

# 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["APIUID"] = "";
    $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);

?>

```

---

Revision #6

Created 14 August 2022 11:42:08 by Stephen D

Updated 10 August 2023 06:24:43 by Otto