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:

addftpuser	**1**
removeftpuser	**1**
regenerateapikey	**1**
getuservalues	**2**
getftpuservalues	**2**
getftpuserslist	**2**
getftpusersusedspace	**2**
getftpusersvirtuallyusedspace	**2**
canchangetotier	**2**
gettierinfo	**2**

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 100.5)

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_passwword=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 and short description

Parameters:

tier (1, 2 or 3)

Example request (POST Data):

api_key=[api_key]&action=gettierinfo&tier

=1

Example response:

[525000,500,1000,5000," - 10\u20ac \/ Month - For advanced Users"]

In this case, 525000 refers to FTP space in MB, 500 to the FTP users, 1000kb/s for down limit, 5000kb/s, and short description including price (Use regex to extract). This is for tier 3.

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	true

		API key	
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}*
Always check for ID 1!

All other messages are non-messages (so just values returned, for example; **getftpusersusedspace**).