

# 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:

Name	Type
addftpuser	1
removeftpuser	1
regenerateapikey	1
getuservalues	2
getftpusersvalues	2
getftpuserslist	2

**All API POST requests must contain the parameter 'api\_key' and 'action'.** Your API key can be found by logging into 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_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=regenerateapikey
```

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"]
```

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

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	1, 2, 20, 25	Database connection errors (1, 2)/Unexpected error (20)/Error getting values (25) *ALWAYS CHECK FOR THIS*	false
addftpuser	21	Error while creating FTP user	false
addftpuser	31	Successfully created FTP user	true
removeftpuser	22	Couldn't remove FTP user	false
removeftpuser	32	Successfully removed FTP user	true
regenerateapikey	26	Error while regenerating API key	false
regenerateapikey	35	Successfully regenerated API key	true
getuservalues, getftpusersvalues, getftpuserslist	25	Error while getting values	false
API	41, 42, 43	No API key found (41), API key is wrong (42), Not enough parameters for API (43)	false

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

All other messages are non-messages (so just values returned).