API Documentation

THE API

Welcome to the LaciCloud API documentation. In this tutorial, you will learn how to issue commands to your account using just an API key and how to program applications using our API.

LaciCloud API is an ***full json REST-less API***, which uses parameters sent via POST to communicate with the server. It returns a JSON string as response:

{"ID":[msg_id],"MSG":"[error_message]","Success":[true/false]}

Error ID's and messages will be discussed later.

This is a type 1 API call.

There are type 2, API calls which return a JSON string, but different than the one returned by type 1, example getuservalues:

{"ftp_space_user_has":6626,"bitcoin_paid":12.701929740039,"first_time_boolean":"1","api_key

":"c308dc58364b38c9bdc309c0579667e1770a6999d3ba5d5b812cd0ee27f29f7d","limit":496,"id" :34}

Here is a list of API calls and their types:

Action	Type
addftpuser	**1**
removeftpuser	**1**
regenerateapikey	**1**
getuservalues	**2**
getftpuservalues	**2**
getftpuserslist	**2**
getftpusersusedspace	**2**
getftpusersvirtuallyusedspace	**2**
canchangetotier	**2**
gettierinfo	**2**
getusedbandwidth	**2**
getwebhostingvalues	**2**
resetwebhostingmysql	**1**
resetwebhostingperms	**1**
addwebhostingenv	**1**

All API POST requests must contain the parameter 'api_key' and 'action'. Your API key can be found by logging into to the LaciCloud UI. The 'action' parameter is equal to the name row, but we will go over those now.

Now we will go over each action and the required parameters.

addftpuser:

Create's an FTP user.

Parameters:

ftp_username – the name of the FTP user to be created (ex user)

ftp_password – password of the ftp user, can be empty only If not a master account, but must be set (ex secretpassword11)

ftp_space – how much space user will have (ex 100)

ftp_space_currency – specifies what currency the ftp_space parameter is to be handled in (mb/gb/tb)

starting_directory – the directory of the FTP user (ex /user)

Example request (POST Data):

api_key=[api_key]&action=addftpuser&ftp_username=user&ftp_password=secretpassword1&ftp_space=100&ftp_space_currency=mb&starting_directory=/user

Example response:

{"ID":31, "MSG":"Successfully created FTP user!", "Success": true}

removeftpuser

Removes an FTP user.

Parameters:

ftp_username – the name of the FTP user to remove

Example request (POST Data):

api_key=[api_key]&action=removeftpuser&ftp_username=user

Example response:

{"ID":32, "MSG":"Successfully removed FTP user!", "Success": true}

regenerateapikey

Regenerates your API key.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=regeneratapikey

Example response:

{“ID”:35, “MSG”:”Successfully regenerated API key!”, “Success”: true}

getuservalues

Outputs a JSON array of all user values, such as total FTP space and user limits.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getuservalues

Example response:

{"ftp_space_user_has":6626,"bitcoin_paid":12.701929740039,"first_time_boolean":"1","api_key

":"c308dc58364b38c9bdc309c0579667e1770a6999d3ba5d5b812cd0ee27f29f7d","limit":496,"id" :34}

On error it will output a default ID, MSG, Success JSON string.

getftpusersvalues

Will return a JSON array of every FTP user’s (in your account) quota, home directory, and FTP space.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getftpusersvalues

Example response:

[{"user":"user","home":"\/var\/ftp\/users\/34\/user","quota":100}]

On error it will output a default ID, MSG, Success JSON string.

getftpuserslist

If you only need to get the names of all the FTP users (in your account) in a JSON string, use this.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getftpuserslist

Example response:

["user","user1"]

getftpusersusedspace

Get the true (= actually used on server) total space in MB used by all your FTP users in your account.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getftpusersused

space

Example response:

3351

getftpusersvirtuallyusedspace

Get the virtual (= not actually used on server) total space in MB used by all your FTP users in your account.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getftpusersvirtu

allyusedspace

Example response:

8192

canchangetotier

Tests if you can change to specified tier.

Parameters:

tier (1, 2 or 3)

Example request (POST Data):

api_key=[api_key]&action=canchangetotier

&tier=1

Example response:

{"ID":28,"MSG":"You can change to this tier if you wish... Yay!","Success":true}

gettierinfo

Returns array of tier data including; FTP space, FTP users, down/up limits, monthly cost, short

description, bandwidth limit, and whether that tier has access to webhosting functions.

Parameters:

tier (1, 2 or 3)

Example request (POST Data):

api_key=[api_key]&action=gettierinfo&tier

=1

Example response:

[250000,125,32768,4096," - 25\u20ac \/ Month - For Expert Users", 2000000, true]

In this case, 250000 refers to FTP space in MB, 125 to the FTP users, 32768 kilobit per second for upload limit, 8192 kilobit per second for download limit, a short description including price (Use regex to extract), the bandwidth limit in MB, 2000000MB, and whether that tier has access to webhosting functions , in this case yes (true). This is for tier 3.

getusedbandwidth

Returns the used bandwidth of the LaciCloud account in megabytes.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getusedbandwidth

Example response:

375

getwebhostingvalues

Returns an array of webhosting values, such as sitename (without .lacicloud.net), mysql host, mysql username and mysql_password

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=getwebhostingvalues

Example response:

{"sitename":"laci","mysql_host":"localhost","mysql_username":"0a79539059876478","mysql_password":"cc1bb2c6416d598c4b89848e29839e1e"}

addwebhostingenv

Adds a new webhosting environment for a tier 2 or 3 user, provided the account does not yet have one.

Parameters:

sitename – the name of the subdomain you want

Example request (POST Data):

api_key=[api_key]&action=addwebhostingenv&sitename=laci

Example response:

{"ID":50,"MSG":"Successfully added webhosting environment!","Success":true}

resetwebhostingmysql

Resets the MySql password for your MySql user.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=resetwebhostingmysql

Example response:
{"ID":52,"MSG":"Successfully reset MySql password on webhosting environment!","Success":true}

resetwebhostingperms

Resets the permissions on the account's webhosting environment.

Parameters:

none

Example request (POST Data):

api_key=[api_key]&action=resetwebhostingperms

Example response:
{"ID":51,"MSG":"Successfully reset permissions on webhosting environment!","Success":true}

Now we will go over the different error codes and messages that can be returned by the API.

MESSAGES

Here is a table that you can use in your application.

Can be returned by	ID	MSG	Success
Everything (even type 2 calls)	1	DB/SQL/Internal error	false
addftpuser	31	Error validating FTP user info	false
addftpuser	32	Successfully created FTP user	true
removeftpuser	33	Couldn't remove FTP user	false
removeftpuser	34	Successfully removed FTP user	true
regenerateapikey	39	Successfully regenerated API key	True
addwebhostingenv	50	Successfully added webhosting environment	true
addwebhostingenv	48	Sitename already exists or is not in valid alphanumeric format	False
addwebhostingenv	54	User already has a subdomain or tier is not sufficient for this function	False
resetwebhostingmysql	52	Successfully reset MySql credentials	True
resetwebhostingmysql	54	User does not have a subdomain or tier is not sufficient for this function	False
resetwebhostingperms	51	Successfully reset permissions	True
resetwebhostingperms	54	User does not have a subdomain or tier is not sufficient for this function	False
API	43, 44	API key Is wrong (43), Not enough parameters for API (44)	false

Example: *{"ID":32, "MSG":"Successfully created FTP user!", "Success": true}*

Don't forget to always check for ID 1. Best is to check for success code, and error out if returned code is not the success code. You can also take the MSG field from the JSON return and display that to your user.
All other messages are non-messages (so just values returned, for example; **getftpusersusedspace, getusedbandwidth**).

If you have any issues with the API, you can contact us.