

# 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. The following components are supported by the API:

Attribute	Description
node	A single node
backplane	A backplane can be equipped with zero or more baseboards
baseboard	A baseboard can be equipped with zero or more nodes
rcu	A RECS® Box Computing Unit (RCU) can be equipped with zero or more baseboards
rack	A rack consists of several RCUs

Many resources also return lists of components. These are named according to the scheme <component name>List (e.g. nodeList, rcuList) and contain the elements of the list.

Example of a backplaneList:

```
<backplaneList>
<backplane position="1" id="RCU_84055620466592_BP_1"
infrastructurePower="0.0">
<temperatures>24.0</temperatures>
<temperatures>25.0</temperatures>
<temperatures>26.0</temperatures>
<temperatures>27.0</temperatures>
<temperatures>28.0</temperatures>
</backplane>
</backplaneList>
```

## Node

Example XML:

```
<node baseBoardPosition="0" maxPowerUsage="44"
```

```
actualNodePowerUsage="32.426884399865166"  
actualPEGPowUsage="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
actualPEGPowUsage	Actual power consumption of a PEG card	W	Double
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 nodeList which returns multiple instances of node.

## Backplane

Example XML:

```
<backplane position="1" id="RCU_84055620466592_BP_1"  
infrastructurePower="0.0">  
<temperatures>24.0</temperatures>  
<temperatures>25.0</temperatures>  
<temperatures>26.0</temperatures>  
<temperatures>27.0</temperatures>  
<temperatures>28.0</temperatures>
```

```
</backplane>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
position	Position of the backplane in the RECS® Box Computing Unit	-	Integer
infrastructurePower	Power usage of the infrastructure components on the backplane	W	Double
temperatures	List of temperatures measured on the backplane	°C	Double

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

## Baseboard

Example XML:

```
<baseBoard rcuPosition="6" baseboardType="APLS" id="RCU_84055620466592_BB_6"
infrastructurePower="9.8" rcuId="RCU_84055620466592">
<nodeId>RCU_84055620466592_BB_6_1</nodeId>
<nodeId>RCU_84055620466592_BB_6_2</nodeId>
<nodeId>RCU_84055620466592_BB_6_3</nodeId>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
</baseBoard>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
rcuId	Unique ID of the RECS® Box Computing Unit hosting the baseboard	-	String
rcuPosition	Position of the baseboard inside the RECS® Box Computing Unit	-	Integer
infrastructurePower	Power usage of the infrastructure components on the baseboard	W	Double
baseboardType	Type of the baseboard (CXP, APLS)	-	String
nodeId	List of IDs of the nodes installed on the baseboard	-	String
temperatures	List of temperatures measured on the backplane	°C	Double

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

## RCU

Example XML:

```
<rcu rcuType="ANTARES" fanSpeed="60" rackId="RCK_1" name="RECSMaster (RCU)
on 192.168.56.195" rackPosition="0" id="RCU_84055620466592">
<backplaneId>RCU_84055620466592_BP_1</backplaneId>
<baseBoardId>RCU_84055620466592_BB_1</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_2</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_3</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_4</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_5</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_6</baseBoardId>
</rcu>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
rackId	ID of the rack which hosts the RECS® Box Computing Unit	-	String
rackPosition	Position of the RECS® Box Computing Unit in the rack	-	Integer
name	Name of the RECS® Box Computing Unit	-	String
rcuType	Type of the RECS® Box Computing Unit (SIRIUS, ARNEB, ANTARES)	-	String
kvmNode	ID of the node to which the KVM system is switched (optional)	-	String
fanSpeed	Current speed setting of the fans in the RECS® Box Computing Unit	%	Integer
backplaneId	List of IDs of backplanes which are installed in the RECS® Box Computing Unit	-	String
baseBoardId	List of IDs of baseboards which are installed in the RECS® Box Computing Unit	-	String

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

## Rack

Example XML:

```
<rack description="Default rack" id="RCK_1">
<rcuId>RCU_84055620466592</rcuId>
</rack>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
description	Description of the rack	-	String
rcuId	List of IDs of RECS® Box Computing Units which are installed in the rack	-	String

In accordance to the component rack the API offers rackList 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 nodeList with all nodes of the cluster	GET
/node/{node_id}	Returns information about the node with the given ID	GET
/baseboard	Returns a baseboardList with all baseboards of the cluster	GET
/baseboard/{baseboard_id}	Returns information about the baseboard with the given ID	GET
/baseboard/{baseboard_id}/node	Returns a nodeList with all nodes that are installed on the baseboard with the given ID	GET
/backplane	Returns a backplaneList with all backplanes of the cluster	GET
/backplane/{backplane_id}	Returns information about the backplane with the given ID	GET
/rcu	Returns an rcuList with all RECS® Box Computing Units of the cluster	GET
/rcu/{rcu_id}	Returns information about the RECS® Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/baseboard	Returns a baseboardList with all baseboards that are installed in the RECS® Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/backplane	Returns a backplaneList with all backplanes that are installed in the RECS® Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/node	Returns a nodeList with all nodes that are installed in the RECS® Box Computing Unit with the given ID	GET
/rack	Returns a rackList with all racks of the cluster	GET
/rack/{rack_id}	Returns information about the rack with the given ID	GET
/rack/{rack_id}/rcu	Returns a rcuList 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	
/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 current 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.

From:  
<https://recswiki.christmann.info/wiki/> - RECS®|Box Wiki

Permanent link:  
[https://recswiki.christmann.info/wiki/doku.php?id=documentation:rest\\_api&rev=1476096797](https://recswiki.christmann.info/wiki/doku.php?id=documentation:rest_api&rev=1476096797)

Last update: **2016/10/10 10:53**

