

# API Essentials

## API Essentials

### API Key Generation

### Autoscaling

- Add Autoscaling Rules
- Get List of Autoscaling Rules for VS
- Remove Autoscaling Rules

### Backups

- Add/Edit Note
- Convert Backup to Template
- Create Backup
- Delete Backup
- Get List of All VS Backups
- Get List of Incremental Backups
- Get List of Normal Backups
- Restore Disk from Backup

### Credit

- Get Credit

### DNS

- Add DNS Record
- Add DNS Zone
- Delete DNS Record
- Delete DNS Zone
- Edit DNS Record
- Get DNS Zones
- Get List of DNS Zone Records
- Get List of Name Servers

### Firewall Rules

- Add Firewall Rule
- Get Firewall Rules
- Update Firewall Rules

### Flex Cloud VM Control

### Getting Started with Superb Flex Cloud

### IP Addresses

- Get IP Address Joins

### Logs

- Get List of Log Items
- Get List of Transactions
- Get list of VS Transaction

### Network Interfaces

## Introduction

The Flex Cloud API is a powerful utility for Superb Flex Cloud users, with the API you can manage every aspect of your virtual machines and integrate that control with your existing frameworks or internal tools. This guide is a complete reference for all API calls, includes detailed API call information, code and output examples.

- The API is RESTful
- All function calls respond to XML and JSON exchange formats (JSON is the default)
- All function calls need authorization and authentication (Basic Auth using the API key as the password and the account ID as the username)

### API URL: <https://mycp.superb.net/flexapi>

<api\_url> is used as a placeholder for this url

The API accepts HTTP Basic Authentication with your [Account ID](#) as the user and your generated [API key](#) as the password. Curl example:

```
curl -u <account_id>:<api_key>
```

### Flex Cloud VM control

A simple Virtual Server management tool.

Check it out: [Flex Cloud VM Control](#)

### HTTP Methods

The API uses the following HTTP methods:

**GET** - used for retrieving information from a particular URI

**POST** - used for creating new object and adding new transactions into the queue

**PUT** - used for altering object properties, updated\_at value is changed in PUT requests even if the request fails.

**DELETE** - used for object deletion

### HTTP response codes

The API returns appropriate HTTP status codes for every request:

<b>200 OK</b>	The request completed successfully
<b>204 No content</b>	The request completed successfully. The 204 status is returned on DELETE and PUT requests
<b>207 Multi-Status</b>	Multiple requests made, multiple statuses returned

- Get VS Network Interfaces
- Rebuild VS Network

## Recipes

- Add Recipe
- Add Recipe Step
- Assign Recipe to Virtual Server
- Delete Recipe
- Delete Recipe Step
- Edit Recipe
- Edit Recipe Step
- Get All Recipes
- Get Recipe Steps
- Get Virtual Server Recipes
- Remove recipe from Virtual Server
- Run Recipe on Multiple Virtual Servers
- Swap Recipe Step Number

## SSH Keys

- Add SSH Key
- Delete SSH Key
- Edit SSH Key
- Get SSH Keys
- Set SSH Keys on VS

## Templates

- Get Templates

## Test Route

## Troubleshooting API Issues

## Viewing Activity Logs

## Virtual Server Operating Systems

## Virtual Servers

- Add Virtual Server
- Billing Statistics
- Build Virtual Server
- Delete Virtual Server
- Edit Virtual Server
- Get CPU Usage Statistics
- Get List of Virtual Machines
  - Get specific VM Details
  - Search Virtual Servers by label
- Get statuses for All VMs
  - Get Specific VS Status
- Reboot Virtual Server
- Reset VS Password
- Shutdown Virtual Server
- Startup a Virtual Server
- Stop Virtual Server

<b>201 Scheduled</b>	The request has been accepted and scheduled for processing
<b>403 Forbidden</b>	The request is correct, but could not be processed.
<b>404 Not Found</b>	The requested URL is incorrect or the resource does not exist. For example, if you request to delete a disk with ID {5}, but there is no such disk in the cloud, you will get a 404 error.
<b>422 Unprocessable Entity</b>	The sent parameters are erroneous.
<b>500 Internal Server Error</b>	An error occurred. Please contact support.

## Formatting and naming conventions

The table below represents all the existing formatting and naming conventions used in this guide:

Convention	Explanation	Example
user:userpass	stands for <i>username:password</i> (<account_id>:<api_key>)	723412:efd0d4b5995ac61adf11a2e193275bb5db1febd1
<api_url>	placeholder for the api address	curl -i -X GET --url <api_url>/virtual_machines
:id	stands for the resource ID. Sometimes also: :resource_id	23
<i>italics</i>	all the parameters are italicised	<i>currency_code</i>
* (asterisk)	marks the required parameters	<i>label</i> *
preformatted	indicates request examples in XML or JSON	GET /roles.xml
info	An info message emphasizes or explains the information within the chapter.	Clicking the OFF button performs graceful shutdown and then powers off the VS.
note	A note message contains information essential for the task completion.	The maximum length of a Mount Point is 256 characters.
warning	A warning message informs you of something you should not do or be cautious.	You won't be able to restore a VS after deleting it

## Templates

This is a reference for the available templates, *template\_label* may be used in place of the required *template\_id* when creating new virtual servers.

## VS Disks

- Add New Disk
- Delete Disk
- Edit Disk
- Get VS Disks

<template_label>	
Minimum memory	128
Minimum disk size	10

e.g.

```
{
  "virtual_machine": {
    "template_label": "CentOS 7.1 x64", ...
  }
}
```

Labels must be exact and are case sensitive!

## Available systems

- CentOS 6.7
- CentOS 7.7
- Debian 8.0
- FreeBSD 10.1
- Red Hat 6.7
- Red Hat 7.1
- Ubuntu 12.04
- Ubuntu 14.04
- Windows 2008 DC (datacenter)
- Windows 2008 STD (standard)
- Windows 2012 DC (datacenter)
- Windows 2012 STD (standard)

## CentOS 6.7

Label	CentOS 6.7 x64
Minimum memory	384 MB
Minimum disk size	5 GB
Virtualization type	XEN, KVM

## CentOS 7.7

Label	CentOS 7.1 x64
Minimum memory	384 MB
Minimum disk size	5 GB
Virtualization type	XEN, KVM

## Debian 8.0

Label	Debian 8.0 x64
Minimum memory	512 MB
Minimum disk size	6 GB
Virtualization type	XEN, KVM

## FreeBSD 10.1

Label	FreeBSD 10.1 x64
Minimum memory	256 MB
Minimum disk size	5 GB
Virtualization type	XEN, KVM

## Red Hat 6.7

Label	Red Hat Enterprise Linux 6.7 x64
Minimum memory	512MB
Minimum disk size	6 GB
Virtualization type	XEN, KVM

## Red Hat 7.1

Label	Red Hat Enterprise Linux 7.1 x64
Minimum memory	512MB
Minimum disk size	6 GB
Virtualization type	XEN, KVM

## Ubuntu 12.04

Label	Ubuntu 12.04 x64
Minimum memory	128 MB
Minimum disk size	5 GB
Virtualization type	XEN, KVM

## Ubuntu 14.04

Label	Ubuntu 14.04 x64
Minimum memory	256 MB

Minimum disk size	5 GB
Virtualization type	XEN, KVM

## Windows 2008 DC (datacenter)

Label	Windows 2008 x64 DC R2
Minimum memory	1024 MB
Minimum disk size	20 GB
Virtualization type	KVM
License	KMS

## Windows 2008 STD (standard)

Label	Windows 2008 x64 STD R2
Minimum memory	256 MB
Minimum disk size	5 GB
Virtualization type	KVM
License	KMS

## Windows 2012 DC (datacenter)

Label	Windows 2012 x64 DC R2
Minimum memory	1024 MB
Minimum disk size	20 GB
Virtualization type	KVM
License	KMS

## Windows 2012 STD (standard)

Label	Windows 2012 x64 STD R2
Minimum memory	1024 MB
Minimum disk size	20 GB
Virtualization type	KVM
License	KMS