

# Add Recipe Step

## 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

## /recipes/:recipe\_id/recipe\_steps

Adds a step (script) to specified recipe

### Sections:

- Response Parameters
- JSON Request example
- XML Request example

### Request Parameters

*script* - recipe step code

*result\_source* - step result source:

- *exit\_code* - an exit status, e.g. 0 will be returned on success

To use exit code in the VBS or PowerShell scripts, you have to specify it directly in the script. For example:

```
VBS  
Script:  
WScript.Echo "test"  
WScript.Quit 95
```

```
PowerShell  
Script:  
get-date -displayhint date  
exit 227
```

- *std\_out* - standard output
- *std\_err* - standard error
- *std\_out\_and\_std\_err* - standard output and standard error

*pass\_anything\_else* - set true, if you have specified the recipe fail value, otherwise set false

*pass\_values* - step pass value

*on\_success* - step behavior in case of success:

- *proceed* - proceed to the next step.
- *fail* - terminate the recipe and mark it as failed.
- *stop* - terminate the recipe and mark it as successful.
- *go to step* - specify the step to proceed to. If you specify the nonexistent step, the recipe will be stopped.

*success\_goto\_step* - if the *on\_success* parameter = *go to step*, specify the step to proceed to. If you specify the nonexistent step, the recipe will be stopped.

*fail\_anything\_else* - set true, if you have specified the recipe pass value, otherwise set false

*fail\_values* - step fail value

*on\_failure* - step behavior in case of failure:

- *proceed* - proceed to the next step.
- *fail* - terminate the recipe and mark it as failed.
- *stop* - terminate the recipe and mark it as successful.
- *go to step* - specify the step to proceed to. If you specify the nonexistent step, the recipe will be stopped.

*failure\_goto\_step* - if the *on\_failure* parameter = *go to step*, specify the step to proceed to. If you specify the nonexistent step, the recipe will be stopped.

You can only specify behavior for one scenario: for example, if the *fail\_anything\_else* = f

- Get VS Network Interfaces
- Rebuild VS Network

**also, `pass_anything_else` must be set to `true`.**

## 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

## Response Parameters

201 status is returned on success

## JSON Request example

```
curl -i -X POST -u user:userpass
<api_url>/recipes/:recipe_id/recipe_steps.json
-d '{"recipe_step":{"script":"uptime
>/tmp/uptime.log",
"result_source":"exit_code", "pass_anything_else
":"0", "pass_values":"0", "on_success":"proceed",
"success_goto_step":"","fail_anything_else":"1"
, "fail_values":"","on_failure":"proceed",
"failure_goto_step":""}}' -H 'Accept:
application/json' -H 'Content-type:
application/json'
```

## XML Request example

```
curl -i -X POST -u user:userpass
<api_url>/recipes/:recipe_id/recipe_steps.xml
-d
'<recipe_step><script>uptime</script>/tmp/uptime.log
</script><result_source>exit_code</result_sourc
e><pass_anything_else>0</pass_anything_else><pa
ss_values>0</pass_values><on_success>proceed</o
n_success><success_goto_step></success_goto_ste
p><fail_anything_else>1</fail_anything_else><fa
il_values></fail_values><on_failure>proceed</on
_failure><failure_goto_step></failure_goto_step
></recipe_step>' -H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

## VS Disks

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