Parallels Operations Automation 5.5.4

PACI RESTful API Programmer's Guide

Revision 5.0 (May 16, 2014)

Parallels IP Holdings GmbH
Vordergasse 59
8200 Schaffhausen
Switzerland
Tel: + 41 52 632 0411
Fax: + 41 52 672 2010
www.parallels.com

Contents

CHAPTER 1

Introduction

This document is a programming and reference guide to the Parallels Automation for Cloud Infrastructure (PACI) RESTful API. The guide is intended for users who would like to write their own programs and scripts to automate management of their PACI resources.

CHAPTER 2

PACI REST API Basics

The PACI RESTful API provides programmatic access to REST resources. Using the API you can obtain information about the resources and perform actions on them. For example, you can obtain a list of the existing servers, start or stop a server, modify server configuration, create a new or delete an existing server, and perform many other management tasks. This chapter provides an overview of the API and describes the basics of using it in your programs.

In This Chapter

Accessing a Resource

A PACI resource is accessed by sending an HTTPS request to a PACI server. When the server receives a request, it processes it accordingly (preforms actions or retrieves data) and sends back a response that can contain the data that you requested, an operation status code, or an error message. All PACI resources are accessed at the following base URL (referred to as *baseURL* in the API reference topics of this guide):

```
https://{ip_address | hostname}:port/paci/version
```

where:

- `ip_address | hostname` is the PACI server IP address or hostname.
- `port` is the port number on which the server is listening for REST requests
- `paci` must be typed exactly as shown
- `version` is the API version number

Note: Ask your service provider for the values listed above (except the "paci" string) and then use them when composing the base URL in your code.

Base URL example

The following base URL sample contains the PACI server hostname ("paci-web"), port number (4465), the "paci" string, and the version number (v1.0).

```
https://paci-web:4465/paci/v1.0
```

Resource path and parameters

When accessing a particular resource, you append the following values to the base URL:

- The path that identifies the resource
- Optional resource-specific parameters (where available)

Each resource is identified by a path within the base URL. An individual resource may have subresources within its own path, each of which is also identified by a path. For example, the following sample URL references the `ve` resource, which is "server" in general:

```
https://paci-web:4465/paci/v1.0/ve
```

The following sample URL references a specific server identified by its name, which in this instance is my-server-01:

```
https://paci-web:4465/paci/v1.0/ve/my-server-01
```

Resource-specific parameters

Some requests allow to specify additional (usually optional) parameters. For example, when obtaining the list of the operating system templates, you may add a filter to retrieve the templates for a particular operating system (e.g. a particular Linux distribution). Additional parameters are included in the URL after the question mark (?).

HTTP Request Headers

When sending an HTTPS request, the request headers must contain authentication and content type information as described below.

Authentication

PACI REST API uses the basic authentication scheme as defined by RFC 1945 (Hypertext Transfer Protocol – HTTP/1.0). An HTTP request header must contain credentials in the form of user name and password separated by a colon and encoded using the Base64 encoding scheme. The following is an example of an authorization header:

```
Authorization: Basic dG9ib3RyYXM6cTE=
```

Please note that the password to be used in PACI REST API calls must be obtain from the POA customer's control panel as follows:

1. In the customer's control panel, open the **Cloud Infrastructure** tab.
2. Switch to the **API Access** tab and click the **Generate API Access Key** button.
3. Your key appears in a message box. Use this key as the password in your API calls.

Service providers can obtain the REST API access key using the provider's control panel. Please see the provider's guide for additional information.

Content Type

Input data is included in the PACI REST API request in the form of an XML document. Therefore, Content-type should be specified as "application/xml":

```
Content-type: application/xml
```

Callbacks

PACI REST API provides a callback functionality that allows you to receive responses from PACI server asynchronously. To use a callback, the request header must include the "x-callback-url:" string followed by the URL at which your HTTP server listens for callbacks from PACI server. For example:

```
x-callback-url: http://192.168.3.77:8081/bd6fsdb4-24sd-4a64-8fd1-403d1b11tf0a
```

For more information about using callbacks, please see **Callbacks** (p. 90).

CHAPTER 3

HTTP Methods

REST requests are sent to the PACI server as HTTP **POST**, **GET**, **PUT**, and **DELETE** messages. In many cases different methods can be used on the same resource to achieve different goals. For example, you can obtain the information about a server using the **GET** method, modify its configuration using the **PUT** method, and delete it using the **DELETE** method. The resource (identified by a path) will be the same in all three instances, but each request parameters and input/output data will be different. In this guide, each API reference topic describes a particular operation and provides information about the HTTP method that must be used.

Data Input and Output

When sending or receiving data, it is included in a request or received from the PACI server as an XML document. Each request that sends or receives data has a corresponding XML specification that defines the XML document structure. When composing a request, the specification must be followed precisely. The validation of the request is performed on the server side against the XML specification. If a mandatory parameter is missing or an invalid parameter is passed, you will receive an HTTP error. An XML specification for each request and response is described in individual request topics of this guide.

The sample XML document below contains a server information. A document like this is received when a server information is requested or is included in the request when a new server is created (or an existing server is modified).

```
<?xml version="1.0" encoding="UTF-8"?>
<ve>
  <name>my-server-01</name>
  <description>Test server</description>
  <cpu number="2" power="1600"/>
  <ram size="512"/>
  <disk local="true" size="1"/>
  <platform>
    <template-info name="ubuntu-9.10-x86_64"/>
  </platform>
  <backup-schedule name="daily"/>
</ve>
```

In general, the same XML specification is used in the API for input and output when dealing with the same resource type. Certain parameters, however, cannot be used in the input version and therefore must be omitted. The reference topics in this guide describe the input and output XML specifications separately.

Error Handling

If there's an error executing a PACI REST API request, an HTTP page containing the error information is returned to the caller. The page contains an HTTP code and the description of the error. HTTP error 406 usually means that there was an error in the request or the operation cannot be completed for other reasons. In this case, the error page will also contain a PACI REST API error code and a text describing the error. The following is an example of such code and description:

```
P80100: Template [centos-6-x86_64] is already registered at the node ...
```

The code in the example above (P80100) can be interpreted as follows:

- The first letter can be either P (permanent error) or T (transient error).
- The first digit after the letter means one of the following:
  - 1 — internal error
  - 2 — resource is in use
  - 3 — resource is not available
  - 4 — unsupported operation
  - 5 — invalid modification
  - 6 — the operation cannot be performed at this time
  - 7 — the request is missing required information
  - 8 — invalid data was passed in the request
  - 9 — invalid operation
- The rest of the number is formed by multiplying the first digit by 10000 and adding an actual error code to it. In the example above, the actual error code is 100, therefore: 8 * 10000 + 100 = 80100. For the complete list of the actual error codes, see **Appendix/Error Codes** (p. 93).

HTTP errors other than 406 could mean other, usually general, errors and will contain a proper description of the problem.

CHAPTER 4

Format and Conventions

The baseURL convention

The string "baseURL" is used in the API reference topics as a shorthand for the base URL at which the PACI resources can be accessed. When composing an HTTP request, the "baseURL" string must be substituted with the base URL specific to your PACI environment. See the **Accessing a Resource** section (p. 6) for more information about the base URL.

API reference topics

Each API reference topic in this guide provides information on how to compose an HTTP request that will perform a particular operation on a particular resource. Each entry contains the following information:

- An HTTP method (POST, GET, PUT, DELETE) used to access the resource. Depending on the type of the operation, different methods may be used to access the same resource. Each operation type has its own topic in this documentation.
- A full path to the resource in the form *baseURL/resource_path*.
- A description of the resource and the operation.
- A list of additional parameters that can be used with the request (where applicable).
- An XML specification of the input and/or output XML documents (included only with the requests that send and/or receive data). Use these specifications to compose an XML input and parse the XML output.
- Samples of HTTP request, HTTP response, and input/output XML documents.

To compose an HTTP request that will perform a particular task on a particular resource, find the corresponding reference topic (each topic name contains the short task description) and follow the provided instructions.

API reference format

Each topic describing an HTTP request has the following sections:

Description

Explains the purpose of the request.

Syntax

Specifies which HTTP method is used with the request (GET, POST, PUT, DELETE) and the resource URL. See **Accessing a Resource** (p. 6) for more info on the resource URL format.

Request Parameters

Describes the XML specification used to specify the request parameters.

Response

For requests that don't output data, describes the HTTP message returned. For requests that return data, describes the XML Schema of the output XML document.

Example

Provides samples of HTTP request, HTTP response, and XML input/output.

Testing code samples and creating your own programs

You can test the samples provided in this guide using a REST client for a Web browser. For example, you can use a simple but effective RESTClient extension for Firefox or any other available REST plug-in and a browser of your choice.

To write your own programs using the API, you will need a development tool that will allow you to make Web requests from the command line or from a program (e.g a program written in C or Python). One of the commonly used tools is cURL. With cURL you can use the API in a script or a C program.

CHAPTER 5

PACI REST API Reference

This chapter contains PACI REST API reference. The requests described here are intended for end users, who want to use the API to manage their PACI resources.

In This Chapter

Server Management

The requests described in this section are used to obtain information and perform operations on individual servers.

List Servers

Description

Use this request to obtain the list of servers owned by the current user.

Syntax

```
GET baseURL/ve
```

Options

You can append an optional subscription ID to the request URL to list only the servers that belong to a specific subscription. The option is appended in the form "?subscription=*id*" where *id* is a numeric subscription ID (see examples below).

Request Parameters

None

Response

Element	Attribute	Description
`ve-list`		Container for server list.
`ve-info`		Container for an individual server info. This element may appear more than once (one for each server).
	`description`	Server description. Type: String
	`state`	A string describing the server state (e.g. CREATED, STOPPED, etc). Type: String
	`name`	Server name. Type: string
	`subscription-id`	ID of the subscription to which this server belongs. If subscription ID is specified in the request URL, this parameter is omitted. Type: int

Examples

The following example lists all servers owned by the user regardless of subscription.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-list>
    <ve-info subscription-id="1000001" name="web1" state="CREATED" description="Web
server 1"/>
    <ve-info subscription-id="1000002" name="web2" state="CREATED" description="Web
server 2"/>
</ve-list>
```

The following example lists only the servers for subscription 1000001.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve?subscription=1000001
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-list>
    <ve-info name="web1" state="CREATED" description="Web server 1"/>
</ve-list>
```

Start/Stop a Server

Description

Use this call to start or stop a specified server.

Syntax

```
PUT baseURL/ve/{ve-name}/(start | stop)
```

Request Parameters

None

Response

A text message describing the state of the operation. For example, "VE START initiated"

Example

Request

```
PUT  https://c2u-web:4465/paci/v1.0/ve/C2U-Linux-05/start
```

Response

```
VE START initiated
```

Create Server

Description

Use this request to create a new server. To create a server you must choose an operating system template, which will be used to configure the server and install an operating system in it. If the customer account to which the user belongs has more than one subscription, you will have to supply the subscription ID for which to create the server (see **Request Parameters** below). If the server you are trying to create exceeds the subscription server limit, the operation will fail with HTTP error 406: "Subscription limit for VE number exceeded".

Syntax

```
POST baseURL/ve
```

Request Parameters

Element	Attribute	Description
`ve`		Container for new server information.
	`custom-ns`	Specifies whether the user will be allowed to permanently modify the name server settings inside the server being created. Type: boolean

`name`		Server name. Type: string
`hostname`		Server hostname. Type: string
`description`		Server description. Type: string
`subscription-id`		Subscription ID for which to create the new server. This element must be included and populated when the customer account to which the user belongs has more than one subscription. If there's only one subscription, the element may be omitted. Type: int
`cpu`		Container for CPU information.
	`number`	Number of CPU cores. Type: int
	`power`	CPU clock rate in Mhz. Type: int
`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`no-of-public-ip`		Number of public IPv4 addresses. Type: int
`no-of-public-ipv6`		Number of public IPv6 addresses. Type: int
`ve-disk`		Container for hard disk information. *At the time of this writing only one hard disk can be added to a server. This will change in the future.*
	`local`	Local or network hard disk. Only local disks are supported at the time of this writing. Type: boolean
	`primary`	Specifies whether the disk should be a system disk. This is a reserved parameter. Type: boolean
	`size`	Hard disk size in gigabytes. Type: int

`platform`		Container for computing platform information.
`template-info`		Container or operating system template info.
	`name`	OS template name. A template contains all the necessary information about the virtualization technology it's using, the operating system, and other information. To obtain the list of the available templates, use the GET /template API call (p. 85). Type: string
`os-info`		Container for operating system information.
	`technology`	Virtualization technology: `CT` — Virtuozzo Container. `VM` — Parallels virtual machine. Type: string
	`type`	Operating system type.
`backup-schedule`		Container for backup schedule information. This element is optional. If omitted, no backups of the server will be performed.
	`name`	Backup schedule name. Backup schedules are created and maintained by the system administrator. Use the `schedule` interface (p. 88) to obtain the list of the available backup schedules. Type: string

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		A text message describing the status of the operation.
`password`		Server administrator password. If the password was supplied in the request, this element will contain that password. If the password was not provided, this element will contain an automatically generated one.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve
```

Request body

```
<?xml version="1.0" encoding="UTF-8"?>
<ve custom-ns="true">
  <name>Web40</name>
  <hostname>Web40.1000043.com</hostname>
  <description>VE Linux 40</description>
  <subscription-id>1000001</subscription-id>
  <cpu number="2" power="1600"/>
  <ram-size>512</ram-size>
  <bandwidth>100</bandwidth>
  <no-of-public-ip>2</no-of-public-ip>
  <no-of-public-ipv6>2</no-of-public-ipv6>
  <ve-disk local="true" size="3"/>
  <platform>
    <template-info name="centos-6-x86_64"/>
      <os-info technology="CT" type="linux-free"/>
  </platform>
  <backup-schedule name="daily"/>
</ve>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>VE create initiated</message>
    <password>152eyyBHO</password>
</pwd-response>
```

Create Server From Image

Description

Use this request to create a server from an existing image. The `{ve-name}` part of the request URL must contain the new server name (user-defined). The `{image-name}` part must contain the source image name. See also **Server Image Management** (p. 66).

Syntax 1

Use this syntax to create a server from an image when the customer account to which this user belongs has only one subscription.

```
POST baseURL/ve/{ve-name}/from/{image-name}
```

Syntax 2

Use this syntax when the customer account has more than one subscription. The `{subscription-id}` part must contain the subscription ID.

```
POST baseURL/ve/{subscription-id}/{ve-name}/from/{image-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example creates a server named Web105 from an image named Web-image-101.

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web105/from/Web-Image-101
```

The following example creates a server named Web106 from an image named Web-image-101 for subscription 1000001.

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/1000001/Web105/from/Web-Image-101
```

Clone Server

Description

Use this request to create a clone of an existing server. The `{ve-name}` part of the URL must contain the name of the source server. The `{new-server-name}` part must contain the new server name (user-defined). The IP addresses, gateway, and DNS settings for the new server will be set automatically. The rest of the server configuration will be inherited from the source server. The administrator password for the new server will be automatically generated and returned to the caller. If the server you are trying to create exceeds the subscription server limit, the operation will fail with HTTP error 406: "Subscription limit for VE number exceeded".

Syntax 1

Use this syntax to clone a server when the customer account has only one subscription.

```
POST baseURL/ve/{ve-name}/clone-to/{new-server-name}
```

Syntax 2

Use this syntax when the customer account has more than one subscription. The `{subscription-id}` part must contain the subscription ID.

```
POST baseURL/ve/{ve-name}/clone-to/{new-server-name}/for/{subscription-id}
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		A text message describing the status of the operation.
`password`		An automatically generated administrator password for the new server.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web40/clone-to/My_new_server
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>Clone VE initiated</message>
    <password>621zwPqs</password>
</pwd-response>
```

Recreate Server

Description

Use this request to recreate an existing server with the possibility of using a different OS template. This request essentially deletes an existing server and then creates a new server keeping the original server specifications. Instead of deleting a server and creating a new one using two separate operations, you can do the same using this single request.

The new server will have the same resources (CPU, RAM, bandwidth, disk size, and storage type) as the previous one. The new server will also have the same IP addresses, which cannot be guaranteed when deleting and then creating a server in two separate steps. The new server will also have the same auto scale and firewall rules and backup schedule if they were configured for the original server. If the original server was attached to a load balancer, the new one will be attached as well.

Syntax

```
POST baseURL/ve/ve-name/recreate[?template={new-template-name}][?drop-apps={true|false}]
```

The optional `template` parameter is used to specify an OS template. If the parameter is not specified, the server will be recreated using the same OS template as the original. To obtain the list of the available templates, use the GET /template API call (p. 85).

The optional `drop-apps` parameter is used to specify whether to reinstall the applications in the new server. If the parameter is specified and set to 'true', the applications installed in the original server will also be installed in the new server. The default value of this parameter is 'false', which means that all applications that are installed in the original server will be installed in the new one.

Request Parameters

None.

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		Contains the internal ID of the new server.
`password`		Contains the root/admin password.

Notes

If the original server is running, it will be stopped. The new server state will be CREATED.

The internal server id will be different from the original, but the history of the new server will include events from the original server, including all previous server versions if it was recreated more than once. The usage report for the new server will be reset.

The original server ID can be obtained from the optional `<previous-id>` element of the `<ve>` structure (p. 26). The element is only included for a server that was recreated in the past. For all other servers, the element is hidden.

Modify Server Configuration

Description

Use this request to modify the configuration of an existing server. Please note that this request cannot be used to modify the server backup schedule. See **Backup and Restore** (p. 44) for the information on how to change the backup schedule assignment.

Syntax

```
PUT baseURL/ve/{ve-name}
```

Request Parameters

Element	Attribute	Description
`reconfigure-ve`		Container for new configuration information.
`description`		Server description. Type: string
`change-cpu`		Container for CPU information.
	`number`	Number of CPU cores. Type: int
	`power`	CPU clock rate in Mhz. Type: int

`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`reconfigure-ipv4`		Container for public IPv4 address changes. **Note:** You can either add or remove IP addresses in one call, but not both.
`add-ip`		Container for IPv4 addresses to add to the server configuration.
	`number`	Specifies the number of addresses to add to the server configuration from the pool. Type: int
`drop-ip`		Container for IPv4 addresses to remove from the server configuration.
	`ip`	Whitespace-separated list of CIDR-compliant (ip/mask) IPv4 addresses to remove from the server configuration. Type: string
`reconfigure-ipv6`		Container for public IPv6 address changes. **Note:** You can either add or remove IP addresses in one call, but not both.
`add-ip`		Container for IPv6 addresses to add to the server configuration.
	`number`	Specifies the number of addresses to add to the server configuration from the pool. Type: int
`drop-ip`		Container for IPv6 addresses to remove from the server configuration.
	`ip`	Whitespace-separated list of CIDR-compliant (ip/mask) IPv6 addresses to remove from the server configuration. Type: string

`primary-disk-size`		Hard disk size, in gigabytes. Type: int
`custom-ns`		Specifies whether the user should be allowed to permanently modify the name server settings inside the server. This element is optional. Possible values: 0 — do not allow the server user to modify the name server settings. This is the default value. 1 — allow to modify the settings. Type: int

Response

A string describing the state of the operation.

Examples

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web40
```

Request body 1

Modifying the CPU, RAM, and disk size.

```
<?xml version="1.0" encoding="UTF-8"?>
<reconfigure-ve>
  <change-cpu number="2" power="900"/>
  <ram-size>1280</ram-size>
  <primary-disk-size>15</primary-disk-size>
  <custom-ns>1</custom-ns>
</reconfigure-ve>
```

Request body 2

Adding two IPv4 addresses.

```
<?xml version="1.0" encoding="UTF-8"?>
<reconfigure-ve>
  <reconfigure-ipv4>
    <add-ip number="2"/>
  </reconfigure-ipv4>
</reconfigure-ve>
```

Request body 3

Removing the specified IPv4 addresses.

```
<?xml version="1.0" encoding="UTF-8"?>
<reconfigure-ve>
  <reconfigure-ipv4>
    <drop-ip ip="10.29.184.100 10.29.184.102"/>
  </reconfigure-ipv4>
```

```
</reconfigure-ve>
```

Reset Server Administrator Password

Description

Use this request to reset the server administrator password. The new password will be automatically generated.

Syntax

```
POST baseURL/ve/{ve-name}/reset-password
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		A text message describing the status of the operation.
`password`		New password.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web40/reset-password
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>Password reset initiated</message>
    <password>151rhpJEB</password>
</pwd-response>
```

Obtain Server Information

Description

Use this request to obtain the information about the specified server. The `{ve-name}` part of the request URL must contain the server name.

Syntax

```
GET baseURL/ve/{ve-name}
```

Request Parameters

None

Response

Element	Attribute	Description
ve		Container for server information.
id		Server database ID (used internally). Type: int
previous-id		Previous server database ID. This element is optional and is only included for servers that were recreated (p. 22) in the past.
uuid		Server UUID. Type: string
hnId		ID of the hardware node on which this server resides. Type: string
customer-id		ID of the customer account to which the server belongs. Type: int
name		Server name. Type: string
description		Server description. Type: string
subscription-id		ID of the subscription to which the server belongs. Type: int
cpu		Container for CPU information.
	number	Number of CPU cores. Type: int
	power	CPU clock speed in megahertz. Type: int

`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`ve-disk`		Container for hard disk information. *At the time of this writing, a server can have only one hard disk. This will change in the future.*
	`id`	Disk database ID (used internally) Type: int
	`local`	Specified whether this is a local or a network disk. *Only local disks are supported at the the time of this writing.* Type: boolean
	`size`	Disk size in gigabytes. Type: int
	`created`	Disk status. Type: boolean Possible values: `true` — the disk creation process completed. `false` — the disk is being created and is not available for use.
`platform`		Container for operating system template and related information.
`template-info`		Container for OS template information.
	`name`	Template name. Type: string
	`vendorId`	Template vendor ID. Type: string
	`c2uId`	PACI template ID (used internally). Type: string
`os-info`		Container for operating system information. This information is a part of the OS template info.
	`type`	OS type (Linux. Windows, etc.) Type: string

	`technology`	Virtualization technology used. Type: string Possible values: `CT` - Virtuozzo Container, `VM` - Parallels virtual machine.
`network`		Container for network information.
	`private_ip`	Private IP address and mask. Type: string
`public-ip`		Container for public IPv4 address info. This element may appear more than once (one for each IP address).
	`id`	Database ID (used internally). Type: int
	`address`	IP address and mask. Type: string
	`gateway`	Gateway IP address. Type: string
	`chunk-ref`	Used internally. Type: int
`public-ipv6`		Container for public IPv6 address info. This element may appear more than once (one for each IPv6 address).
	`id`	Database ID (used internally). Type: int
	`address`	IPv6 address and mask. Type: string
	`gateway`	Gateway IPv6 address. Type: string
`backup-schedule`		Container for backup schedule info. A backup schedule is created and configured by the system administrator. The user can select one of the existing backup schedules when a server is created.
	`name`	Backup schedule name. Type: string

`state`		A string describing the server state or transition. Type: string
`primary-disk-id`		Used internally. Type: int
`template-id`		Used internally. Type: int
`admin`		Container for the server administrator credentials.
	`login`	Server administrator login name. Type: string
	`password`	Password. Type: string
`last-operation-rc`		The returned code of the last operation. Type: int
`app-info`		Container for installed application template information. This element may appear more than once (one for each installed template).
	`app-template`	Application template name. Type: string
	`for-os`	The name of the OS template for which this application is designed. Type: string
	`c2u-version`	The template PACI version. Type: string
	`installed-at`	Template location. Type: string
	`installed-ok`	Specifies whether the template was installed properly. Type: boolean
	`uninstalled-at`	The template previous location. Type: string
	`uninstalled-ok`	Specifies whether the template uninstall was completed successfully. Type: boolean
	`app-template-id`	Application template ID (used internally). Type: int

`load-balancer`		Name of the load balancer attached to this server. If no load balancer is used, the element will be absent. Type: string
`steady-state`		A string describing the server steady state. This parameter normally never contains transition states that change quickly. When monitoring transitions, look at the `state` parameter above. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve>
    <id>4</id>
    <uuid>6733915.132fd360949._7ffe</uuid>
    <hnId>5</hnId>
    <customer-id>1</customer-id>
    <name>server1</name>
    <description></description>
    <subscription-id>1</subscription-id>
    <cpu power="1000" number="1"/>
    <ram-size>128</ram-size>
    <bandwidth>100</bandwidth>
    <ve-disk created="true" size="1" local="true" id="0"/>
    <platform>
        <template-info c2uId="1" vendorId="5" name="centos-5-x86"/>
        <os-info technology="CT" type="linux-free"/>
    </platform>
    <network private-ip="10.40.119.201/8">
        <public-ip chunk-ref="1" gateway="10.30.0.1" address="10.30.119.201/16"
id="3"/>
    </network>
    <backup-schedule name="hourly"/>
    <state>STOPPED</state>
    <primary-disk-id>0</primary-disk-id>
    <template-id>1</template-id>
    <admin password="[hidden]" login="root"/>
    <steady-state>STOPPED</steady-state>
</ve>
```

Obtain Server History

Description

Use this request to obtain the modification history for the specified server. Every time a server configuration is changed, a snapshot of the configuration is taken and saved. This request allows to retrieve these records and use them for statistics.

Syntax 1

Use this syntax to retrieve the specified number of history records. The `{records}` part is used to specify the number of records (from the end) to include in the result set.

```
GET baseURL/ve/{ve-name}/history/{records}
```

Syntax 2

Use this syntax to retrieve the records that were created during the specified date-time period. The `{from-inclusive}` and `{to-exclusive}` parts must contain the datetime values specifying the datetime interval.

```
GET baseURL/ve/{ve-name}/history/{from-inclusive}/{to-exclusive}
```

See **Datetime Format** (p. 93) for the information on datetime formatting.

Request Parameters

None

Response

Element	Attribute	Description
`ve-history`		
`ve-snapshot`		
	`cpu`	CPU clock speed, in megahertz. Type: int
	`ram`	RAM size, in megabytes. Type: int
	`local-disk`	Local disk size, in gigabytes. Type: int
	`nbd`	*This parameter is reserved for future use.* Type: int
	`bandwidth`	Network bandwidth, in kbps. Type: int
	`backup-schedule`	Backup schedule name. Type: string
	`last-operation-rc`	Last operation return code. Type: int

	`last-touched-from`	Used internally. Type: string
	`state`	Server state right after the modification operation began executing. Type: string
	`steady-state`	Last known steady state. Type: string
	`last-changed-by`	The Name of the user who performed the modification. Type: string
	`event-timestamp`	The modification event timestamp. Type: string
	`no-of-public-ip`	Number of public IPv4 addresses. Type: int
	`no-of-public-ipv6`	No of public IPv6 addresses. Type: int
	`is-lb`	Specifies whether this server is a load balancer or a regular server: `true` -- load balancer `false` — regular server Type: boolean
	`private-incoming-traffic`	Private incoming traffic, in bytes. Type: long
	`private-outgoing-traffic`	Private outgoing traffic, in bytes. Type: long
	`public-incoming-traffic`	Public incoming traffic, in bytes. Type: long
	`public-outgoing-traffic`	Public outgoing traffic, in bytes. Type: long

Examples

The following example retrieves the three most recent server history records.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40/history/3
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-history>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2013-06-23 16:34:32.911962+04" last-changed-
by="tobotras" state="CREATION_IN_PROGRESS" last-touched-from="im1" bandwidth="100"
nbd="0" local-disk="0" ram="512" cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2013-06-23 16:34:33.131139+04" last-changed-
by="tobotras" state="CREATION_IN_PROGRESS" last-touched-from="im1" backup-
schedule="hourly" bandwidth="100" nbd="0" local-disk="0" ram="512" cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2013-06-23 16:35:03.408376+04" last-changed-
by="InstanceManager" steady-state="CREATED" state="CREATED" last-touched-from="im1"
backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512" cpu="1000"/>
</ve-history>
```

The following example retrieves the server history records that were created between the specified start and end dates.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40/history/2013-06-27 21:00 CET/2013-06-29
21:00 CET
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-history>
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="49785508"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2013-06-28 00:00:01.321945+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="50043427"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2013-06-28 00:30:01.288039+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="50301986"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2011-06-28 01:00:01.30681+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
```

```
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="50559169"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2013-06-28 01:30:01.287452+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
</ve-history>
```

Obtain Server Usage Info

Description

Use this request to obtain the usage information for the specified server. The `{from-inclusive}` and `{to-exclusive}` parts must contain the datetime values specifying the datetime interval.

Syntax

```
GET baseURL/ve/{ve-name}/usage/{from-inclusive}/{to-exclusive}
```

See **Datetime Format** (p. 93) for the information on datetime formatting.

Request Parameters

None

Response

Element	Attribute	Description
`ve-resource-usage-report`		Container for usage report data.
	`ve-name`	Server name. Type: string
	`ve-id`	Server ID. Type: int
	`os`	Operating system type. Type: string
	`technology`	Virtualization technology used. Type: string
	`life-time-in-minutes`	Server lifetime in minutes. Type: long
	`is-load-balancer`	Specifies whether the server is a load balancer. Type: boolean

resource-usage		This element may appear more than once, one for each resource type.
	value	Resource usage value. Type: long
	resource-usage-type	Resource usage type. Possible values are: "while-running" "while-stopped" Type: string
	resource-type	Resource type. Possible values are: "cpu-hours" "ram-hours" "hdd-hours" "pcs-hours" "bandwidth-hours" "public-ip-hours" Type: string
ve-traffic		Container for server traffic info. This element may appear more than once, one for each traffic type.
	traffic-type	Traffic type. Possible values are: "private-incoming" "private-outgoing" "public-incoming" "public-outgoing" Type: string
	used	Total traffic used. Type: long
active-backup-schedule		Active backup schedule info. This element may appear more than once.
	schedule-name	Backup schedule name. Type: string

Examples

The following example retrieves the server information between the specified start and end dates.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40/usage/2013-04-27 21:00 CET/2013-04-29 21:00
CET
```

Delete Server

Description

Use this request to permanently delete a server. Please note that you can only delete a fully stopped server. If a server is in a transition state (stopping, starting, a disk is being attached to it, etc.) it cannot be deleted.

Syntax

```
DELETE baseURL/ve/{ve-name}
```

Request Parameters

None

Response

A string describing the status of the operation.

Example

Request

```
DELETE  https://c2u-web:4465/paci/v1.0/ve/Web40
```

Response

```
VE DELETE initiated
```

Firewall Management

This section describes requests that allow to configure a firewall in the the specified server.

List Firewall Rules

Description

Use this request to obtain a list of existing firewall rules for the specified server. The `{ve-name}` part of the request URL must contain the server name.

Syntax

```
GET baseURL/ve/{ve-name}/firewall
```

Request Parameters

None

Response

Element	Attribute	Description
`firewall`		Container for firewall rule list.
`rule`		Container for firewall information. This element may appear more than once (one for each rule).
	`id`	Firewall rule database ID (used internally). Type: int
	`name`	Firewall rule name. Type:string
	`protocol`	Communication protocol (TCP, UDP). Type: string
	`local-port`	Local port number. Type: int
	`remote-port`	Remote port number. Type: int

`remote-net`		Remote address and an optional mask. If no mask is specified, the address indicates a host address. This element may appear more than once. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/firewall
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<firewall>
    <rule remote-port="0" local-port="25" protocol="TCP" name="smtp" id="6">
        <remote-net>192.168.1.0/24</remote-net>
        <remote-net>192.168.2.0/32</remote-net>
        <remote-net>dead:beef::/64</remote-net>
    </rule>
    <rule remote-port="2" local-port="80" protocol="TCP" name="http" id="7">
        <remote-net>192.168.3.0/24</remote-net>
        <remote-net>192.168.4.0/32</remote-net>
        <remote-net>dead:beef:abcd::/64</remote-net>
    </rule>
</firewall>
```

Create Firewall Rules

Description

Use this request to create firewall rules. The `{ve-name}` part of the request URL must contain the server name.

Syntax

```
POST baseURL/ve/{ve-name}/firewall
```

Request Parameters

Element	Attribute	Description
`firewall`		Container for firewall rule list. You can create more than one rule in a single request.
`rule`		Container for firewall information. This element may appear more than once (one for each rule).
	`name`	Firewall rule name. Type:string
	`protocol`	Communication protocol (TCP, UDP). Type: string

	`local-port`	Local port number. Type: int
	`remote-port`	Remote port number. Type: int

`remote-net`		Remote address and an optional mask. If no mask is specified, the address indicates a host address. This element may appear more than once. Type: string

Response

A text message describing the operation status.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web60/firewall
```

Request Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<firewall>
  <rule name="smtp" protocol="TCP" local-port="25" remote-port="0">
    <remote-net>192.168.1.0/24</remote-net>
    <remote-net>192.168.2.0</remote-net>
    <remote-net>DEAD:BEEF::/64</remote-net>
  </rule>
  <rule name="http" protocol="TCP" local-port="80" remote-port="2">
    <remote-net>192.168.3.0/24</remote-net>
    <remote-net>192.168.4.0</remote-net>
    <remote-net>DEAD:BEEF:ABCD::/64</remote-net>
  </rule>
</firewall>
```

Response

```
Firewall configuration started
```

Modify Firewall Rules

Description

Use this request to modify an existing firewall. The request replaces all existing rules with the new ones. To keep existing rules and add more, first obtain the list of the existing rules, then add new rules to it and use the complete list as an input.

Syntax

```
PUT baseURL/ve/{ve-name}/firewall
```

Request Parameters

Element	Attribute	Description
`firewall`		Container for firewall rule list. You can specify more than one rule in a single request.

`rule`		Container for firewall information. This element may appear more than once (one for each rule).
	`name`	Firewall rule name. Type:string
	`protocol`	Communication protocol (TCP, UDP). Type: string
	`local-port`	Local port number. Type: int
	`remote-port`	Remote port number. Type: int

`remote-net`		Remote address and an optional mask. If no mask is specified, the address indicates a host address. This element may appear more than once. Type: string

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web40/firewall
```

Request Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<firewall>
  <rule name="smtp1" protocol="TCP" local-port="25" remote-port="0">
    <remote-net>192.168.2.0/24</remote-net>
    <remote-net>192.168.3.0</remote-net>
    <remote-net>DEAD:BEEF::/64</remote-net>
  </rule>
  <rule name="http1" protocol="TCP" local-port="80" remote-port="2">
    <remote-net>192.168.4.0/24</remote-net>
    <remote-net>192.168.5.0</remote-net>
    <remote-net>DEAD:BEEF:ABCD::/64</remote-net>
  </rule>
</firewall>
```

Response

```
Firewall re-configuration started
```

Delete Firewall Rules

Description

Use this request to delete all existing firewall rules. To delete a specific rule, retrieve all existing rules, modify the result set as needed and then use it as an input in the firewall modification request (p. 41).

Syntax

```
DELETE baseURL/ve/{ve-name}/firewall
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/ve/Web60/firewall
```

Response

```
Firewall removal started
```

Backup and Restore

This section describes requests that allow to manage server backups. You can backup your servers using one of the following methods:

- Automatically, according to a selected backup schedule (p. 44).
- On demand, whenever a backup is required (p. 45).

The following sections describe each backup type and additional operations in detail.

Set Backup Schedule for Server

Description

Use this request to assign a backup schedule to the specified server. Backup schedules are created and configured by the system administrator and define when and how often the server backups will be performed. Backup schedules also define the maximum number of incremental backups and a maximum number of backups to keep on the backup server.

The `{schedule-name}` part must contain the name of a predefined backup schedule. You can obtain the list of the existing schedules using the `schedule` interface (p. 88). You cannot create your own backup schedules.

Syntax

```
PUT baseURL/ve/{ve-name}/schedule/{schedule-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/schedule/daily
```

Response

```
Backup schedule assigned
```

Remove Backup Schedule from Server

Description

Use this call to remove a backup schedule from a server.

Syntax

```
PUT baseURL/ve/{ve-name}/nobackup
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/nobackup
```

Response

```
Backup schedule removed
```

Perform On-Demand Backup

Description

Use this request to perform an on-demand backup.

Syntax

```
POST baseURL/ve/{ve-name}/backup
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/backup
```

Response

```
VE BACKUP initiated
```

List Backups

Description

List available backups for the specified server. The `{from-inclusive}` and `{to-exclusive}` parts of the request URL must specify a datetime interval for which to retrieve the backups.

Syntax

```
GET baseURL/ve/{ve-name}/backups/{from-inclusive}/{to-exclusive}
```

See **Datetime Format** (p. 93) for the information on datetime formatting.

Request Parameters

None

Response

Element	Attribute	Description
`ve-backups`		Container for backups.
`backup`		Container for backup information. This element may appear more than once (one for each backup).
	`im-backup-id`	Backup ID (used internally) Type: int
	`cloud-backup-id`	Cloud backup ID. Use this ID when restoring a server from a backup, getting the backup info, or deleting a backup. Type: string

	`schedule-name`	Backup schedule name. Scheduled backups will have this attribute with a value specifying the schedule used. On-demand backup will not have it. This can be used as an indication of whether this is a scheduled or an on-demand backup. Type: string
	`started`	Backup start date/time. Type: string
	`ended`	Backup end date/time. Type: string
	`successful`	Specifies whether the backup was successful. Type: boolean
	`backup-size`	Backup file size. Type: int
	`backup-node-name`	Name of the backup node. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/backups/2013-06-27 19:00 CET/2013-06-29
22:00 CET
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-backups>
    <backup backup-node-name="c2u-deploy.pl.com" backup-size="1" successful="true"
ended="2013-06-28 20:01:34.710062+04" started="2013-06-28 20:00:00.107158+04" schedule-
name="hourly" cloud-backup-id="{d79d63f2-1136-409f-8573-a3f37bffe83b}" im-backup-
id="3"/>
    <backup delta_of="3" backup-node-name="c2u-deploy.pl.com" backup-size="1"
successful="true" ended="2013-06-28 21:01:34.176135+04" started="2013-06-28
21:00:00.142492+04" schedule-name="hourly" cloud-backup-id="{d79d63f2-1136-409f-8573-
a3f37bffe83b}.2" im-backup-id="5"/>
    <backup backup-node-name="c2u-deploy.pl.com" backup-size="1" successful="true"
ended="2013-06-29 12:01:32.550893+04" started="2013-06-29 12:00:00.100254+04" schedule-
name="daily" cloud-backup-id="{d79d63f2-1136-409f-8573-a3f37bffe83b}.3" im-backup-
id="11"/>
</ve-backups>
```

Restore Server from Backup

Description

Restore a specified server from a specified backup. A server must be stopped in order to perform a restore operation. The `{cloud-backup-id}` part of the request URL must contain the backup ID. See **List Backups** (p. 46) for the information on how to obtain the list of the available backups and their IDs. Please note that the complete backup ID string must be specified in the request, including curly brackets and any other leading and trailing characters (if any).

Syntax

```
PUT baseURL/ve/{ve-name}/restore/{cloud-backup-id}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/restore/{d79d63f2-1136-409f-8573-
a3f37bffe83b}
```

Response

```
VE RESTORE_FROM_BACKUP initiated
```

Obtain Backup Info

Description

Obtain the information about the specified backup. The `{cloud-backup-id}` part of the of the request URL must contain a valid backup ID (see **List Backups** (p. 46)).

Syntax

```
GET baseURL/ve/{ve-name}/backup/{cloud-backup-id}
```

Request Parameters

None

Response

Element	Attribute	Description
`backup`		Container for backup information.
	`im-backup-id`	Backup ID (used internally) Type: int
	`cloud-backup-id`	Cloud backup ID. Type: string
	`schedule-name`	Backup schedule name. Type: string
	`started`	Backup start date/time. Type: string
	`ended`	Backup end date/time. Type: string
	`successful`	Specifies whether the backup was successful. Type: boolean
	`backup-size`	Backup file size. Type: int
	`backup-node-name`	Name of the backup node. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/backup/{d79d63f2-1136-409f-8573-
a3f37bffe83b}
```

Delete Backup

Description

Delete a specified backup. Please note that you can only delete an on-demand backup. Scheduled backups cannot be deleted by the user.

The `{cloud-backup-id}` part of the of the request URL must contain a valid backup ID (see **List Backups** (p. 46)).

Syntax

```
DELETE baseURL/ve/{ve-name}/backup/{cloud-backup-id}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/ve/Web60/backup/{d79d63f2-1136-409f-8573-
a3f37bffe83b}
```

Response

```
VE DELETE_BACKUP initiated
```

Auto Scaling Management

The functionality described in this section can be used to manage automatic resource scaling on your servers. Using the auto scaling API you can create and manage rules to adjust CPU and RAM settings on a server at peak times.

Auto Scaling Data Types

This section describes common data types used in the Auto Scaling API calls.

Type

autoscale-settings

Syntax

Element	Attribute	Description
`autoscale-rule`		Container for an individual auto scaling rule. This element may appear more than once (one for each rule).
	`enabled`	*Not currently used.* Type: boolean.
	`deleted`	*Used for output only.* If set to true, indicates that a request to delete the rule has been received but has not been completed yet. Type: boolean.
	`metric`	The type of resource to auto scale. Possible values: "cpu" — CPU "ram" — RAM Type: string.
	`version`	*Used for output only.* Version information. Type: int
	`updated`	*Used for output only.* The date and time of the last rule update operation. Type: timestamp
	`update-delivered-ok`	*Used for output only.* Indicates whether the rule update has been processed and delivered to the server. True - update completed successfully. False - update is still in progress (could also indicate a failure). Type: boolean
	`update-delivered`	*Used for output only.* The date and time when the rule update has been delivered to the server. Type: timestamp

	`allow-migration`	Specifies whether to allow migration of the server to a different hardware node if needed. Type: boolean. Default is True.
	`allow-restart`	Specifies whether to allow the server restart if needed. Type: boolean. Default is False.
`limits`		Container for scaling limits.
	`min`	The minimum allowed resource limit. This value is used when a resource is downscaled. Specified in megabytes for RAM and megahertz for CPU. Type: int
	`max`	The maximum allowed resource limit. This value is used when a resource is upscaled. Specified in megabytes for RAM and megahertz for CPU. Type: int
	`step`	An increment or decrement of a resource usage to be performed in a single scaling step. Specified in megabytes for RAM and megahertz for CPU. Type: int
`thresholds`		Container for threshold information.
`up`		Upscaling threshold.
	`thresholds`	Specifies the total load of the resource, in percent. When this value is reached and maintained for the specified period of time (see `period` below), the resource upscaling will be performed. Type: int (0-100).
	`period`	Specifies the time period, in seconds, of the threshold state after which the resource upscaling will be performed. Type: int
`down`		Downscaling threshold.
	`thresholds`	Specifies the total load of the resource, in percent. When this value is reached and maintained for the specified period of time (see `period` below), the resource downscaling will be performed. Type: int (0-100).
	`period`	Specifies the time period, in seconds, of the threshold state after which the resource downscaling will be performed. Type: int

Create Auto Scaling Rules

Description

Use this request to create auto scaling rules for the specified server.

Syntax

```
POST baseURL/ve/ve-name/autoscale
```

Request Parameters

Element	Type	Description
`autoscale-data`	`autoscale-settings` (p. 50)	Container for auto scaling settings to create on the server.

Response

Element	Type	Description
`autoscale`		Container for auto scaling settings returned from the PACI server.
`current`	`autoscale-settings` (p. 50)	Contains current auto scaling rules.
`ongoing`	`autoscale-settings` (p. 50)	Contains auto scaling rules that are in the process of being applied to the server.

Examples

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web60/autoscale
```

Request body

```
<?xml version="1.0" encoding="UTF-8"?>
<autoscale-data>
  <autoscale-rule metric="cpu">
    <limits min="2" max="4" step="1"/>
    <thresholds>
      <up threshold="90" period="5"/>
      <down threshold="40" period="20"/>
    <thresholds>
  </autoscale-rule>
  <autoscale-rule metric="ram">
    <limits min="2000" max="4000" step="1000"/>
    <thresholds>
      <up threshold="90" period="5"/>
      <down threshold="40" period="20"/>
```

```
    <thresholds>
  </autoscale-rule>
</autoscale-data>
```

Update Auto Scaling Rules

Description

Use this request to update existing auto scaling rules for the specified server.

Syntax

```
PUT baseURL/ve/ve-name/autoscale
```

Request Parameters

Element	Type	Description
`autoscale-data`	`autoscale-settings` (p. 50)	Container for the new auto scaling settings.

Response

Element	Type	Description
`autoscale`		Container for auto scaling data returned from the PACI server.
`current`	`autoscale-settings` (p. 50)	Contains current auto scaling rules.
`ongoing`	`autoscale-settings` (p. 50)	Contains auto scaling rules that are in the process of being applied to the server.

Examples

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/autoscale
```

Request body

```
<?xml version="1.0" encoding="UTF-8"?>
<autoscale-data>
  <autoscale-rule metric="cpu">
    <limits min="2" max="4" step="1"/>
    <thresholds>
      <up threshold="90" period="5"/>
      <down threshold="40" period="20"/>
    <thresholds>
  </autoscale-rule>
  <autoscale-rule metric="ram">
    <limits min="2000" max="4000" step="1000"/>
    <thresholds>
      <up threshold="90" period="5"/>
      <down threshold="40" period="20"/>
```

```
    <thresholds>
  </autoscale-rule>
</autoscale-data>
```

Get Auto Scaling Rules

Description

Use this request to obtain auto scaling rules for the specified server.

Syntax

```
GET baseURL/ve/ve-name/autoscale
```

Request Parameters

None

Response

Element	Type	Description
`autoscale`		Container for auto scaling data returned from the PACI server.
`current`	`autoscale-settings` (p. 50)	Contains current auto scaling rules.
`ongoing`	`autoscale-settings` (p. 50)	Contains auto scaling rules that are in the process of being applied to the server.

Examples

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/autoscale
```

Drop Auto Scaling Rules

Description

Use this request to drop auto scaling rules from the specified server.

Syntax

```
DELETE baseURL/ve/ve-name/autoscale
```

Request Parameters

None

Response

A text message describing the state of the operation.

Examples

Request

```
DELETE https://c2u-web:4465/paci/v1.0/ve/Web60/autoscale
```

Get Auto Scaling History

Description

Use this request to obtain auto scaling history.

Syntax 1

Use this syntax to retrieve the specified number of history records. The `{nrecrods}` part must contain the number of records from the end to include in the result set.

```
GET baseURL/ve/ve-name/autoscale/history/{nrecords}
```

Syntax 2

Use this syntax to retrieve the records that were created during the specified date-time period. The `{from-inclusive}` and `{to-exclusive}` parts must contain the datetime values specifying the interval. See **Datetime Format** (p. 93) for the information on datetime formatting.

```
GET baseURL/ve/ve-name/autoscale/history/{from-inclusive}/{to-
inclusive}?average-period={seconds}&tail={seconds}
```

Options

The optional `average-period` and `tail` parameters in **Syntax 2** can be used for the server-side averaging of the resource consumption values. You can use these parameters to limit the volume of the returned data and to obtain the data more quickly if needed.

The `average-period` parameter is used to specify an interval (in seconds) for which to calculate an average resource consumption value. For example, if the parameter is included in the request and contains a value of 900 seconds, only an average value of every consecutive 15-minute period will be included in the response. If the parameter is not included, the entire set of the actual values will be returned. Please note that the network traffic values are calculated as maximum over the specified period. All other statistics are calculated as average values.

The `tail` parameter is used to specify the time (in seconds) at the end of the average-period for which the averaging should NOT be performed. For example, if `average-period=3600` and `tail=900`, one average value (or the maximum value for network traffic) will be included in the response for the first 45 minutes of every consecutive one-hour period. For the last 15 minutes of every hour, the response will contain a complete set of the actual values.

You should also be aware that if multiple requests are executed against the same server and are expecting to receive the complete statistical data, the CPU, memory, and network load on the server side may increase drastically, which may significantly slow down the processing of the requests. Using the averaging approach, you can avoid this potential problem.

Request Parameters

None

Response

Element	Attribute	Description
`resource-consumption-and-autoscale-history`		Container for auto scaling history.
`resource-consumption-sample`		
	`ram-usage`	RAM usage. Type: int
	`cpu-usage`	CPU usage. Type: int
	`private-incoming-traffic`	Private incoming traffic. Type: long
	`private-outgoing-traffic`	Private outgoing traffic. Type: long
	`public-incoming-traffic`	Public incoming traffic. Type: long
	`public-outgoing-traffic`	Public outgoing traffic. Type: long
	`node-seq-no`	Node sequence number. Type: long
	`node-timestamp`	Node timestamp. Type: string
	`paci-timestamp`	PACI timestamp. Type: string

	`cpu`	Number of CPUs. Type: int
	`ram`	RAM size. Type: int
	`bandwidth`	Bandwidth. Type: int
`autoscale-event`		
	`direction`	Auto scale direction. Type: string
	`rule-version`	Auto scale rule version. Type: int
	`node-seq-no`	Node sequence number. Type: long
	`node-timestamp`	Node timestamp. Type: long
	`metric`	Auto scale metric. Type: string
	`new-value`	New value. Type: int
	`node_uuid`	Node UUID. Type: string
	`started`	Event started. Type: string
	`ended`	Event ended. Type: string
	`ended-ok`	Event ended OK or not. Type: boolean
`autoscale-rule`		
	`enabled`	Rule enabled or not. Type: boolean
	`deleted`	Rule deleted. Type: boolean
	`metric`	Rule metric. Type: string
	`version`	Rule version. Type: int

	`updated`	Rule updated. Type: timestamp
	`update-delivered-ok`	Updated delivered OK or not. Type: boolean
	`update-delivered`	Update delivered. Type: timestamp
	`allow-migration`	Allow migration or not. Type: boolean
	`allow-restart`	Allow restart or not. Type: boolean
`limits`		
	`min`	Minimum value. Type: int
	`max`	Maximum value. Type: int
	`step`	Step. Type: int
`thresholds`		
`up`		
	`threshold`	Threshold. Type: int
	`period`	Period. Type: int
`down`		
	`threshold`	Threshold. Type: int
	`period`	Period. Type: int

Examples

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/autoscale/history/10
```

Application Template Installation

The requests described in this section are used to obtain information about the available application templates and install them into servers. An application template represents a software application that can be installed into a server. An application template is created for a specific operating system type, so it must be compatible with the OS template that was used to create a server in order to be installed in it. To install an application template, first obtain a list of the available templates, then choose the template of interest and obtain a detailed information about it. Make sure that the template is compatible with the OS template of the target server and then install it.

List Application Templates

Description

Use this request to obtain a list of the available application templates.

Syntax

```
GET baseURL/application-template
```

Request Parameters

None

Response

Element	Attribute	Description
`application-list`		Container for application template list.
`application-template`		Container for application template information. This element may appear more than once (one for each template).
	`id`	Template ID (used internally). Type: int
	`name`	Template name. Use this value in other requests that expect template name as a parameter. Type: string

	`active`	Specifies whether the template is active or not. Inactive templates cannot be installed into servers. Type: boolean
	`c2u-version`	Template PACI version (used internally). Type: string
	`for-os`	Name of the operating system template for which this application template is designed. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/application-template
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application-list>
    <application-template for-os="centos-6-x86_64" c2u-version="1" active="true"
name="php" id="1"/>
</application-list>
```

Get Application Template Info

Description

Use this request to obtain a detailed information about a specified application template. The `{name}` parameter must contain the application template name. The `{for-os}` parameter must contain the name of the operating system template for which the template is designed. The `{name}` and `{for-os}` parameters together uniquely identify an application template. There could be multiple templates with the same name but designed for different operating systems.

Syntax

```
GET baseURL/application-template/{name}/{for-os}
```

Request Parameters

None

Response

Element	Attribute	Description
`application-template`		Container for application template information. This element may appear more than once (one for each template).
	`id`	Template ID (used internally). Type: int
	`name`	Template name. Type: string
	`c2u-version`	Template PACI version (used internally). Type: string

	`active`	Specifies whether the template is active or inactive. Inactive templates cannot be installed into servers. Type: boolean
	`for-os`	Name of the operating system for which this template was designed. Type: string

description		Template description. Type: string

Example

The following example obtains information about an application template named "php".

Request

```
GET https://c2u-web:4465/paci/v1.0/application-template/php/centos-6-x86_64
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application-template for-os="centos-6-x86_64" c2u-version="1" active="true" name="php"
id="1">
    <description>php</description>
</application-template>
```

Install Application Template Into Server

Description

Use this request to install an application template (or multiple templates) into a server. The application template must be compatible with the OS template installed in the target server.

Syntax 1

Installs a specified template. The `{app-name}` part of the URL must contain the application template name.

```
PUT baseURL/ve/{ve-name}/install/{app-name}
```

Syntax 2

Installs multiple templates. The `name={app1}&name={app2}&...` list must contain the names of the templates to install separated by an ampersand (&).

```
PUT baseURL/ve/{ve-name}/install?name={app1}&name={app2}&...
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example installs an application template named "ha-proxy" into a server named "Web101".

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web101/install/ha-proxy
```

Response

```
VE INSTALL_APPLICATION initiated
```

Reset Application Templates in Server

Description

Use this request to reset the installed application templates in a server. As an input the request take a list of application templates that you want installed in the server. If the templates are already installed, they will be left untouched; if not, they will be installed. All other installed application templates will be removed from the server.

The `name={app1}&name={app2}&...` list must contain the names of the application templates to install separated by an ampersand (&).

Syntax

```
POST baseURL/ve/{ve-name}/application?name={app1}&name={app2}&...
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web101/application/name=ha-proxy&name=some-app&name=super-app
```

Response

```
VE SET_APPLICATIONS initiated
```

Delete Application Template from Server

Description

Use this request to remove an application template from a server. The `{app-name}` part of the URL must contain the application template name.

Syntax

```
DELETE baseURL/ve/{ve-name}/application/{app-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

The following example removes the "ha-proxy" application template from the "Web101" server.

```
DELETE https://c2u-web:4465/paci/v1.0/ve/Web101/application/ha-proxy
```

Response

```
VE UNINSTALL_APPLICATION initiated
```

Image Management

The requests described in this section are used to manage server images. A server image is created from an existing server and can be used later to create new servers. The new servers will be exact copies of the source server except the network settings to avoid collisions.

List Images

Description

Use this request to obtain a list of the existing server images.

Syntax

```
GET baseURL/image
```

Request Parameters

None

Response

Element	Attribute	Description
`image-list`		Container for image list.
`image-info`		Container for image information. This element may appear more than once (one for each image).
	`name`	Image name. Type: string
	`size`	Image size, in gigabytes. Type: int
	`created`	Image creation date. Type: string
	`subscription-id`	Subscription ID to which this image belongs. Type: int
	`image-of`	The name of the server from which this image was created. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/image
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<image-list>
    <image-info image-of="Web40" subscription-id="100001" created="2013-09-01
19:19:17.84091+04" size="2" name="lbimage"/>
</image-list>
```

Get Image Info

Description

Use this request to obtain a detailed information for the specified server image.

Syntax

```
GET baseURL/image/{image-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`ve-image`		Container for image information.
	`id`	Image database ID (used internally). Type: int
	`bnode-uuid`	Used internally. Type: string
	`customer-id`	Customer ID. Type: int
	`subscription-id`	Subscription ID. Type: int
	`image-of`	The name of the server from which this image was created. Type: string

	`name`	Image name. Type: string
	`cpu-number`	Number of CPU cores. Type: int
	`cpu-power`	CPU clock speed in MHz. Type: int
	`ram-size`	RAM size in MB. Type: int
	`bandwidth`	Network bandwidth in kbps. Type: int
	`login`	User login name. Type: string
	`template-id`	Used internally. Type: int
	`primary-disk-id`	Used internally. Type: int
	`image-size`	Image size, in gigabytes. Type: int
	`created`	Image creation date. Type: string
	`no-of-public-ip`	Number of public IPv4 addresses. Type: int
	`no-of-public-ipv6`	Number of public IPv6 addresses. Type: int
`description`		Image description. Type: string
`disks`		Container for disk information. This element may appear up to 10 times (one for each disk). *At the time of this writing only one disk can be added to a server. This will change in the future*.
	`id`	Disk database ID (used internally). Type: int
	`local`	Specifies whether this is a local or a network disk. *At the time of this writing only local disks are supported*. Type: boolean
	`primary`	This is a reserved parameter. Type: boolean

	`size`	Disk size in GB. Type: int

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/image/Web-Image-101
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-image no-of-public-ipv6="1" no-of-public-ip="1" created="2013-06-29
17:28:01.50428+04" image-size="1" primary-disk-id="0" template-id="1" login="root"
bandwidth="100" ram-size="512" cpu-power="1600" cpu-number="2" name="Web-Image-101"
image-of="Web40" subscription-id="1000001" customer-id="1000001" bnode-uuid="6" id="1">
    <description>Image of Web101</description>
    <disks size="1" primary="true" local="true" id="0"/>
</ve-image>
```

Create Image from Server

Description

Use this request to create an image from an existing server. The server must be stopped before you attempt to create an image from it. The `{ve-name}` part must contain the source server name. The `{image-name}` part must contain the desired image name. **Syntax 1** can only be used when the customer account to which this user belongs has a single subscription. In case of multiple subscriptions use **Syntax 2**.

Syntax 1

Use this syntax when the customer account to which this user belongs has just one subscription.

```
POST baseURL/image/{ve-name}/create/{image-name}
```

Syntax 2

Use this syntax when the customer account has multiple subscriptions. The `{subscription-id}` must contain the desired subscription ID.

```
POST baseURL/image/{ve-name}/{subscription-id}/create/{image-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example creates an image from the Web101 server for subscription 1000001.

Request

```
POST https://c2u-web:4465/paci/v1.0/image/Web101/1000001/create/Web-Image-101
```

Response

```
Image creation initiated
```

Delete Image

Description

Use this request to delete an existing server image.

Syntax

```
DELETE baseURL/image/{image-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/image/Web-Image-102
```

Response

```
Delete image initiated
```

Load Balancer Management

The requests described in this section are used to manage load balancers. To use load balancing, first create a load balancer and then attach the desired servers to it. You can use the PACI REST API to list existing load balancers, to create load balancers, to attach and detach servers to/from load balancers, and to delete load balancers.

List Load Balancers

Description

Use this request to obtain a list of the available load balancers.

Syntax

```
GET baseURL/load-balancer
```

Request Parameters

None

Response

Element	Attribute	Description
`lb-list`		Container for load balancer list.
`load-balancer`		Container for load balancer information. The element may appear more than once (one for each load balancer).
	`name`	Load balancer name. Type: string
	`subscription-id`	Subscription ID. Type: int
	`state`	Load balancer state (e.g. STARTED, etc). Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/load-balancer
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lb-list>
    <load-balancer state="STARTED" subscription-id="1" name="LB1"/>
    <load-balancer state="STARTED" subscription-id="1" name="LB2"/>
</lb-list>
```

Get Load Balancer Info

Description

Use this request to obtain the information about a specified load balancer. The `{lb-name}` part of the URL must contain the load balancer name.

Syntax

```
GET baseURL/load-balancer/{lb-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`load-balancer`		Container for load balancer information.
`id`		Load balancer database ID (used internally). Type: int
`uuid`		Server UUID. Type: string
`hnId`		ID of the hardware node on which this load balancer resides. Type: string
`customer-id`		ID of the customer account to which the load balancer belongs. Type: int
`name`		Server name. Type: string

`description`		Load balancer description. Type: string
`subscription-id`		ID of the subscription to which the load balancer belongs. Type: int
`cpu`		Container for CPU information.
	`number`	Number of CPU cores. Type: int
	`power`	CPU clock speed in megahertz. Type: int
`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`ve-disk`		Container for hard disk information. *At the time of this writing, a server can have only one hard disk. This will change in the future.*
	`id`	Disk database ID (used internally) Type: int
	`local`	Specified whether this is a local or a network disk. *Only local disks are supported at the the time of this writing.* Type: boolean
	`primary`	Specifies whether this is a system disk. This is a reserved parameter. Type: boolean
	`size`	Disk size in gigabytes. Type: int
	`created`	Disk status. Type: boolean Possible values: `true` — the disk creation process completed. `false` — the disk is being created and is not available for use.

`platform`		Container for operating system template and related information.
`template-info`		Container for OS template information.
	`name`	Template name. Type: string
	`vendorId`	Template vendor ID. Type: string
	`c2uId`	PACI template ID (used internally). Type: string
`os-info`		Container for operating system information. This information is a part of the OS template info.
	`type`	OS type (Linux. Windows, etc.) Type: string
	`technology`	Virtualization technology used. Type: string Possible values: `CT` - Virtuozzo Container, `VM` - Parallels virtual machine.
`network`		Container for network information.
	`private_ip`	Private IP address and mask. Type: string
`public-ip`		Container for public IPv4 address info. This element may appear more than once (one for each IP address).
	`id`	Database ID (used internally). Type: int
	`address`	IP address and mask. Type: string
	`gateway`	Gateway IP address. Type: string
	`chunk-ref`	Used internally. Type: int
`public-ipv6`		Container for public IPv6 address info. This element may appear more than once (one for each IPv6 address).
	`id`	Database ID (used internally). Type: int

	address	IPv6 address and mask. Type: string
	gateway	Gateway IPv6 address. Type: string
	ipv6-net-id	Used internally. Type: int
state		A string describing the load balancer state or transition. Type: string
primary-disk-id		Used internally. Type: int
template-id		Used internally. Type: int
admin		Container for the server administrator credentials.
	login	Server administrator login name. Type: string
	password	Password. Type: string
used-by		Information about a server attached to this load balancer. This element may appear more than once (one for each server).
	ve-name	Server name. Type: string
	ip	Server IP address. Type: string

Example

The following example obtains information for the load balancer named LB1.

Request

```
GET https://c2u-web:4465/paci/v1.0/load-balancer/LB1
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<load-balancer>
    <id>10</id>
    <uuid>_374f525b.1322595125d._7ffb</uuid>
    <hnId>4</hnId>
    <customer-id>1</customer-id>
    <name>LB1</name>
    <description>Created from image [lbimage]</description>
    <subscription-id>1</subscription-id>
    <cpu power="700" number="1"/>
    <ram-size>1024</ram-size>
    <bandwidth>1000</bandwidth>
    <ve-disk created="true" size="2" local="true" id="0"/>
    <platform>
        <template-info c2uId="1" vendorId="5" name="centos-5-x86"/>
        <os-info technology="CT" type="linux-free"/>
    </platform>
    <network private-ip="1.1.1.230/8">
        <public-ip chunk-ref="1" gateway="2.2.2.1" address="2.2.2.208/16" id="12"/>
    </network>
    <state>STARTED</state>
    <primary-disk-id>0</primary-disk-id>
    <template-id>1</template-id>
    <admin password="[hidden]" login="root"/>
    <used-by ip="1.1.1.224" ve-name="Web20"/>
    <used-by ip="1.1.1.225" ve-name="CT1"/>
    <used-by ip="1.1.1.226" ve-name="CT2"/>
</load-balancer>
```

Create Load Balancer

Description

Use this request to create a load balancer. The `{name}` part of the request URL must contain the user-defined load balancer name.

Syntax

```
POST baseURL/load-balancer/create/{name}
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response info.
`message`		A text message describing the status of the operation.
`password`		Automatically generated password.

Example

The following example creates a load balancer named LB100.

Request

```
POST https://c2u-web:4465/paci/v1.0/load-balancer/create/LB100
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>LB creation initiated</message>
    <password>668ttoAHL</password>
</pwd-response>
```

Create Load Balancer for Subscription

Description

Use this request to create a load balancer for a specified subscription. The current user may be a member of a customer account with more than one subscription. This request allows to create a load balancer for a specific subscription. The `{subscription-id}` part of the request URL must contain a valid subscription ID. The `{lb-name}` part must contain a user-defined load balancer name.

Syntax

```
POST baseURL/load-balancer/{subscription-id}/create/{lb-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response info.

message		A text message describing the status of the operation.
password		Automatically generated password.

Example

The following example creates a load balancer named LB100001 for subscription 100001.

Request

```
POST https://c2u-web:4465/paci/v1.0/load-balancer/100001/create/LB100001
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>LB creation initiated</message>
    <password>262uhdQEM</password>
</pwd-response>
```

Obtain Load Balancer History

Description

Use this request to obtain a load balancer history. The `{lb-name}` part of the request URL must contain the name of the load balancer for which to retrieve the history. The `{nrecords}` part must contain the number of records (from the end) to include in the result set.

Syntax

```
GET baseURL/load-balancer/{lb-name}/history/{nrecords}
```

Request Parameters

None

Response

Element	Attribute	Description
ve-history		Container for load balancer history.
ve-snapshot		Container for a single history record. This element may appear more than once (one for each record).
	cpu	CPU clock speed, in megahertz. Type: int
	ram	RAM size, in megabytes. Type: int

	`local-disk`	Local disk size, in gigabytes. Type: int
	`nbd`	*This parameter is reserved for future use*. Type: int
	`bandwidth`	Network bandwidth, in kbps. Type: int
	`last-touched-from`	Used internally. Type: string
	`state`	Server state right after the modification operation began executing. Type: string
	`last-changed-by`	Name of the user initiating the modification. Type: string
	`event-timestamp`	The modification event timestamp. Type: string
	`no-of-public-ip`	Number of public IPv4 addresses. Type: int
	`no-of-public-ipv6`	No of public IPv6 addresses. Type: int
	`is-lb`	Specifies whether this is a load balancer or a regular server: `true` — load balancer `false` — regular server Type: boolean
	`private-incoming-traffic`	Private incoming traffic, in bytes. Type: long
	`private-outgoing-traffic`	Private outgoing traffic, in bytes. Type: long
	`public-incoming-traffic`	Public incoming traffic, in bytes. Type: long
	`public-outgoing-traffic`	Public outgoing traffic, in bytes. Type: long

Example

The following example retrieves the last three records from load balancer LB101 history.

Request

```
GET https://c2u-web:4465/paci/v1.0/load-balancer/LB101/history/3
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-history>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2013-09-01 19:22:50.326641+04" last-changed-
by="admin" state="CREATION_IN_PROGRESS" last-touched-from="im1" bandwidth="1000"
nbd="0" local-disk="0" ram="1024" cpu="700"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="1" event-timestamp="2013-09-01 19:22:50.538825+04" last-changed-
by="InstanceManager" steady-state="CREATED" state="CREATED" last-touched-from="im1"
bandwidth="1000" nbd="0" local-disk="2" ram="1024" cpu="700"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="1" event-timestamp="2013-09-01 19:22:50.760722+04" last-changed-
by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-from="im1"
bandwidth="1000" nbd="0" local-disk="2" ram="1024" cpu="700"/>
</ve-history>
```

Restart Load Balancer

Description

Use this request to restart a load balancer.

Syntax

```
PUT baseURL/load-balancer/{lb-name}/restart
```

Request Parameters

None

Response

A text message describing the operation status.

Example

The following example restart a load balancer named LB101.

Request

```
PUT https://c2u-web:4465/paci/v1.0/load-balancer/LB101/restart
```

Response

```
LB restart initiated
```

Delete Load Balancer

Description

Use this request to delete an existing load balancer.

Syntax

```
DELETE baseURL/load-balancer/{lb-name}
```

Request Parameters

None

Response

A text message describing the status of the operation. If there are servers attached to a load balancer the response will contain error message: "P2000023: Load balancer is in use". In such a case the servers will have to be detached first.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/load-balancer/LB101
```

Response

```
LB removing initiated
```

Attach Server To Load Balancer

Description

Use this request to attach a server to a load balancer. Once this request is completed, the server load will be managed by the specified load balancer. The `{lb-name}` and `{ve-name}` parts of the request URL must contain the load balancer and the server names respectively.

Syntax

```
POST baseURL/load-balancer/{lb-name}/{ve-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example attaches the server named Web60 to the load balancer named LB101.

Request

```
POST https://c2u-web:4465/paci/v1.0/load-balancer/LB101/Web60
```

Detach Server From Load Balancer

Description

Use this request to detach a server from a load balancer. The `{lb-name}` and `{ve-name}` parts of the request URL must contain the load balancer and the server names respectively.

Syntax

```
DELETE baseURL/load-balancer/{lb-name}/{ve-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/load-balancer/LB101/Web60
```

Utilities

The request described in this section are used to obtain information required by other requests.

List Installed OS Templates

Use this call to obtain a list of the available operating system templates. An operating system template is a package which is used to create new servers. It contains a particular operating system type and version (and software applications in some cases) together with necessary instructions and is used to preconfigure a server and install the operating system into it. When creating a server, use this request to obtain a list of the available OS templates, then choose the template of interest and use its name as an input parameter in the server creation call (p. 16). The `{name}` part of the request URL is optional and may contain the OS template name. When it is included, only the information about the specified template will be retrieved.

Syntax

```
GET baseURL/template/{name}
```

Request Parameters

None

Response

Element	Attribute	Description
`template-list`		Container for template list.
`template`		Container for an individual template info. This element may appear more than once (one for each available OS template).
	`id`	Used internally.
	`name`	Template name. Use this value in API requests that expect a template identifier as an input parameter. Type: string
	`vendorId`	Template vendor ID. Type: string
	`c2uId`	Used internally.
	`osType`	Operating system type (Linux, Windows) Type: string

	`technology`	Specifies the virtualization technology used. Type: string Possible values: `CT` - Virtuozzo Container. `VM` - Parallels virtual machine.
	`active`	Specifies whether the template is active. Inactive templates cannot be used to create servers. Type: boolean
	`default`	Specifies whether this is the default template. Type: boolean
	`root-login`	Administrator login name (e.g. "root"). Type: string
	`min-hdd-size`	Minimum required hard disk space. Type: int
	`pwd-regex`	Used internally.
	`high-watermark-for-delivery`	Used internally.
	`low-watermark-for-delivery`	Used internally.
`option`		Additional template information in name/value pairs (e.g. the target CPU type and OS info). This element may appear more than once.
	`value`	Value. Type: string
	`name`	Name. Type: string

Examples

The following example retrieves the list of all available OS templates.

Request

```
GET https://c2u-web:4465/paci/v1.0/template
```

Response

Only a small portion of the actual OS template list is shown for brevity.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<template-list>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="1" root-login="root" default="true"
technology="CT" osType="linux-free" c2uId="1" vendorId="6" active="true" name="centos-
6-x86_64" id="1">
        <option value="x86_64" name="arch"/>
        <option value="centos" name="edition"/>
        <option value="English" name="lang"/>
    </template>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="10" root-login="root" default="true"
technology="VM" osType="linux-free" c2uId="1" vendorId="6" active="true" name="paci-
centos-6" id="2">
        <option value="x86_64" name="arch"/>
        <option value="centos" name="edition"/>
        <option value="English" name="lang"/>
    </template>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="20" root-login="Administrator"
default="true" technology="VM" osType="windows" c2uId="1" vendorId="2008" active="true"
name="paci-win2k8r2sp1" id="3">
        <option value="x86_64" name="arch"/>
        <option value="datacenter" name="edition"/>
        <option value="English" name="lang"/>
    </template>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="10" root-login="Administrator"
default="false" technology="VM" osType="windows" c2uId="1" vendorId="2008"
active="true" name="w2k8r2SP1x64_ja_dtc" id="7">
        <option value="x86_64" name="arch"/>
        <option value="datacenter" name="edition"/>
        <option value="Japan" name="lang"/>
    </template>

    ........
</template-list>
```

The following example retrieves the information about the centos-6-x86_64 operating system template.

Request

```
GET https://c2u-web:4465/paci/v1.0/template/centos-6-x86_64
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="1" root-login="root" default="true"
technology="CT" osType="linux-free" c2uId="1" vendorId="6" active="true" name="centos-
6-x86_64" id="1">
        <option value="x86_64" name="arch"/>
        <option value="centos" name="edition"/>
        <option value="English" name="lang"/>
</template>
```

List Backup Schedules

Use this call to obtain the list of the available backup schedules. Backup schedules are created by system administrator. If you would like to perform server backups on a regular basis, you can obtain the list of the available schedules using this call, then choose a schedule that suits your needs and specify its name when configuring your server.

Syntax

```
GET baseURL/schedule
```

Request Parameters

None

Response

Element	Attribute	Description
`backup-schedule-list`		Container for backup schedule list.
`backup-schedule`		Container for an individual backup schedule.
	`id`	Schedule ID. Type: int
	`name`	Schedule name. Type: string
	`cron-expression`	Standard CRON expression. Type: string
	`enabled`	Specifies whether the backup schedule is enabled. Disabled schedules cannot be assigned to servers. Type: boolean
	`backups-to-keep`	Maximum number of server backups to keep on the backup server. Type: int

	`no-of-incremental`	Number of incremental backups before a full backup is performed. Type: int

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/schedule
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<backup-schedule-list>
    <backup-schedule no-of-incremental="6" backups-to-keep="30" enabled="true" cron-
expression="0 0 12 * * ?" name="daily" id="1">
        <description>Daily</description>
    </backup-schedule>
    <backup-schedule no-of-incremental="5" backups-to-keep="24" enabled="true" cron-
expression="0 0 * * * ?" name="hourly" id="3">
        <description>hourly</description>
    </backup-schedule>
    <backup-schedule no-of-incremental="4" backups-to-keep="25" enabled="true" cron-
expression="0 0 6 ? * 7" name="weekly" id="2">
        <description>Weekly</description>
    </backup-schedule>
</backup-schedule-list>
```

Callbacks

PACI REST API provides a callback functionality that allows you to receive responses from PACI server asynchronously. Callbacks can be used when you initiate a time consuming operation on a PACI resource, such as starting a server, creating a server, and so forth. When using a callback, the PACI server will notify the caller when the operation is completed on the server side and will send back the success or failure information, which can then be processed as needed.

To use a callback, you will need to set up an HTTP server that will be listening for HTTP requests from the PACI server. The URL at which your server is listening for requests must then be included in the header of an HTTPS request that you are sending to the PACI server using the following format: `x-callback-url: <URL>`. For example:

```
x-callback-url: http://192.168.3.77:8081/bd6fsdb4-24sd-4a64
```

Once the operation is completed on the server side, the PACI server will send back an HTTP GET response to the URL that you provided in the original request. The response header will contain the following information:

Name	Value
`x-paci-rc`	0 (zero) if the the operation completed successfully (e.g. a server has been started). Any other value indicates that the operation has failed. In such a case, X-Error-Code and X-Error-Message (see below) will be included in the response and will contain the error code and the error message respectively.
`x-error-code`	In case of failure, contains the error code.
`x-error-message`	In case of failure, contains the error message.

When your HTTP server receives the response, it should send back a response back to PACI server as quickly as possible (without doing any additional processing). The response should contain the HTTP return code 200, which will tell the PACI server that you received the response. Until this happens, the PACI server will continue sending responses to your HTTP server at predefined intervals.

Examples

The following is a sample server creation request that provides a URL for the callback in its header (x-callback-url):

```
POST https://c2u-web:4465/paci/v1.0/ve
accept-encoding: identity
content-length: 913
content-type: application/xml
x-callback-url: http://192.168.3.77:8081/bd6fsdb4-24sd-4a64
authorization: <hidden>
<?xml version="1.0" ?>
  <ve>
    <name>test-lin-01</name>
    <description>Test_ve</description>
    <subscription-id>1</subscription-id>
    <cpu number="2" power="160"/>
    <ram-size>512</ram-size>
    <bandwidth>10</bandwidth>
    <no-of-public-ip>2</no-of-public-ip>
    <no-of-public-ipv6>2</no-of-public-ipv6>
    <ve-disk size="5"/>
    <platform>
      <template-info name="centos-6-x86_64"/>
    </platform>
  </ve>
```

You will receive an immediate response from the PACI server as usual, but once the server is created, your HTTP server will receive an HTTP GET response with a header similar to the following:

```
Host: 10.30.3.77:8081
x-paci-rc: 0
Connection: keep-alive
Accept: */*
User-Agent: NING/1.0
```

In the example above, the `x-paci-rc` value is 0, which indicates that the server has been created successfully.

Initiating VNC Session

A VNC client can be used to remotely control a server. To initiate a VNC session, you will need a VNC client installed on your computer. To connect to a server, you need to obtain the server's IP address, port number, and a password using the following syntax:

```
POST baseURL/ve/{ve-name}/console
```

The call will return a password to be used to establish a VNC connection in the following format:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <id>2</id>
    <message>Console configuration initiated</message>
    <password>4cnk2J0L</password>
</pwd-response>
```

The VNC initiation procedure will take some time. Once it is completed on the server side, the `ve/console` structure will be populated with the server connection information. You can use a callback to be notified when the initiation procedure is completed. To obtain the server connection information, use the `GET baseURL/ve/{ve-name}` call. The IP address and port number will be included in the `<console>` element:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve>
  ...
  <console>
    <address>192.168.1.5</address>
    <port>4468</port>
  </console>
  ...
</ve>
```

CHAPTER 6

Appendix

In This Chapter

Datetime Format

The PACI REST API uses the following datetime format in requests and responses. When composing a request that accepts a datetime value, format it according to these specifications.

Format

```
yyyy-MM-dd HH:mm Z
```

Specification

Format Character	Description	Example Values
yyyy	Year, four digits.	2014
MM	Month, two digits with a leading zero.	01 through 12
dd	Day of month, two digits with a leading zero.	01 through 31
HH	Hour using the 24-hour notation (military time).	00 through 24
mm	Minutes, two digits with a leading zero.	00 through 59
Z	Time zone using standard time zone abbreviations.	CET, JST, EST, MDT...

Example

```
2014-01-08 21:00 CET
```

Error Codes

This section contains the complete list of PACI REST API errors. Each error definition is presented using the following format:

```
name(error_nature, error_type, error_code, error_description)
```

where

- `name` is the name of the enumeration constant defining the error.
- `error_nature` is a value describing whether this is a permanent error (P) or transient error (T).
- `error_type` is the error type (category).
- `error_code` is a numeric error code.
- `error_description` is the error description.

To evaluate error codes in your own program, you need to construct a complete alphanumeric error code as it will be returned via the HTTP error page.

To construct the complete code:

4. Identify a numeric error code (the second parameter inside the parenthesis; e.g. 1, 2, 3 ,etc).
5. Read the error description next to the numeric code. Modify if needed for your own needs.
6. Resolve the nature of the error. If the error definition has the `Nature.TRANSIENT` as the first parameter, the nature is "T". If not, the nature is "P".
7. Resolve the numeric code of the error type using the following list:
   - `Type.RESOURCE_IN_USE = 2`
   - `Type.NO_RESOURCE = 3`
   - `Type.UNSUPPORTED_OPERATION = 4`
   - `Type.INVALID_MUTATION = 5`
   - `Type.NOT_NOW = 6`
   - `Type.MISSED_INFO = 7`
   - `Type.INVALID_DATA = 8`
   - `Type.INVALID_OPERATION = 9`
8. Multiply the error type (step 4) by 10000 (ten thousand) and add the error code (step 1) to it.
9. Insert the nature of the error (step 3) in front of the error code that you've constructed in step 5. The resulting string is a complete error code as it will appear on the HTTP error page.

Example

Let's say you want to evaluate the following error in your program:

```
BACKUP_NODE_ID_IN_USE(Type.RESOURCE_IN_USE, 1, "Backup node is in use")
```

The error code is 1. The nature of the code is "P". The error type is 2. Therefore, the complete error code is P + (2 * 10000 + 1) or P20001.

The following is the complete list of errors grouped by error type.

RESOURCE_IN_USE

```
BACKUP_NODE_ID_IN_USE(Type.RESOURCE_IN_USE, 1, "Backup node is in
use"),

NODE_IN_USE(Type.RESOURCE_IN_USE, 2, "Node is in use"),

NODE_HOSTS_VLANS(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 3, "Node still
hosts one or more VLANs. Please try again later"),

TEMPLATE_USED_BY_VE(Type.RESOURCE_IN_USE, 4, "Template is used by one
or more VE(s)"),

TEMPLATE_USED_BY_IMAGES(Type.RESOURCE_IN_USE, 5, "Template is used by
one or more VE image(s)"),

VE_BUSY(Type.RESOURCE_IN_USE, 6, "VE is busy"),

IMAGE_IN_USE(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 7, "Image is in
use. Please try again later"),

NG_USED_BY_VLANS(Type.RESOURCE_IN_USE, 8, "One or more VLAN(s) have
subnets from this netgroup"),

NG_USED_BY_NODES(Type.RESOURCE_IN_USE, 9, "One or more hardware node(s)
are associated with this netgroup"),

NG_PUBLIC_IP_IN_USE(Type.RESOURCE_IN_USE, 10, "One or more public IP
address(es) from this netgroup are in use"),

INCOMPATIBLE_SETTINGS(Type.RESOURCE_IN_USE, 11, "Template [%s] which is
in use on this node is not compatible with new settings", true),

APP_TEMPLATE_IS_IN_USE(Type.RESOURCE_IN_USE, 20, "Application template
is in use by one or more VEs"),

NG_PUBLIC_IPV6_IN_USE(Type.RESOURCE_IN_USE, 21, "One or more public
IPV6 address(es) from this netgroup are in use"),

NODE_IN_USE_BY_OPS(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 22, "There
are %d ongoing operations with this node. Please try again later",
true),

LB_IN_USE(Type.RESOURCE_IN_USE, 23, "Load balancer is in use"),

LB_BUSY(Type.RESOURCE_IN_USE, 24, "Load balancer is busy"),

TEMPLATE_USED_BY_HOSTED_VE(Type.RESOURCE_IN_USE, 25, "Template is used
by one or more VE(s) hosted on the node"),
```

```
APP_TEMPLATE_IS_INSTALLED(Type.RESOURCE_IN_USE, 26, "Application
template is installed on one or more nodes"),

TEMPLATE_DELIVERIES_IN_PROGRESS(Type.RESOURCE_IN_USE, 27, "One or more
template deliveries are in progress"),

APP_TEMPLATE_IS_IN_USE_BY_IMAGE(Type.RESOURCE_IN_USE, 28, "Application
template is in use by one or more VE images"),

NODE_IS_TEMPLATE_SOURCE(Type.RESOURCE_IN_USE, 29, "The node is a source
for active template delivery. Please try again later"),

IMAGE_DELETE_IN_PROGRESS(Type.RESOURCE_IN_USE, 30, "Image delete is in
progress."),

OBJECT_IS_BUSY(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 31, "Another
operation in progress. Please try again later"),

BACKUP_IN_PROGESS(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 32, "VE
backup in progress. Please try again later"),

HN_OPERATIONS_IN_PROGESS(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 33,
"There are %s ongoing operations with this node. Please try again
later", true),

HN_RUNNING_VE(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 34, "There are %s
started VEs on this node. Please try again later", true),

ONGOING_TRAFFIC_REQUESTS(Type.RESOURCE_IN_USE, 35, "There are ongoing
traffic requests for this node. Please try again later"),

APPLICATIONS_FOR_OS_EXIST(Type.RESOURCE_IN_USE, 36, "There are %s
registered applications for this template", true),

HN_NOT_READY(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 37, "Hardware not
is not initialized yet"),

IMAGE_IS_NOT_CREATED(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 38, "Image
creation is still in progress"),

IMAGE_MIGRATION_IN_PROGRESS(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 39,
"Image migration(s) to this node is/are in progress"),

IMAGES_IN_USE_ON_BNODE(Nature.TRANSIENT, Type.RESOURCE_IN_USE, 40, "One
or more images are in use on this node"),
```

NO_RESOURCE

```
NO_SUITABLE_BN(Type.NO_RESOURCE, 1, "Cannot find suitable backup/image
node"),
```

```
NO_SUITABLE_HN(Type.NO_RESOURCE, 2, "Cannot find suitable hardware
node"),

HN_FULL(Type.NO_RESOURCE, 3, "Hardware node resource limits exceeded"),

NO_PUBLIC_IPS(Type.NO_RESOURCE, 4, "Cannot assign required number of
public IPs"),

CANNOT_ALLOCATE_NBD(Type.NO_RESOURCE, 5, "Cannot allocate remote
disk"),

NO_DEFAULT_TEMPLATE(Type.NO_RESOURCE, 6, "No default template was found
for specified os/technology"),

TEMPLATE_NOT_FOUND(Type.NO_RESOURCE, 7, "Requested template not
found"),

NO_PUBLIC_IPV6(Type.NO_RESOURCE, 8, "Cannot assign required number of
public IP V6 addresses"),

CANNOT_ALLOCATE_PRIVATE_IP(Type.NO_RESOURCE, 9, "Cannot allocate
private IP"),

CANNOT_FIND_SUITABLE_NETGROUP(Type.NO_RESOURCE, 10, "Cannot find
suitable netgroup"),

NO_SUITABLE_HN_FOR_RESIZE(Type.NO_RESOURCE, 11, "Cannot find suitable
hardware node for resized VE"),

HN_IS_NOT_ACTIVE(Type.NO_RESOURCE, 20, "Hardware node [%s] is locked or
inactive", true),

HN_VE_LIMIT_EXCEEDED(Type.NO_RESOURCE, 21, "Hardware node VE limit
exceeded"),

HN_CPU_LIMIT_EXCEEDED(Type.NO_RESOURCE, 22, "Hardware node CPU limit
exceeded"),

HN_RAM_LIMIT_EXCEEDED(Type.NO_RESOURCE, 23, "Hardware node RAM limit
exceeded"),

HN_DISK_LIMIT_EXCEEDED(Type.NO_RESOURCE, 24, "Hardware node disk limit
exceeded"),

HN_BANDWIDTH_LIMIT_EXCEEDED(Type.NO_RESOURCE, 25, "Hardware node
bandwidth limit exceeded"),

HN_CPU_NUMBER_LIMIT_EXCEEDED(Type.NO_RESOURCE, 26, "Hardware node has
less number of CPU than required"),
```

```
HN_CPU_POWER_LIMIT_EXCEEDED(Type.NO_RESOURCE, 27, "Hardware node has
less powerful CPU than required"),

HN_IS_INACTIVE(Type.NO_RESOURCE, 28, "Hardware node is inactive"),

HN_PCS_LIMIT_EXCEEDED(Type.NO_RESOURCE, 29, "Hardware node pcs storage
limit exceeded"),

NO_FREE_SUBNETS_IN_NG(Type.NO_RESOURCE, 30, "Netgroup [%s] doesn't have
free private subnets", true),

NO_PRIVATE_IP_IN_NG(Type.NO_RESOURCE, 31, "Netgroup [%s] doesn't have
free private IPs", true),

NO_PUBLIC_IP_IN_NG(Type.NO_RESOURCE, 32, "Netgroup [%s] doesn't have
enough free public IPs", true),

NO_PUBLIC_IPV6_IN_NG(Type.NO_RESOURCE, 33, "Netgroup [%s] doesn't have
enough free public IP V6 addresses", true),

HN_TEMPLATE_MISMATCH(Type.NO_RESOURCE, 40, "Template [%s] is not
compatible with the hardware node [%s]: %s", true),

BN_IS_INACTIVE(Type.NO_RESOURCE, 50, "Backup node [%s] is inactive",
true),

BN_IS_NOT_ACTIVE(Type.NO_RESOURCE, 51, "Backup node [%s] is not
active", true),
```

UNSUPPORTED_OPERATION

```
UNSUPPORTED_IMAGE_CREATION(Type.UNSUPPORTED_OPERATION, 1, "Image
creation is not supported for VEs with %s disks", true),

UNSUPPORTED_CREATION_FROM_IMAGE(Type.UNSUPPORTED_OPERATION, 2,
"Creating VE from images containing %s disks is not currently
supported", true),

UNSUPPORTED_CLONING(Type.UNSUPPORTED_OPERATION, 3, "Cloning VE
containing %s disks is not currently supported", true),
```

INVALID_MUTATION

```
NODE_NAME_CHANGE_PROHIBITED(Type.INVALID_MUTATION, 1, "Node name can
not be changed"),

NETGROUP_CHANGE_PROHIBITED(Type.INVALID_MUTATION, 2, "Netgroup can not
be changed"),
```

```
NODE_IP_CHANGE_PROHIBITED(Type.INVALID_MUTATION, 3, "Node network
location can not be changed"),

NG_LABEL_CANNOT_BE_CHANGED(Type.INVALID_MUTATION, 4, "Netgroup label
cannot be changed"),

ONLY_PRIMARY_LOCAL_DISK_CAN_BE_CHANGED(Type.INVALID_MUTATION, 5, "Only
primary local disk size can be changed"),

CANNOT_CHANGE_VE_NAME(Type.INVALID_MUTATION, 6, "VE name cannot be
changed"),

CANNOT_REDUCE_BANDWIDTH(Type.INVALID_MUTATION, 7, "Bandwidth setting is
below the current usage: %s", true),

CANNOT_REDUCE_DISK(Type.INVALID_MUTATION, 8, "Disk setting is below the
current usage: %s", true),

CANNOT_REDUCE_RAM(Type.INVALID_MUTATION, 9, "RAM setting is below the
current usage: %s", true),

CANNOT_REDUCE_CPU(Type.INVALID_MUTATION, 10, "CPU settings
(number*power) are below the current usage: %s", true),

CANNOT_REDUCE_VE_LIMIT(Type.INVALID_MUTATION, 11, "VE limit settings
are below the current usage: %s", true),

CANNOT_CHANGE_USER_ID(Type.INVALID_MUTATION, 12, "User Id cannot be
changed"),

CANNOT_CHANGE_USER_ASSOCIATION(Type.INVALID_MUTATION, 13,
"User/Customer association cannot be changed"),

INVALID_NODE_STATE_MUTATION(Type.INVALID_MUTATION, 14, "Node state
cannot be changed this way"),

CANNOT_REDUCE_CPU_NUMBER(Type.INVALID_MUTATION, 15, "CPU settings
(number) are below the current usage: %s", true),

CANNOT_REDUCE_CPU_POWER(Type.INVALID_MUTATION, 16, "CPU settings
(power) are below the current usage: %s", true),

CANNOT_CHANGE_SCHEDULE_NAME(Type.INVALID_MUTATION, 17, "Backup schedule
name cannot be changed"),

CANNOT_CHANGE_NETGROUP(Type.INVALID_MUTATION, 18, "Netgroup can not be
changed as the node hosts one or more VE(s)"),

CANNOT_CHANGE_VE_IN_STATE(Type.INVALID_MUTATION, 19, "%s cannot be
changed in the %s state", true),
```

```
CANNOT_REDUCE_PRIMARY_DISK_SIZE(Type.INVALID_MUTATION, 20, "Primary
disk size cannot be decreased"),
```

MISSED_INFO

```
NO_DATA(Type.MISSED_INFO, 1, "No data"),

NODE_NAME_MISSED(Type.MISSED_INFO, 2, "Node name must be specified"),

NODE_ACCESS_INFO_MISSED(Type.MISSED_INFO, 3, "Node network location
must be specified"),

NODE_IP_ADDRESS_MISSED(Type.MISSED_INFO, 4, "Node IP address is not
specified"),

NODE_TAGS_MISSED(Type.MISSED_INFO, 5, "Node w/o tags is unusable"),

NETGROUP_MISSED(Type.MISSED_INFO, 6, "Either id or label if the node's
netgroup must be specified"),

HARDWARE_INFO_MISSED(Type.MISSED_INFO, 7, "Node hardware must be
specified"),

CPU_INFO_MISSED(Type.MISSED_INFO, 8, "Node CPU specification is
missed"),

RAM_INFO_MISSED(Type.MISSED_INFO, 9, "Node RAM specification is
missed"),

HDD_INFO_MISSED(Type.MISSED_INFO, 10, "Node HDD specification is
missed"),

VE_NAME_MISSED(Type.MISSED_INFO, 11, "VE name must be specified"),

VE_CPU_MISSED(Type.MISSED_INFO, 12, "Required CPU must be specified"),

VE_RAM_MISSED(Type.MISSED_INFO, 13, "Required RAM must be specified"),

VE_DISK_MISSED(Type.MISSED_INFO, 14, "At least one disk must be
specified"),

NG_LABEL_MISSED(Type.MISSED_INFO, 15, "Netgroup label must be
specified"),

CRON_EXPRESSION_MISSED(Type.MISSED_INFO, 16, "Cron expression must be
specified"),

VE_PRIMARY_DISK_MISSED(Type.MISSED_INFO, 17, "At least one local or
primary disk must be specified"),
```

```
NO_PLATFORM_INFO(Type.MISSED_INFO, 18, "Either template or os info must
be specified"),

NO_BANDWIDTH_INFO(Type.MISSED_INFO, 19, "Required network bandwidth
must be specified and be positive"),

NO_SUBSCRIPTION_FOUND(Type.MISSED_INFO, 20, "No subscription found"),

MORE_THAN_ONE_SUBSCRIPTION_FOUND(Type.MISSED_INFO, 21, "Subscription is
not specified and more than one subscription found"),

NG_DNS_SERVERS_MISSED(Type.MISSED_INFO, 22, "At least one DNS server
must be specified"),

MAX_BACKUPS_TO_KEEP_MISSED(Type.MISSED_INFO, 23, "'backups-to-keep'
must be specified"),

NO_PCS_CLUSTER_NAME(Type.MISSED_INFO, 30, "PCS cluster name must be
specified"),

NO_CALLBACK_URI(Type.MISSED_INFO, 40, "Please provide callback URL"),
```

INVALID_OPERATION

```
CANNOT_DELETE_DEFAULT_TEMPLATE(Type.INVALID_OPERATION, 1, "Cannot
delete default template"),

INVALID_METHOD_FOR_VE_OPERATION(Type.INVALID_OPERATION, 2, "Operation
is not allowed in this method"),

INVALID_TARGET_STATE_FOR_DEACTIVATION(Type.INVALID_OPERATION, 10,
"Cannot DEACTIVATE iSCSI target which is not LOCKED"),

INVALID_TARGET_STATE_FOR_DELETION(Type.INVALID_OPERATION, 11, "Cannot
DELETE iSCSI target which is not LOCKED or INACTIVE"),

BUSY_TARGET(Type.INVALID_OPERATION, 12, "Cannot DEACTIVATE/DELETE iSCSI
target as it has LUNs which are in use"),

APP_TEMPLATE_IS_ACTIVE(Type.INVALID_OPERATION, 20, "Cannot unregister
active template"),

INVALID_APP_TEMPLATE_OPERATION(Type.INVALID_OPERATION, 21, "Invalid
operation. Please use ACTIVATE/DEACTIVATE (case-insensitive)"),

APP_TEMPLATE_NAME_CHANGE_PROHIBITED(Type.INVALID_MUTATION, 22,
"Application template name can not be changed"),

APP_TEMPLATE_FOROS_CHANGE_PROHIBITED(Type.INVALID_MUTATION, 23,
"Application template OS can not be changed"),
```

```
APP_TEMPLATE_NOT_INSTALLED(Type.INVALID_OPERATION, 24, "Application
template is not installed"),

FEATURE_NOT_SUPPORTED(Type.INVALID_OPERATION, 30, "Feature %s is not
supported in this installation", true),

LB_ALREADY_ATTACHED(Type.INVALID_OPERATION, 40, "VE is already attached
to a load balancer"),

NO_LB_ATTACHED(Type.INVALID_OPERATION, 41, "VE is not attached to a
load balancer"),

ANOTHER_LB_ATTACHED(Type.INVALID_OPERATION, 42, "VE is attached to
another load balancer"),

NOT_AN_LB(Type.INVALID_OPERATION, 43, "Image %s is not a load
balancer", true),

INVALID_LB_TECHNOLOGY(Type.INVALID_OPERATION, 44, "Only CT images are
supported as LB ones"),

LB_ATTACHED(Type.INVALID_OPERATION, 45, "Cannot delete VE attached to a
load balancer"),

NO_PUBLIC_IPS_FOR_LB(Type.INVALID_OPERATION, 46, "Image w/o piblic IPs
cannot be registered as LB image"),

INVALID_VE_TECHNOLOGY(Type.INVALID_OPERATION, 47, "VE technology is not
valid for operation"),

SUBSCRIPTION_HAS_VE(Type.INVALID_OPERATION, 50, "There are still %s VEs
for this subscription", true),

SUBSCRIPTION_HAS_IMAGES(Type.INVALID_OPERATION, 51, "There are still %s
images for this subscription", true),

CUSTOMER_HAS_SUBSCRIPTIONS(Type.INVALID_OPERATION, 52, "There are still
exists %s subscription(s) for this customer", true),

CUSTOMER_HAS_USERS(Type.INVALID_OPERATION, 53, "There are still exists
%s user(s) for this customer", true),

CUSTOMER_HAS_VLANS(Type.INVALID_OPERATION, 54, "There are still exists
%s VLAN(s) for this customer", true),

SAME_NODE_FOR_MIGRATION(Type.INVALID_OPERATION, 60, "Cannot migrate VE
to the same node"),

PLEASE_LOCK_HN(Type.INVALID_OPERATION, 61, "Hardware node should be in
a LOCKED state for this operation"),
```

```
LOCATION_HAS_SUBSCRIPTIONS(Type.INVALID_OPERATION, 70, "There are %s
subscriptions for this location", true),

LOCATION_HAS_HNS(Type.INVALID_OPERATION, 71, "There are %s
virtualization nodes for this location", true),

LOCATION_HAS_BNS(Type.INVALID_OPERATION, 72, "There are %s storage
nodes for this location", true),

LOCATION_EXISTS(Type.INVALID_OPERATION, 73, "Location [%s] already
exists", true),

IMAGE_MIGRATION_ALREADY_STARTED(Type.INVALID_OPERATION, 74, "Image
migration is already in progress"),

PCS_CLUSTER_IN_USE(Type.INVALID_OPERATION, 80, "PCS cluster [%s] is
used by at least one HN", true),

PCS_CLUSTER_IN_USE_BY_VES(Type.INVALID_OPERATION, 81, "HN's PCS cluster
is used by at least one hosted VE"),
```

NOT_NOW

```
INVALID_VE_STATE_FOR_IMAGE_CREATION(Type.NOT_NOW, 1, "Cannot create
image while VE is in the current state [%s]", true),

OPERATION_INVALID_FOR_VE_STATE(Type.NOT_NOW, 2, "This operation is
invalid for the current VE state: %s", true),

OPERATION_INVALID_FOR_LB_STATE(Type.NOT_NOW, 3, "This operation is
invalid for the current LB state: %s", true),

OPERATION_INVALID_FOR_RUNNING_STATE(Type.NOT_NOW, 4, "This operation is
invalid for the STARTED VE state"),
```

INVALID_DATA

```
INVALID_REQUEST(Type.INVALID_DATA, 1, "Invalid request: %s", true),

CANNOT_PARSE_REQUEST(Type.INVALID_DATA, 2, "Cannot parse request: %s",
true),

REQUEST_UNMARSHALLING_FAILED(Type.INVALID_DATA, 3, "Failed to unmarshal
the request"),

INVALID_NODE_STATE_SPECIFIED(Type.INVALID_DATA, 10, "Invalid state
name. Please use ACTIVE, LOCKED or INACTIVE (case-insensitive)"),

UNKNOWN_VE_OPERATION(Type.INVALID_DATA, 11, "Unknown VE operation"),
```

```
CRON_EXPRESSION_INVALID(Type.INVALID_DATA, 12, "Invalid cron
expression"),

PWD_REGEX_INVALID(Type.INVALID_DATA, 13, "Invalid password regex"),

INVALID_TIME_INTERVAL(Type.INVALID_DATA, 20, "'from' must be less than
'to'"),

INVALID_DATE_FORMAT(Type.INVALID_DATA, 21, "Bad %s date format. Should
be '" + DATE_FORMAT +"'", true),

TEMPLATE_OS_MISMATCH(Type.INVALID_DATA, 30, "Requested template/os
mismatch"),

TEMPLATE_IS_INACTIVE(Type.INVALID_DATA, 31, "Requested template is
inactive"),

INVALID_RAM_SIZE(Type.INVALID_DATA, 40, "VE RAM size has to be positive
and a multiple of 4"),

VE_PRIMARY_DISK(Type.INVALID_DATA, 41, "At most one primary disk can be
specified"),

VE_PRIMARY_DISK_TOO_SMALL(Type.INVALID_DATA, 42, "Primary disk size for
this configuration, cannot be less than %sGb", true),

INVALID_NODE_RAM_SIZE(Type.INVALID_DATA, 43, "Node RAM size has to be
positive"),

INVALID_OPTION_NAME(Type.INVALID_DATA, 50, "Invalid option name: [%s]",
true),

NOT_A_MULTI_VALUED_OPTION(Type.INVALID_DATA, 51, "Option [%s] can be
specified only once", true),

DUPLICATE_OPTION(Type.INVALID_DATA, 52, "Duplicate option [%s]", true),

INVALID_WATERMARKS(Type.INVALID_DATA, 53, "Low watermark must be less
than high one"),

BAD_LIMITS(Type.INVALID_DATA, 54, "MinIp >= MaxIp for public net: %s",
true),

BOUNDS_NOT_IN_PRIVATE_NET(Type.INVALID_DATA, 60, "Private net bounds
are not in the enclosing private network: %s", true),

BOUNDS_NOT_IN_SUBNET(Type.INVALID_DATA, 61, "Subnet bounds are not in
the same subnet: %s", true),
```

```
BAD_GATEWAY(Type.INVALID_DATA, 62, "Gateway is not in subnet for public
net: %s", true),

TOO_WIDE_SUBNET(Type.INVALID_DATA, 63, "Private subnet is too wide"),

INVALID_MASKS(Type.INVALID_DATA, 64, "Invalid values for masks; private
net: %s", true),

INVALID_SUBNET_BOUNDS(Type.INVALID_DATA, 65, "Invalid subnet bounds:
%s", true),

TOO_SMALL_IPV6_SUBNET(Type.INVALID_DATA, 66, "IP V6 subnet is too
small"),

INVALID_IP_ADDRESS(Type.INVALID_DATA, 67, "Invalid IP address: [%s]",
true),

DUPLICATE_DNS_IP_ADDRESS(Type.INVALID_DATA, 68, "Duplicate DNS IP
address: [%s]", true),

NON_CIDR_PUBLIC_V6(Type.INVALID_DATA, 69, "Public IP V6 subnet is not
CIDR-compliant: [%s]", true),

BAD_LIMIT(Type.INVALID_DATA, 70, "Min/Max IP [%s] is not in subnet for
public net: %s", true),

INVALID_TARGET_STATE_SPECIFIED(Type.INVALID_DATA, 71, "Invalid state
name. Please use ACTIVE, LOCKED or INACTIVE (case-insensitive)"),

INVALID_NO_OF_BLOCK_UNIT(Type.INVALID_DATA, 72, "Invalid no_of_blocks
unit: %s", true),

NON_IP_V4_ADDRESS(Type.INVALID_DATA, 73, "IP address is not IP V4-
compliant: [%s]", true),

NON_CIDR_V4(Type.INVALID_DATA, 74, "IP V4 subnet is not CIDR-compliant:
[%s]", true),

NON_IP_V6_ADDRESS(Type.INVALID_DATA, 75, "IP address is not IP V6-
compliant: [%s]", true),

INVALID_NETWORK(Type.INVALID_DATA, 76, "Invalid network: [%s]", true),

DUPLICATED_FIREWALL_RULE(Type.INVALID_DATA, 77, "Duplicated firewall
rule name: [%s]", true),

NON_CIDR_IP(Type.INVALID_DATA, 78, "IP subnet is not CIDR-compliant:
[%s]", true),
```

```
NO_OS_TEMPLATE_FOR_APP_TEMPLATE(Type.INVALID_DATA, 81, "OS Template is
not specified"),

INVALID_OS_TEMPLATE(Type.INVALID_DATA, 82, "INVALID OS TEMPLATE: %s",
true),

UNKNOWN_OS_TEMPLATE(Type.INVALID_DATA, 83, "UNKNOWN OS TEMPLATE: %s",
true),

INCOMPATIBLE_APPLICATION(Type.INVALID_DATA, 84, "Application template
is incompatible with VE template"),

APPLICATION_ALREADY_INSTALLED(Type.INVALID_DATA, 85, "Application is
already installed"),

INVALID_TEMPLATE_STATE(Type.INVALID_DATA, 86, "Invalid template state
name. Please use ACTIVE or INACTIVE (case-insensitive)"),

COLON_IN_USER_NAME(Type.INVALID_DATA, 90, "Colon is not allowed in
userid (RFC 2617)"),

TEMPLATE_ALREADY_REGISTERED(Type.INVALID_DATA, 100, "Template [%s] is
already registered at the node [%s]", true),

TEMPLATE_IS_NOT_REGISTERED(Type.INVALID_DATA, 101, "Template [%s] is
not registered at the node [%s]", true),

NO_NODE_WITH_TEMPLATE(Type.INVALID_DATA, 102, "Cannot find active node
with registered template [%s] at location[%s]", true),

UNKNOWN_NODE(Type.INVALID_DATA, 103, "Unknown hardware node [%s]",
true),

INVALID_TEMPLATE_DELETE_MODE(Type.INVALID_DATA, 104, "Invalid template
delete mode, valid modes are: 'unregister', 'purge' or 'please' (case-
insensitive) "),

NO_TEMPLATE_ON_NODE(Type.INVALID_DATA, 105, "Target node does not have
required OS template and on demand template deliver is disabled"),

NULL_IP_POOL_NAME(Type.INVALID_DATA, 120, "IP Pool name cannot be
null"),

DUPLICATE_PRIVATE_IP_POOL_LABEL(Type.INVALID_DATA, 121, "Private IP
pool [%s] is already registered", true),

DUPLICATE_PUBLIC_IP_POOL_LABEL(Type.INVALID_DATA, 122, "Public IP pool
[%s] is already registered", true),
```

```
DUPLICATE_PUBLIC_IPV6_POOL_LABEL(Type.INVALID_DATA, 123, "Public IPv6
pool [%s] is already registered", true),

NO_POOL_TYPE(Type.INVALID_DATA, 124, "Please specify IP pool type"),

INVALID_POOL_TYPE(Type.INVALID_DATA, 125, "Invalid pool type. Please
specify %s (case insensitive)", true),

INVALID_NO_OF_CPU(Type.INVALID_DATA, 200, "Invalid value for the number
of CPU"),

INVALID_CPU_POWER(Type.INVALID_DATA, 201, "Invalid value for the CPU
power"),

INVALID_BANDWIDTH(Type.INVALID_DATA, 202, "Invalid value for the
bandwidth"),

INVALID_ADD_IP(Type.INVALID_DATA, 204, "Invalid value for the <add-ip>
'number' attribute"),

INVALID_IP_RECONFIGURATION(Type.INVALID_DATA, 205, "Either <add-ip> or
<drop-ip> can be specified (not both)"),

INVALID_DISK_RECONFIGURATION(Type.INVALID_DATA, 206, "One (and only
one) of the <add-local-disk>/<drop-local-disk>/<resize-local-disk>
operations can be specified"),

INVALID_PRIMARY_DISK(Type.INVALID_DATA, 207, "Primary disk size must be
positive"),

TOO_MANY_CPU_CORES(Type.INVALID_DATA, 208, "Cannot increase number of
CPUs that much"),

TOO_HIGH_CPU_FREQ(Type.INVALID_DATA, 209, "Cannot increase CPU power
that much"),

UNKNOWN_IP(Type.INVALID_DATA, 210, "Unknown IP: [%s]", true),

UNKNOWN_LOCAL_DISK(Type.INVALID_DATA, 300, "No local disk with vm2vfId
%s found", true),

INVALID_DISK_SIZE(Type.INVALID_DATA, 301, "Disk size should be
positive", true),

INVALID_PUBLIC_IP_LIMIT(Type.INVALID_DATA, 310, "Public IP limit cannot
be negative"),

INVALID_PUBLIC_IPV6_LIMIT(Type.INVALID_DATA, 311, "Public IP V6 limit
cannot be negative"),
```

```
DUPLICATE_TECHNOLOGY(Type.INVALID_DATA, 312, "Duplicated technology:
%s", true),

UNKNOWN_OS_TYPE(Type.INVALID_DATA, 313, "Unknown OS type: %s", true),

DUPLICATE_OS_TYPE(Type.INVALID_DATA, 314, "Duplicate OS type: %s",
true),

INVALID_VE_LIMIT(Type.INVALID_DATA, 315, "VE limit cannot be
negative"),

CANNOT_REDUCE_PUBLIC_IP_LIMIT(Type.INVALID_DATA, 316, "Cannot reduce
public IP limit below the current usage = %s", true),

CANNOT_REDUCE_PUBLIC_IPV6_LIMIT(Type.INVALID_DATA, 317, "Cannot reduce
public IP V6 limit below the current usage = %s", true),

TECHNOLOGY_IS_USED(Type.INVALID_DATA, 318, "There are %s VEs for this
subscription with technology %s", true),

OS_TYPE_IS_USED(Type.INVALID_DATA, 319, "There are %s VEs for this
subscription with OS type [%s]", true),

TOO_MANY_PUBLIC_IP(Type.INVALID_DATA, 320, "Subscription limit for
public IP exceeded"),

TOO_MANY_PUBLIC_IPV6(Type.INVALID_DATA, 321, "Subscription limit for
public IP V6 exceeded"),

TOO_MANY_VE_FOR_OS_TYPE(Type.INVALID_DATA, 322, "Subscription limit for
VE with os type = [%s] exceeded", true),

NO_TECHNOLOGY_IN_SUBSCRIPTION(Type.INVALID_DATA, 323, "Subscription
does not allow VE with technology = [%s]", true),

CANNOT_REDUCE_VE_LIMIT_FOR_SUBSCRIPTION(Type.INVALID_DATA, 324, "There
are %s VEs for this subscription", true),

TOO_MANY_VE(Type.INVALID_DATA, 325, "Subscription limit for VE number
exceeded"),

NO_SPACE_FOR_BACKUPS(Type.INVALID_DATA, 326, "Subscription limit for
backup space exceeded"),

NO_SPACE_FOR_IMAGES(Type.INVALID_DATA, 327, "Subscription limit for
image space exceeded"),

INVALID_BACKUP_IMAGE_LIMIT(Type.INVALID_DATA, 328, "Backup/image limit
cannot be negative"),
```

```
SUBSCRIPTION_IS_DISABLED(Type.INVALID_DATA, 330, "Subscription %s is
disabled", true),

INVALID_SUBSCRIPTION_STATUS(Type.INVALID_DATA, 331, "Invalid
subscription status, please use enable/disable (case-insensitive)"),

INCONSITENT_IMPERSONATION(Type.INVALID_DATA, 332, "Impersonated
customerId doesn't match the one specified in subscription"),

CROSS_LOCATION(Type.INVALID_DATA, 333, "Source and target subscriptions
have different 'location' attribute values"),

CANNOT_REDUCE_BACKUP_LIMIT(Type.INVALID_DATA, 334, "Cannot reduce
backup limit below the current usage = %s", true),

CANNOT_REDUCE_IMAGE_LIMIT(Type.INVALID_DATA, 335, "Cannot reduce image
limit below the current usage = %s", true),

BN_HOLDS_BACKUPS(Type.INVALID_DATA, 400, "Cannot revoke 'for-backups'
as node already hosts one(s)"),

BN_HOLDS_IMAGES(Type.INVALID_DATA, 401, "Cannot revoke 'for-images' as
node already hosts one(s)"),

INVALID_CAPACITY(Type.INVALID_DATA, 402, "Node capacity should be
positive"),

UNKNOWN_TAG(Type.INVALID_DATA, 403, "Unknown node tag id : %s", true),

UNKNOWN_OPTION(Type.INVALID_DATA, 404, "Unknown option: [%s] = [%s]",
true),

DUPLICATE_OPTION_VALUE(Type.INVALID_DATA, 405, "Duplicate option=value
: [%s] = [%s]", true),

TOO_LONG_STRING(Type.INVALID_DATA, 410, "String value '%s' should be
not longer than %s chars", true),

INVALID_NODE_FILTER(Type.INVALID_DATA, 411, "Invalid node filter.
Please use %s (case-insensitive", true),

INVALID_CHARACTER_IN_VE_NAME(Type.INVALID_DATA, 412, "VE name contains
non RFC 3490 - compliant characters"),

NON_LDH_CHARACTER_IN_VE_NAME(Type.INVALID_DATA, 413, "VE name contains
non-LDH characters"),

VE_NAME_HAS_HYPHEN(Type.INVALID_DATA, 414, "VE name has leading or
trailing hyphen"),
```

```
DOT_IN_VE_NAME(Type.INVALID_DATA, 415, "VE name cannot contain dots"),

TOO_LONG_VE_NAME(Type.INVALID_DATA, 416, "VE name cannot be longer than
%d characters", true),

NULL_STRING(Type.INVALID_DATA, 417, "String value '%s' cannot be null",
true),

TOO_LONG_HOSTNAME(Type.INVALID_DATA, 418, "Hostname cannot be longer
than %d characters", true),

BN_CANNOT_HOST_IMAGES(Type.INVALID_DATA, 420, "The node cannot host
images"),

BN_IS_INCOMPATIBLE_WITH_TEMPLATE(Type.INVALID_DATA, 421, "The node is
not compatible with image template"),

NO_ROOM_FOR_IMAGE(Type.INVALID_DATA, 422, "Image size exceeds the
available node capacity"),

TOO_WIDE_IPV6_SUBNET(Type.INVALID_DATA, 430, "IP V6 subnet is too
wide"),

INVALID_PRIVATE_POOL(Type.INVALID_DATA, 431, "Private subnet %s
overlaps with the already existing one", true),

INVALID_PUBLIC_POOL(Type.INVALID_DATA, 432, "Public subnet %s overlaps
with the already existing one", true),

INVALID_CLUSTER_CAPACITY(Type.INVALID_DATA, 440, "PCS cluster capacity
should be non-negative"),

CLUSTER_ALREADY_IN_USE(Type.INVALID_DATA, 441, "PCS cluster [%s] is
already registered", true),

DUPLICATED_STORAGE_DEFINITION(Type.INVALID_DATA, 442, "Duplicated
storage definition: %s, '%s', %s", true),

DUPLICATED_STORAGE_DEFAULT(Type.INVALID_DATA, 443, "Duplicated storage
default for: %s, '%s'", true),

UNKNOWN_PCS__CLUSTER(Type.INVALID_DATA, 444, "Unknown PCS cluster:
[%s]", true),

INVALID_STORAGE__TYPE(Type.INVALID_DATA, 445, "Storage type [%s] is not
allowed for the subscription", true),

DISK_STORAGE_TYPE_IS_NOT_SUPPORTED(Type.INVALID_DATA, 446, "Storage
type [%s] is not supported in this version of product", true),
```

```
MULTIPLE_DISKS_OF_STORAGE_TYPE_ARE_NOT_SUPPORTED(Type.INVALID_DATA,
447, "Multiple disks of storage type [%s] are not supported in this
version of product", true),

DISKS_OF_MIXED_STORAGE_TYPE_ARE_NOT_SUPPORTED(Type.INVALID_DATA, 448,
"Disks of both storage types [%s] and [%s] are not supported in this
version of product", true),

UNKNOWN_STORAGE_TYPE(Type.INVALID_DATA, 449, "Unknown storage type
[%s]. please use LOCAL or PCS (case-insensitive) ", true),

CONFLICTING_STORAGE_TYPE_SETTINGS(Type.INVALID_DATA, 450, "Disk
attributes 'local' and 'type' are both specified and are not
compatible"),

INVALID_PRIMARY_DISK_SIZE_MUTATION(Type.INVALID_DATA, 451, "'primary-
disk-size' and 'change-primary-disk' cannot be used together"),

NO_HTTPS_FOR_CALLBACK(Type.INVALID_DATA, 500, "HTTPS is not supported
for callbacks")
```

Index

P

R

S

U