RACKCORP REST API

RACKCORP REST API

Rackcorp has migrated many of its functionalities to work with REST API architecture to modernize and make easy the process to get data through HTTP protocol. Every month we are adding new services through REST API. It is important you follow us to get the last updates and the last version of our API.

API Information:

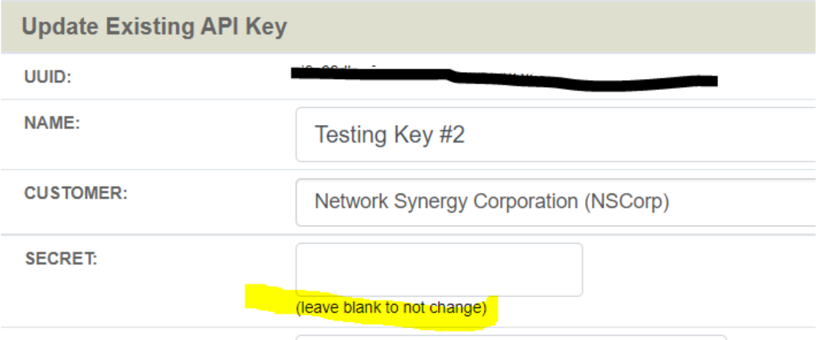
Current Version: v2.8

API URL: https://www.rackcorp.net/api/v2.8

Before you start to create any code or connect through our API, you need to create an API Key access which allows your code to make HTTPS requests to our services and be authorized to get data for your services. We will explain here how you add this API Key in your code.

To create API credentials, goto ADMINISTRATION -> API in our portal. URL:
https://portal.rackcorp.com/index.php?cmd=api

Then, you click ADD, type a **name** for this new Key and a **secret** (password) and SAVE.



Update Existing API Key

UUID:	[redacted]
NAME:	Testing Key #2
CUSTOMER:	Network Synergy Corporation (NSCorp)
SECRET:	(leave blank to not change)

> Make sure to record your SECRET phrase somewhere safe. It is required for API access and cannot be retrieved, It can only reset via the API portal key details page

API Standards:

As described in our article REST API Architecture and Standards, there is no definitive standard patterns that all engineers/devs should follow to create REST API for their app. In Rackcorp we keep things simple. The main structure for our REST API follows the patterns below:

version: v2.8

main URL: https://api.rackcorp.net/api

object data: customers | dc | network | api | dns | device

URL format: https://api.rackcorp.net/api/+version+/object data/+ID item

Example GET all data: https://api.rackcorp.net/api/v2.8/device

Example GET one Item: https://api.rackcorp.net/api/v2.8/device/1

Bellow is some example code which you can use as example to implement your first REST API connection to our platform:

> **We strongly recommend that you only make REST API calls from your back-end code. Be careful when using javascript code to call a REST API url. Never add your credentials in front-end code. Speak to our team if unsure.**

In this simple introduction to code REST API, we will use PHP and PYTHON programming language.

PHP:

In the sample bellow we add a dummy API KEY and API SECRET. Please, change this data accordingly with your data

```
<?php
// Simple example to get a list of all datacenters
$version = 'v2.8';
$url = "https://api.rackcorp.net/api/". $version . "/dc";
$query = ["cmd"=>"dc.getall"];
$query['APIUUID'] = "";  // No authetnication required for getting datacenter list
$query['APISECRET'] = "";



$curl = curl_init($url);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($query));
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

$response = curl_exec($curl);

if($response) {
	return json_decode($response, true);
}

return false;
?>
```

PYTHON:

```
import json
import logging
import sys
import os
import glob
import re
from bson import json_util
from flask.helpers import make_response
from flask import request, jsonify

version = 'v2.8'
apiurl = 'https://api.rackcorp.net/api'+version+'/'+dc
setheaders = {
        'content-type': 'application/json',
        'User-Agent': 'Mozilla',
        'jwt': jwt
}

data = {}
data['APIUUID'] = ''
data['APISECRET'] = ''

apiresp = None
apiresp = requests.get(self.apiurl+'/sessions/logout',data=self.rcdata,headers=setheaders)
```

As you can see in both examples, the logic to connect through our API is quite simple. You just need the URL, the APIUUID and APISECRET as part of the object or array that gets passed to the API.

Below you can find links with more advanced docs to use our API. Also, the complete list of REST API services (urls) explaining the query data and the expected response data for each situation.

REST API GibHub Docs:

Link: https://github.com/RackCorpCloud/rackcorp-api/wiki/RACKCORP-REST-API

Swagger RACKCORP REST API:

Swagger is a suite of API developer tools from SmartBear Software and a former specification upon which the OpenAPI Specification is based. This platform displays in simple layout all REST API functions allowing you visualize what your code should expect as response for each call (GET, PUT, PUSH, DELETE).

You can see through this platform the schema for each function and also, the expected JSON format response. You can also make tests to connect through your services in our servers using your real APIUUID and APISECRET.

It is good for testing staging environments and make sure that your call will receive exactly what your code expects.

> Dont forget to select which url path API you wanna use for tests. In Swagger page you can see in Server three options. The first one is a swagger mocking URL which is not for tests. The second one is the RACKCORP Production REST API Core (please be careful to use this URL). The third one is the RACKCORP Staging REST API which should be used for tests.

Link: https://app.swaggerhub.com/apis/RackCorp/Rackcorp-REST-API/2.8

RACKCORP REST API EXAMPLES

Reference material:

Swagger App: https://app.swaggerhub.com/apis-docs/RackCorp/Rackcorp-REST-API/2.8

Rackcorp REST API docs: https://wiki.rackcorp.com/books/help-and-support-en/page/rest-api-architecture-and-standards

PHP Code:

Simple server creation:

```
<?php

// IMPORTANT - as described in our documentation, Rackcorp follows the REST API standards
and each function must be requested with the correct METHOD (GET, POST, PUT, DELETE).
Pay attention to this detail when create your code to use CURL

function rackcorpAPI($action, $request) {
□□$URL = 'https://api.rackcorp.net/api/v2.8'+$action;
        $request["APIUUID"] = "";
        $request["APISECRET"] = "";

        $curl = curl_init($URL);
        curl_setopt($curl, CURLOPT_POST, true);
        curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($request));
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
        $response = curl_exec($curl);
        curl_close($curl);
        if ( !$response ) {
                return Array("code" => "FAULT", "message" => "API Error");
        }
        return json_decode($response, true);
}
```

```
$customerID = 1000;  // Change this to your customer ID (available in portal under ADMINISTRATION -> MY DETAILS)
// locations are defined here: https://wiki.rackcorp.com/books/help-and-support-en/page/rackcorp-datacenter-locations-and-codes

$neworder = "data": [
    "customerid": $customerID,
    "currency": "AUD",
    "servicebilltag": "CLOUDSERVER",
    "productdetails": [
      "BILLINGMODEL": "monthly",
      "NOINSTANCES": 1,
      "HOSTNAME": "Test Machine",
      "CLOUDTYPE": "public",
      "DCID": "89",
      "OS": "OS-ALMALINUX-16.1",
      "VMHID": "",
      "CPU": 2,
      "MEMORYGB": 4,
      "STORAGEGB": 20,
      "IPV6": 0,
      "IPV4": 1,
      "NT-SPEED": "NT-SPEED100",
      "TRAFFICGB": "TRAFFICGB-100",
      "BKP": "BKP-FREE",
      "SUPPORT": "SUPPORT-STD",
      "DDOS": "",
      "SECURITY":[]
    ]
  ];
// Lodge the order (this just locks pricing in for up to 72 hours but doesnt actually create any resources)
// IMPORTANT - method POST
$response = rackcorpAPI("/order/create/server", $neworder);
var_dump($response);

// You can look up the order if you want:
// IMPORTANT - method GET
$response = rackcorpAPI("/order/"+$response['data']["orderid"]);
var_dump($response);
```

```

// Then confirm the order to start provisioning:
// IMPORTANT - method GET
$response = rackcorpAPI("/order/confirm/server/"+$response['data']["orderid"]);
var_dump($response);

?>
```

Starting a server using cloud-init:

After creating a server, you can also choose to start it using cloud-init with your own custom code:

```
$cloudInitStartupData = Array(
        "cloudInit" => Array(
                "volumeName" => "config-2",
                "userData" => "#cloud-config
ssh_pwauth: True
users:
  - default
  - name: user1
    groups: sudo
    shell: /bin/bash
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    plain_text_passwd: testtest888
    lock_passwd: false
",
                "metaData" => "instance-id: ServerTest9999
local-hostname: MyServerHostname9999
"
        )
);
$serverIDToStart = 9999;
$tx = Array ("objId"=>$serverIDToStart, "objType"=>"DEVICE", "type"=>"STARTUP",
"data"=>json_encode($cloudInitStartupData));

// See earlier example for rackcorpAPI function
$response = rackcorpAPI("rctransaction.create", $tx);
var_dump($response);
?>
```