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 `https://TOR-Master/REST/` or `http://TOR-Master/REST/`.

Accessing the REST API requires HTTP Basic authentication. The authenticated user has to be in the “Admin” or “User” group to be able to execute the POST/PUT management calls.

Components

The RECS®|Box Management API makes all hardware components in the cluster available as XML trees 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 zero or more nodes
backplane	A backplane can be equipped with zero or more baseboards
rcu	A RECS®|Box Computing Unit (RCU) can be equipped with zero or more baseboards
rack	A rack consists of several RCUs

Node

Example XML:

```
<node baseBoardPosition="0" maxPowerUsage="44"
actualNodePowerUsage="32.426884399865166"
actualPEGPowerUsage="15.12053962324833" actualPowerUsage="47.54742402311349"
architecture="x86"
baseBoardId="RCU_84055620466592_BB_1" health="OK"
id="RCU_84055620466592_BB_1_0" inletTemperature="20.0"
lastSensorUpdate="1465470151268" macAddressCompute="70:b3:d5:56:40:48"
outletTemperature="20.0" state="1"
highestTemperature="20.0" voltage="12.072700851453936"/>
```

The following table shows the possible attributes (some are optional) and their meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
actualPowerUsage	Actual power consumption of a node (Node + PEG)	W	Double
actualNodePowerUsage	Actual power consumption of a node (Node only)	W	Double
actualPEGPowerUsage	Actual power consumption of a PEG card	W	Double

Attribute	Description	Unit	Data type
maxPowerUsage	Maximum power the node can draw	W	Integer
baseBoardId	ID of the baseboard which hosts the node	-	String
baseBoardPosition	Position of the node on the baseboard	-	Integer
state	Power state of the node (0=Off, 1=On, 2=Soft-off, 3=Standby, 4=Hibernate)	-	Integer
architecture	Architecture (x86, arm, UNKNOWN)	-	String
health	Health status of the node (OK, Warning, Critical)	-	String
inletTemperature	Temperature of the inlet air	°C	Double
outletTemperature	Temperature of the outlet air	°C	Double
highestTemperature	Highest temperature measured on the node's baseboard	°C	Double
voltage	Supply voltage of the baseboard	V	Double
lastSensorUpdate	Timestamp of the last sensor update	ms	Long
macAddressCompute	MAC address of the NIC connected to the compute network (optional)	-	String
macAddressMgmt	MAC address of the NIC connected to the management network (optional)	-	String

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
position	Unique ID for referencing of the component
id	Unique ID of 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	HTTP Method
/node	Returns a nodeVector with all nodes of the cluster	GET

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

Management

The management of individual components can be found under the “manage” path of the component.

Attribute	Description	HTTP method	Parameter
/node/{node_id}/manage/power_on	Turns on the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/power_off	Turns off the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/reset	Resets the node with the given ID and returns updated node XML	POST	

Attribute	Description	HTTP method	Parameter
`/node/{node_id}/manage/select_kvm`	Switches the KVM port of the RECS®|Box Computing Unit containing the node to the node with the given ID and returns updated node XML	PUT	
`/rcu/{rcu_id}/manage/set_fans`	Sets the overall fan speed of the RCU with the given ID and returns the curent status of the RCU	PUT	percent={value}

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.