



VSControl API Documentation

Copyright (c) 2009 by Devpond

Table of Contents

VSControl API	2
Formatting and using API calls	2
Functions	3
Vps List	3
Plans List	3
IPs List	3
OS List	3
Themes List	3
Change VPS Name/Hostname	4
Change Password	4
Create VPS	4
Pause Vps	4
Un Pause Vps	5
Start Vps	5
Stop Vps	5
Suspend Vps	5
UnSuspend Vps	5
Delete Vps	6
Migrate Vps	6
Apply Plan	6
Apply Disk	6
Apply Ram	7
Apply Bandwidth Limit	7
Add New IP Address in VSControl	7
Delete IP Address from VSControl	7
Assign IP Address to a VPS	8
Un Assign IP Address from a VPS	8
Synchronize IPs among VSControl and VPS	8
Make Backup of VPS data	8
Restore VPS from its Backup	9
Check the status of last action	9
List Free IPs available in VSControl	9
Code Examples for API Calls	10
Python Code	10
PHP Code	11

VSControl API

VSControl API allows you to issue commands to VSControl via your custom scripts. By using this API you can perform functions remotely, without accessing VSControl web user interface.

There are several possible applications of API. For example, you can manage many servers running VSControl easily by allowing you to issue a single command across multiple servers. You can use the API to integrate provisioning of end user's VPS from your billing or ordering system.

Formatting and using API calls

Each API call is formatted as a URL. To create a function call:

- Begin with the HTTPS address of your web server.
- Add `/vscontrol/functionname`, plus any variables required by the specific function you are calling. (For more information, see the description for each function.) For example, if your web server's hostname is `example.com`, you will include one of the following string to your script:
`https://example.com/vscontrol/functionname`
- You can also execute the VSControl's function by entering the URL string in your web browser's address bar with required fields as query string. This may be useful for testing the function call and viewing its output. For example, to use the `StartVps` function on a server whose hostname is `example.com`, you would include a string similar to the following in your script:
`https://example.com/vscontrol/StartVps/?username=uuuuuu&password=xxxxxx&vps_name=test.vpsname.com`

Sending logins detail over an unsecured connection is not recommended for security reasons. Therefore make sure you are using SSL connection by using "https" instead of "http".

You can see in the above example that we are passing the required input through query string however you can pass the required input through HTTP "POST" method as well. Python and PHP code examples are given at the end of this document.

Functions

Vps List

Function name - VpsList

This function lists all the VPS on the server. This function is usually used to get the VPS name for some other tasks.

Variables: The function requires no additional variable.

Plans List

Function name - PlansList

This function lists all the available plans in the VSControl. This function is usually used to get the name of the plan for use of some other task like change the plan for VPS.

Variables: The function requires no additional variable.

IPs List

Function name - IpsList

This function lists all the IPs on in VSControl both assigned and free.

Variables: The function requires no additional variable.

OS List

Function name - OSList

This function lists all the available OS templates on the server.

Variables: The function requires no additional variable.

Themes List

Function name - ThemesList

This function lists the available themes in VSControl.

Variables: The function requires no additional variable.

Change VPS Name/Hostname

Function name - ChangeName

This function is used to change the name/hostname of the VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Current name of the VPS
- **new_name (string)** – New name for the VPS.

Change Password

Function name - ChangePassword

This function is used to change the root password of the VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS to which new password should be applied.
- **new_password (string)** – New root password for the VPS.

Create VPS

Function name - CreateVps

This function is used to create new VPS on the server.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of new VPS
- **vps_password (string)** – Root password for this new VPS.
- **os (string)** – OS template name, from which VPS will be installed.
- **plan (string)** – Plan name for resource allocation.
- **primary_ip (IP address)** – First IP address.
- **secondary_ip (IP address)** – Second IP address.
- **ternary_ip (IP address)** – Third IP address.
- **email (string)** – Email id of the client.

Pause Vps

Function name - PauseVps

This function is used to pause the running VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS that needed to be paused.

Un Pause Vps

Function name - UnPauseVps

This function is used to un-pause the paused VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS that needed to be un-paused.

Start Vps

Function name - StartVps

This function is used to start the stopped VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS that need to be started.

Stop Vps

Function name - StopVps

This function is used to stop the running/paused VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – name of the VPS that need to be stopped.

Suspend Vps

Function name - SuspendVps

This function is used to suspend the VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – name of the VPS that need to be paused.

UnSuspend Vps

Function name - UnSuspendVps

This function is used to un-suspend the suspended VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – name of the VPS that need to be paused.

Delete Vps

Function name - DeleteVps

This function is used to delete or terminate the VPS from the node resulting all of its resources free.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – name of the VPS that need to be terminated.

Migrate Vps

Function name - MigrateVps

This function is used to migrate the VPS from one node to another.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS that need to be paused.
- **remote_host (string or IP address)** – Name or IP address of the remote server to which the VPS will be migrated.
- **remote_host_password (string)** – Password of the remote/target node.
- **vps_password (string)** – VPS password to set after migration.

Apply Plan

Function name - ApplyPlan

This function is used to change the plan for an existing VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS to apply new plan.
- **new_plan (string)** – Name of the plan to apply.

Apply Disk

Function name - ApplyDisk

This function is used to change the disk size for an existing VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS to apply new plan.
- **new_disk_size (string)** – New disk size in GB e.g 10

Apply Ram

Function name - ApplyRam

This function is used to change the RAM size for an existing VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS of which needed to change the RAM size.
- **new_ram_size (string)** – New RAM size in MB e.g 256

Apply Bandwidth Limit

Function name - ApplyTrafficLimit

This function is used to change the download limit for an existing VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – Name of the VPS of which bandwidth limit needed to be change.
- **new_limit (string)** – New limit in GB e.g 500

Add New IP Address in VSControl

Function name - AddIP

This function is used to add a new IP address in VSControl.

Variables: The function takes the following variables as an input:

- **ip_addr (string)** – IP address that will be added to VSControl.

Delete IP Address from VSControl

Function name - DeleteIP

This function is used to delete an existing IP address from VSControl.

Variables: The function takes the following variables as an input:

- **ip_addr (string)** – IP address to be deleted.

Assign IP Address to a VPS

Function name - AssignIP

This function is used to assign an IP address to the VPS. The IP address should exist in VSControl. If IP is new and does not exist in VSControl then add it first.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – VPS name to which IP address will be assigned.
- **ip_addr (string)** – IP address that will be assigned to VPS.

Un Assign IP Address from a VPS

Function name - UnAssignIP

This function is used to un-assign an IP address that is already assigned to VPS.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – VPS name from which IP will be unassigned.
- **ip_addr (string)** – IP that will be unassigned making it free or available.

Synchronize IPs among VSControl and VPS

Function name - SynchronizeIPs

This function is used to synchronize the IPs among VSControl and VPS. The IPs that are marked as assigned to a specific VPS in VSControl need to be checked whether these are actually assigned to that VPS or not.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – VPS name for which IPs will be synchronized.

Make Backup of VPS data

Function name - MakeBackup

This function is used to make/create backup of VPS data on the backup server. It is required that node should already be configured for backup feature.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – VPS name for which backup will be created.

Restore VPS from its Backup

Function name - RestoreVPS

This function is used to make/create and take backup of VPS data on the backup server. It is required that the node should already be configured for backup feature.

Variables: The function takes the following variables as an input:

- **vps_name (string)** – VPS name to be restored.
- **backup_name (string)** – backup id/name from which VPS will be restored.

Check the status of last action

Function name - ActionStatus

This function is used to check the status of last action whether it is in waiting, completed, or failed state. The API call will put the action in actions queue that will be executed by the vsdaemon. So using this function one can check the last action's status.

Variables: The function requires no variable.

List Free IPs available in VSControl

Function name - ListFreeIPs

This function is used to list the free IPs available in vscontrol.

Variables: The function requires no variable.

Code Examples for API Calls

Python Code

Api call for Free IPs Available.

[<https://example.com/vscontrol/ListFreeIPs/>]

Function Name for Free IPs Available: ListFreeIPs

```
import urllib,urllib2

#Please set following values accordingly
username = "uuuuu"          # vscontrol admin user
password = "xxxxx",        # admin password
hostname = "example.com"   # host name or IP without https:// where VSC is running
function_call = 'ListFreeIPs' # Api call for free ips available

url1 = 'https://' + hostname + '/vscontrol/'
values = {'username' : username,
          'password' : password
          }

opener = urllib2.build_opener(urllib2.HTTPCookieProcessor())
urllib2.install_opener(opener)

url=url1 + function_call + '/'
data = urllib.urlencode(values)
f = opener.open(url,data)
print f.read()
```

PHP Code

Api call for Plan List.

[<https://example.com/vscontrol/PlanList/>]

Function Name for Free IPs Available: PlanList

```
<?php

$username= 'uuuu';           //VSControl admin user
$password= 'xxxx';          //VSControl admin password
$hostname = 'example.com'; //host name or IP without http://
$function_call = 'PlanList'; //API Function call e.g. function call for Plan List is
'PlanList'

////////////////////////////////////

$values = 'username='.urlencode($username).'&password='.urlencode($password);
$url = 'https://'.$hostname.'/vscontrol/'.$function_call.'/';
echo do_post_request($url, $values);

function do_post_request($url, $data, $optional_headers = null)
{
    $params = array('http' => array(
                                                'method' => 'post',
                                                'content' => $data
                                            ));
    if ($optional_headers !== null) {
        $params['http']['header'] = $optional_headers;
    }
    $ctx = stream_context_create($params);
    $fp = @fopen($url, 'rb', false, $ctx);
    if (!$fp) {
        throw new Exception("Problem with $url, $php_errormsg");
    }

    $response = @stream_get_contents($fp);

    if ($response === false) {
        throw new Exception("Problem reading data from $url, $php_errormsg");
    }

    return $response;
} ?>
```