REST API

Access

The RECS®|Box Management API is accessible via the IP-Address or the hostname of the TOR-Master of the cluster. The basic URL of the API has the format `http://TOR-Master/REST/`

Components

The RECS®|Box Management API makes available all hardware available hardware components in software. Right now the following components are supported by the API:

Attribute	Description
node	A single node
baseboard	A baseboard can be equipped with one x86 CPU module or four ARM CPU modules
backplane	A backplane can be equipped up to 6 baseboards for either COM Express or Apalis
rcu	A RECS®|Box Computing Unit (RCU) can be equipped with up to 18 baseboards
rack	A rack consists of several RCUs

Information about all these components are returned as XML code.

Node

The single attributes have the following meaning:

Attribute	Description
id	unique ID for referencing of the component
actualPowerUsage	actual power consumption of a node at the time of the last measurement in Watt
maxPowerUsage	maximum power usage that a node can draw
baseBoardId	ID of the baseboard which hosts the node
baseBoardPosition	position of the node on the baseboard
state	status of the node (on, off, suspend to RAM, suspend to disk)
architecture	Architecture (x86, x86_32, x86_64, arm)
temperature	temperature of the node in °C
health	health status of the node (OK, Warning, Critical)
inletTemperature	temperature of the inlet air in °C
outletTemperature	temperature of the outlet air in °C
voltage	supply voltage in Volt
description	description of the component

In accordance to the component node the API offers nodeVector which returns multiple instances of node.

Backplane

The single attributes have the following meaning:

Attribute	Description
id	Unique ID for referencing of the component
rcuId	Unique ID of the RECS®|Box Computing Unit which hosts the backplane
infrastructurePower	Power usage of the infrastructure components of the backplane in Watt
temperatures	List of temperatures measured on the backplane

In accordance to the component baseboard the API offers backplaneVector which returns multiple instances of backplane.

Baseboard

The single attributes have the following meaning:

Attribute	Description
id	Unique ID for referencing of the component
rcuId	Unique ID of the RECS®|Box Computing Unit which hosts the baseboard
rcuPosition	Position of the baseboard inside the RECS®|Box Computing Unit
infrastructurePower	Power usage of the infrastructure components of the baseboard in Watt
baseboardTyoe	Type of the baseboard (CXP, APLS)
description	description of the component
nodeId	List of IDs of the nodes installed on the baseboard

In accordance to the component baseboard the API offers baseboardVector which returns multiple instances of baseboard.

RCU

The single attributes have the following meaning:

Attribute	Description
id	unique ID for referencing of the component
rackId	ID of the rack which hosts the RECS®|Box Computing Unit
rackPosition	Position of the RECS®|Box Computing Unit in the rack
name	Name of the RECS®|Box Computing Unit
rcuType	Type of the RECS®|Box Computing Unit (Sirus, Arneb, Antares)
kvmNode	ID of the node to which the KVM is switched
description	description of the component
baseBoardId	List of IDs of baseboards which are installed in the RECS®|Box Computing Unit

In accordance to the component rcu the API offers rcuVector which returns multiple instances of rcu.

Rack

The single attributes have the following meaning:

Attribute	Description
`id`	unique ID for referencing of the component
`description`	description of the component
`rcuId`	List of IDs of RECS®|Box Computing Unit s which are installed in the rack

In accordance to the component rack the API offers rackVector which returns multiple instances of rack.

Resources

The resources are split into monitoring resources (for pure information gathering) and management resources (for changing the system configuration or state).

Monitoring

For monitoring the following resources are available:

Attribute	Description
`/node`	Returns a nodeVector with all nodes of the cluster
`/node/{node_id}`	Returns information about the node with the given ID as node
`/baseboard`	Returns a baseboardVector with all baseboards of the cluster
`/baseboard/{baseboard_id}`	Returns information about the baseboard with the given ID as baseboard
`/baseboard/{baseboard_id}/node`	Returns a nodeVector with all nodes that are installed on the baseboard with the given ID
`/backplane`	Returns a backplaneVector with all baseboards of the backplane
`/backplane/{backplane_id}`	Returns information about the backplane with the given ID
`/rcu`	Returns a rcuVector with all RECS®|Box Computing Units of the cluster
`/rcu/{rcu_id}`	Returns information about RECS®|Box Computing Unit with the given ID as rcu
`/rcu/{rcu_id}/baseboard`	Returns a baseboardVector with all baseboards that are installed in the RECS®|Box Computing Unit with the given ID
`/rcu/{rcu_id}/backplane`	Returns a backplaneVector with all backplanes that are installed in the RECS®|Box Computing Unit with the given ID
`/rack`	Returns a rackVector with all racks of the cluster
`/rack/{rack_id}`	Returns information about the rack with the given ID as rack
`/rack/{rack_id}/rcu`	Returns a rcuVector with all RECS®|Box Computing Units that are installed in the rack with the given ID

Management

The management of individual components can be found beneath the subaddress manage. Right now only management functionalities for nodes are implemented:

Attribute	Description
`/node/{node_id}/manage/power_on`	Turns on the node with the given ID and returns actualised information about the node as node
`/node/{node_id}/manage/power_off`	Turns off the node with the given ID and returns actualised information about the node as node
`/node/{node_id}/manage/reset`	Resets the node with the given ID and returns actualised information about the node as node
`/node/{node_id}/manage/select_kvm`	Switches the KVM port of the suitable RECS®|Box Computing Unit to the node with the given ID and returns actualised information about the node as node
`/rcu/{rcu_id}/manage/set_fans?percent={value}`	Sets the overall fan speed of the RCU with the given ID and returns the curent status of the RCU as rcu

Errors

Information about the success or failure of management requests are returned via HTTP status codes. Please have a look at RFC2616 for an overview about the defined HTTP status codes.