```
{
  "fan_speed": rpm,
  "name": string,
  "status": sensor_status,
  "status_msg": string
}
```

| Property Name | Type | Description | Notes |
|------------------------------|------------------------------|------------------------|---|
| <i>system_fan</i> | <i><object></i> | System fan sensor data | Required properties: [status, name, status_msg, fan_speed]; |
| <i>system_fan.fan_speed</i> | <i><rpm></i> | Revolutions per minute | |
| <i>system_fan.name</i> | <i><string></i> | The fan name | |
| <i>system_fan.status</i> | <i><sensor_status></i> | The status of a sensor | Values: ok, missing, unknown, failing, failed; |
| <i>system_fan.status_msg</i> | <i><string></i> | The fan status | |