XML-RPC API for CloudLinux

The API allows to license & unlicense IP. Once IP is licenses, call to:

```
/usr/bin/clnreg_ks
```

This will register the server.

To make use of that API, you have to have secret key. To get secret key, please contact iseletsk@cloudlinux.com

API overview

Server URL is: https://cln.cloudlinux.com/clweb/xmlrpc

XMLRPC Exception codes:

- 1 - Internal (unknown) Error
- 10 - Not authorized
- 30 - Invalid call arguments
- 40 - Invalid IP format

Authentication token is produced by combining: login|timestamp|sha1(secret_key + timestamp).

Timestamp is number of seconds since January 1, 1970 (UTC), or POSIX time.

Namespace registration.*

Responsible for adding/removing Ip licenses and **registration ip based servers**

license(authToken, ip, type)

Register new IP license.

Arguments:

- token(string) - customer authentication token
- ip(string) - ip address
- type(int) - license type (1,2 or 16)

Return (int): 0 on success or -1 on error

Throws: XmlRpcException for critical errors

remove_license(authToken, ip)

Remove all IP licenses with all types for customer. Also un-registers from CLN server associated with IP.

Arguments:

- token(string) - customer authentication token

- ip(string) - ip address to remove

Return (int): 0 on success or -1 on error. Error will be returned also if account have no licenses for provided IP.

Throws: XmlRpcException for critical errors

remove_license(authToken, ip, type)

Remove IP licenses with specified type for customer. Also un-registers from CLN server associated with IP.

Arguments:

- token(string) - customer authentication token
- ip(string) - ip address to remove
- type(int) - license type (1,2 or 16)

Return (int): 0 on success or -1 on error. Error will be returned also if account have no licenses for provided IP.

Throws: XmlRpcException for critical errors

is_licensed(authToken, ip, checkAll)

Check IP license (non KernelCare, types 1 or 2.) was registered by any customer.

Deprecated method - used only for compatibility with old code

Arguments:

- token(string) - customer authentication token
- ip(string) - ip address to remove

Return (int): 0 if no IP license found or license code(1 or 2) depending under which type IP was licensed

Throws: XmlRpcException for critical errors

is_licensed(authToken, ip, checkAll)

Check if IP license was registered by any customer.

Arguments:

- token(string) - customer authentication token
- ip(string) - ip address to remove
- checkAll(boolean) - True will search for any type of license. False - only for types 1 or 2

Return (list<int>): List of registered license types or empty list if no license found

Throws: XmlRpcException for critical errors

reconcile(authToken)

Return list of all IP licenses owned by authorized user

Arguments:

- token(string) - customer authentication token

Return (list<structure>): List of structures or empty list. Each structure contains keys:

- IP(string)
- TYPE(int) - license type
- REGISTERED(boolean) - True if server was registered in CLN with this license

Throws: XmlRpcException for critical errors