

cPanel API 2 Functions - Email::listlists

Guide to cPanel API 2

[Return Data](#)

[Filter and Sort Output](#)

[Paginate Output](#)

[Output Columns](#)

[Calls from the WHM API](#)

[cPanel Tags \(deprecated\)](#)

AddonDomain:: [addaddondomain](#)

AddonDomain:: [delaaddondomain](#)

AddonDomain:: [listaddondomains](#)

Backups:: [listfullbackups](#)

Bandwidth:: [addhttpbandwidth](#)

Bandwidth:: [getbwdata](#)

Bandwidth:: [gethourlyaveragebwoveryear](#)

BoxTrapper:: [accountmanagelist](#)

Branding:: [addbrandingobj](#)

Branding:: [applist](#)

Branding:: [brandingeditor](#)

Branding:: [createpkg](#)

Branding:: [cssmerge](#)

Branding:: [delbrandingobj](#)

Branding:: [delpkg](#)

Branding:: [gensprites](#)

Branding:: [getbrandingpkg](#)

Branding:: [getbrandingpkgstatus](#)

Branding:: [getdefaultbrandingpkg](#)

Branding:: [installbrandingpkgs](#)

Branding:: [installimages](#)

Branding:: [killimgs](#)

Description

This function lists the cPanel account or domain's mailing lists.



Warning:

We **strongly** recommend that you use the following [UAPI](#) function instead of this function:

- [Email::list_lists](#) — This function lists the account's Mailman mailing lists.



Important:

In cPanel & WHM version 76 and later, when you disable the MailReceive role, the system **disables** this function. For more information, read our [How to Use Server Profiles](#) documentation.

Examples

```
https://hostname.example.com:2087/cpsess#####/json-api/cpanel?cpanel_jsonapi_user=user&cpanel_jsonapi_apiversion=2&cpanel_jsonapi_module=Email&cpanel_jsonapi_func=listlists&domain=example.com&regex=list
```



Note:

For more information, read our [Calls from the WHM API](#) documentation.

```
$cpanel = new CPANEL(); // Connect to cPanel - only do this once.
```

```
// List all of example.com's mailing lists.$list_mailing_lists = $cpanel->api2('Email', 'listlists', array('domain' => 'example.com',));
```

```
// List all of the cPanel account's mailing lists.$list_mailing_lists = $cpanel->api2('Email', 'listlists',);
```



Note:

For more information, read our [Guide to the LiveAPI System](#).

Function information

Module:

Email

API

Version:

[cPanel API 2](#)

Available in:

cPanel 11+

Methods:

GET, POST

Required

Parameters:

(none)

Return

Format:

s:

JSON, XML



Notes:

- XML or [call this](#)
- cPanel [deprec](#) XML or

Legacy

Equivalents:

(none)

About this module

The Email module modifies cPanel email account settings.

About cPanel API 2

cPanel API 2 accesses and modifies data for cPanel accounts.



Notes:

Branding:: [listmgtypes](#)

Branding:: [listobjecttypes](#)

Branding:: [preloadconf](#)

Branding:: [resetall](#)

Branding:: [resetcss](#)

Branding:: [resethtml](#)

Branding:: [resolve_file](#)

Branding:: [resolvelocalcss](#)

Branding:: [savelocalcss](#)

Branding:: [setbrandingpkgstatus](#)

Branding:: [showpkgs](#)

Branding:: [spritelist](#)

CSVImport:: [columnchoices](#)

CSVImport:: [configimport](#)

CSVImport:: [data](#)

CSVImport:: [fetchimporadata](#)

CSVImport:: [isenabled](#)

CSVImport:: [loaddata](#)

CSVImport:: [processdata](#)

CSVImport:: [sendcontact](#)

CSVImport:: [uploadimport](#)

Contactus:: [isenabled](#)

Contactus:: [sendcontact](#)

Cron:: [add_line](#)

Cron:: [edit_line](#)

Cron:: [fetchcron](#)

Cron:: [get_email](#)

Cron:: [listcron](#)

Cron:: [remove_line](#)

Cron:: [set_email](#)


CustInfo:: [contactemails](#)


CustInfo:: [contactprefs](#)

```
my $cpliveapi = Cpanel::LiveAPI->new(); # Connect to
cPanel - only do this once.

# List all of example.com's mailing lists.
my $list_mailing_lists = $cpliveapi->api2(
    'Email', 'listlists',
    {
        'domain' => 'example.com',
    }
);


# List all of the cPanel account's mailing lists.
my $list_mailing_lists = $cpliveapi->api2(
    'Email', 'listlists',
);
```

 **Note:**
For more information, read our [Guide to the LiveAPI System](#).

 **Warnings:**

- In cPanel & WHM version 11.30 and later, cPanel tags are **deprecated**. We **strongly** recommend that you **only** use the **LiveAPI** system to call the cPanel APIs. Examples are **only** present in order to help developers move from the old cPanel tag system to our [LiveAPI](#).
- cPanel API 2 calls that use cPanel tags vary in code syntax and in their output.
- For more information, read our [Deprecated cPanel Tag Usage](#) documentation.

```
cpapi2 --user=username Email listlists domain=example.com
regex=list
```

 **Notes:**

- You **must** [URI-encode](#) values.
- username represents your account-level username.
- You **must** run the `--user=username` command.
- For more information and additional output options, read our [Guide to cPanel API 2](#) documentation or run the `cpapi2 --help` command.
- If you run CloudLinux™, you **must** use the full path of the `cpapi2` command:

```
/usr/local/cpanel/bin/cpapi2
```

• These functions may require a feature list. To view the available features list on a cPanel element, view the WHM's [Feature Manager](#) interface.

• Unless you can

CustInfo: [displaycontactinfo](#)
 CustInfo: [savecontactinfo](#)
 DBmap: [status](#)
 DBmap: [version](#)
 DenyIp: [listdenyips](#)
 DiskCounter: [diskcounter](#)
 DiskUsage: [buildcache](#)
 DiskUsage: [clearcache](#)
 DiskUsage: [fetchraw_disk_usage](#)
 DiskUsage: [fetchdiskusage](#)
 DiskUsage: [fetchdiskusagewithextras](#)
 DKIMUI: [available](#)
 DKIMUI: [getrecords](#)
 DKIMUI: [install](#)
 DKIMUI: [installed](#)
 DKIMUI: [uninstall](#)
 DnsLookup: [name2ip](#)
 DomainLookup: [countbaseddomains](#)
 DomainLookup: [getbaseddomains](#)
 DomainLookup: [getdocroot](#)
 DomainLookup: [getdocroots](#)
 DomainLookup: [getmaindomain](#)
 Email: [accountname](#)
 Email: [adddomainforward](#)
 Email: [addforward](#)
 Email: [addmx](#)
 Email: [addpop](#)
 Email: [browseboxes](#)
 Email: [changemx](#)
 Email: [checkmainidscard](#)
 Email: [clearpopcache](#)
 Email: [deletefilter](#)

```

{
  "cpanelresult": {
    "apiversion": 2,
    "func": "listlists",
    "data": [
      {
        "diskused": "20379",
        "listid": "list_example.com",
        "subscribe_policy": 1,
        "archive_private": 0,
        "accesstype": "public",
        "desthost": "172.16.254.1",
        "advertised": 1,
        "humandiskused": "19.9\u00a0KB",
        "listadmin": "admin@example.com",
        "list": "list@example.com"
      }
    ],
    "event": {
      "result": 1
    },
    "module": "Email"
  }
}

```



Note:

Use cPanel's *API Shell* interface (*cPanel >> Home >> Advanced >> API Shell*) to directly test cPanel API calls.

Parameters

Parameter	Type	Description	Possible values	Example
domain	<i>string</i>	The domain to search for mailing lists. If you do not use this parameter, the function lists all of the cPanel account's mailing lists.	A valid domain on the server.	example.com
regex	<i>string</i>	A Perl Compatible Regular Expression (PCRE) that filters the results. If you do not use this parameter, the function lists all of the mailing lists for all of the cPanel account's domains.	A valid PCRE.	list

Returns

Return	Type	Description	Possible values	Example
diskused	<i>integer</i>	The disk space that the mailing list currently uses, in megabytes (MB).	A positive integer.	20379
listid	<i>string</i>	The mailing list's name and domain.	The mailing list name, an underscore (_), and the domain.	list_example.com

If this API violates the WHM API, you must use the appropriate cPanel ports (2080 or 2083). You can also call some functions via cPanel's Webmail port

Email: [delforward](#)

Email: [delmx](#)

Email: [delpop](#)

Email: [disablefilter](#)

Email: [editquota](#)

Email: [enablefilter](#)

Email: [fetchautoresponder](#)

Email: [fetchcharmaps](#)

Email: [filteractions](#)

Email: [filterlist](#)

Email: [filtername](#)

Email: [filterrules](#)

Email: [get_archiving_configuration](#)

Email: [get_archiving_default_configuration](#)

Email: [get_archiving_types](#)

Email: [get_default_email_quota](#)

Email: [get_email_signing](#)

Email: [get_max_email_quota](#)

Email: [getabsbrowse_dir](#)

Email: [getalwaysaccept](#)

Email: [getdiskusage](#)

Email: [getmxcheck](#)

Email: [has_delegated_mailman_lists](#)

Email: [list_system_filter_info](#)

Email: [listaliasbackups](#)

Email: [listautoresponders](#)

Email: [listdefaultaddresses](#)

Email: [listdomainforwards](#)

Email: [listfilterbackups](#)

Email: [listfilters](#)

Email: [listforwards](#)

subscribe_policy	integer	The level of control that the mailing list administrator maintains over new subscribers.	<ul style="list-style-type: none"> 1 — Anyone can subscribe. The system sends a confirmation email. 2 — The administrator must approve subscriptions. The system does not send a confirmation email. 3 — The administrator must approve subscriptions. The system sends a confirmation email. 	1
archive_private	Boolean	Whether the mailing list archive is private.	<ul style="list-style-type: none"> 1 — The mailing list archive is private. 0 — The mailing list archive is public. 	0
access_type	string	The user's level of access to the mailing list.	<ul style="list-style-type: none"> private — The list contains all of the following settings: <ul style="list-style-type: none"> The list contains private archives. The administrator must approve subscriptions. The Mailman directory page does not display the list. public — The list contains any of the following settings: <ul style="list-style-type: none"> The list contains public archives. Anyone can subscribe. The Mailman directory page displays the list. 	public
destdomain	string	The IP address of the domain that you specified in the domain input parameter.	An IPv4 address.	172.16.254.1
advertised	Boolean	Whether the Mailman directory page displays the list.	<ul style="list-style-type: none"> 1 — The Mailman directory page displays the list. 0 — The Mailman directory page does not display the list. 	1
humandiskused	string	The disk space that the mailing list uses, in human-readable format.	The disk space that the mailing list uses, a non-breaking space (\u00a0), and the unit of measure.	19.9 \u00a0KB
listadmin	string	The mailing list's administrators' email addresses.	A comma-separated list of email addresses.	admin@example.com, admin2@example.com
list	string	The mailing list name and domain.	The mailing list name, the @ character, and the domain.	list@example.com

ts
(2
0
95
o
r
2
0
96
).

Find a function

Related functions

- [cPanel API 2 Functions - Email::delpop](#) — This function deletes an email account.
- [cPanel API 2 Functions - Email::editquota](#) — This function modifies an email account's disk quota.
- [cPanel API 2 Functions - Email::passwdpop](#) — This function modifies an email account's password.
- [cPanel API 2 Functions - Email::listdefaultaddresses](#) — This function retrieves the domain's default address and its rules.
- [cPanel API 2 Functions - Email::listaliasbackups](#) — This function retrieves a list of the account's domains that use aliases and custom default addresses.

Email: [listlists](#)

Email: [listmaildomains](#)

Email: [listmx](#)

Email: [listmxs](#)

Email: [listpops](#)

Email: [listpopssingle](#)

Email: [listpopswithdisk](#)

Email: [listpopswithimage](#)

Email: [loadfilter](#)

Email: [passwdpop](#)

Email: [reorderfilters](#)

Email: [set_archiving_configuration](#)

Email: [set_archiving_default_configuration](#)

Email: [set_email_singing](#)

Email: [setalwaysaccept](#)

Email: [setdefaultaddress](#)

Email: [setmxcheck](#)

Email: [storefilter](#)

Email: [tracefilter](#)

EmailTrack: [search](#)

EmailTrack: [stats](#)

EmailTrack: [trace](#)

Encoding: [get_encodings](#)

Encoding: [guess_file](#)

Encoding: [guess_file_opts](#)

Fileman: [autocomplete_dir](#)

Fileman: [fileop](#)

Fileman: [getabsdir](#)


Fileman: [getdir](#)

Fileman: [getdiractions](#)

Fileman: [getdiskinfo](#)

Fileman: [getedittype](#)

Fileman: [getfileactions](#)

reason	<i>string</i>	A reason for failure. <div style="border: 1px solid orange; padding: 5px; background-color: #fff9c4;">  Note: This function only returns a reason value if it failed. </div>	A string that describes the error.	This is an error message.
result	<i>Boolean</i>	Whether the function succeeded.	<ul style="list-style-type: none"> • 1 — The function succeeded. • 0 — The function failed. 	1

Fileman:: [getpath](#)
Fileman:: [listfiles](#)
Fileman:: [mkdir](#)
Fileman:: [mkfile](#)
Fileman:: [savefile](#)
Fileman:: [search](#)
Fileman:: [statfiles](#)
Fileman:: [uploadfiles](#)
Fileman:: [viewfile](#)
ForcePassword:: [get_force_password_flags](#)
ForcePassword:: [update_force_password_flags](#)
Frontpage:: [fpenabled](#)
Frontpage:: [list](#)
FSTest:: [dirisempty](#)
Ftp:: [addftp](#)
Ftp:: [delftp](#)
Ftp:: [listftp](#)
Ftp:: [listftpsessions](#)
Ftp:: [listftpwithdisk](#)
Ftp:: [passwd](#)
Ftp:: [setquota](#)
Gpg:: [listgpgkeys](#)
Gpg:: [listsecretgpgkeys](#)
Gpg:: [number_of_private_keys](#)
Gpg:: [number_of_public_keys](#)
Htaccess:: [listuser](#)
Htaccess:: [listusers](#)
HttpUtils:: [getdirindices](#)
ImageManager:: [thumbnail](#)
LangMods:: [getarchname](#)
LangMods:: [getkey](#)
LangMods:: [getprefix](#)
LangMods:: [install](#)
LangMods:: [langlist](#)
LangMods:: [list_available](#)

LangMods:: [list_installed](#)

LangMods:: [magic_status](#)

LangMods:: [pre_run](#)

LangMods:: [search](#)

LangMods:: [setup](#)

LangMods:: [uninstall](#)

LangMods:: [update](#)

LeechProtect:: [setup](#)

Locale:: [get_encoding](#)

Locale:: [get_html_dir_attr](#)

Locale:: [get_locale_name](#)

Locale:: [get_user_locale](#)

Locale:: [get_user_locale_name](#)

Locale:: [numf](#)

Logaholic:: [adduser](#)

Logaholic:: [adduserprofile](#)

Logaholic:: [deleteuser](#)

Logaholic:: [deleteuserprofile](#)

Logaholic:: [edituser](#)

Logaholic:: [fetchuser](#)

Logaholic:: [listuserprofiles](#)

Logaholic:: [logaholiclink](#)

Logaholic:: [processstats](#)

LVEInfo:: [getUsage](#)

Mime:: [list_hotlinks](#)

Mime:: [listhandlers](#)

Mime:: [listmime](#)

Mime:: [listredirects](#)

Mime:: [redirectname](#)

Mime:: [redirecturlname](#)

MysqlFE:: [authorizehost](#)

MysqlFE:: [changedbuserpassword](#)

MysqlFE:: createdb
MysqlFE:: createdb
user
MysqlFE:: dbuserexi
sts
MysqlFE:: deauthori
zehost
MysqlFE:: deletedb
MysqlFE:: deletedb
user
MysqlFE:: getalldb
info
MysqlFE:: getalldb
usersanddbs
MysqlFE:: getdbuse
rprivileges
MysqlFE:: getdbuse
rs
MysqlFE:: gethosts
MysqlFE:: getmysql
privileges
MysqlFE:: getmysql
serverprivileges
MysqlFE:: has_myc
nf_for_cpuser
MysqlFE:: listdbs
MysqlFE:: listdbsba
ckup
MysqlFE:: listhosts
MysqlFE:: listusers
MysqlFE:: listusers
info
MysqlFE:: revokedb
userprivileges
MysqlFE:: setdbuse
rprivileges
MysqlFE:: userdbpri
vs
MysqlOptimizer:: fet
ch_slow_queries
Net:: dnszone
Net:: traceroute
News:: does_news_
exist
News:: does_news_
type_exist
News:: get_news
NVData:: get
NVData:: set
NVData:: setall

Park:: [listaddondomains](#)

Park:: [listparkeddomains](#)

Park:: [park](#)

Park:: [unpark](#)

Passwd:: [change_password](#)

Passwd:: [set_digest_auth](#)

PasswdStrength:: [appstrengths](#)

PasswdStrength:: [get_password_strength](#)

PasswdStrength:: [get_required_strength](#)

PHPINI:: [getalloptions](#)

PHPINI:: [getoptions](#)

Postgres:: [listdbs](#)

Postgres:: [listusers](#)

Postgres:: [listusersindb](#)

Postgres:: [userexists](#)

Rand:: [getranddata](#)

Resellers:: [get_sub_accounts](#)

RoR:: [addapp](#)

RoR:: [changeapp](#)

RoR:: [importrails](#)

RoR:: [listapps](#)

RoR:: [listrewrites](#)

RoR:: [needsimport](#)

RoR:: [removeapp](#)

RoR:: [removerewrite](#)

RoR:: [restartapp](#)

RoR:: [setuprewrite](#)

RoR:: [softrestartapp](#)

RoR:: [startapp](#)

RoR:: [stopapp](#)

SetLang:: [setlocale](#)

SourceIPCheck:: [addip](#)

SourceIPCheck:: [deleteip](#)

SourceIPCheck:: [getaccount](#)

SourceIPCheck:: list
ips

SourceIPCheck:: loadsecquestions

SourceIPCheck:: resetsecquestions

SourceIPCheck:: samplequestions

SourceIPCheck:: savequestions

SPFUI:: available

SPFUI:: count_settings

SPFUI:: entries_complete

SPFUI:: get_raw_record

SPFUI:: getmainserverip

SPFUI:: install

SPFUI:: installed

SPFUI:: list_settings

SPFUI:: load_current_values

SPFUI:: uninstall

SSH:: authkey

SSH:: converttoppk

SSH:: delkey

SSH:: fetchkey

SSH:: genkey

SSH:: genkey_legacy

SSH:: importkey

SSH:: listkeys

SSL:: fetchcabundle

SSL:: fetchinfo

SSL:: gencrt

SSL:: gencsr

SSL:: genkey

SSL:: getcname

SSL:: installssl

SSL:: listcrts

SSL:: listcsrs

SSL:: listkeys

SSL:: listsslitems

SSL:: uploadcrt

SSL:: [uploadkey](#)

SSLInfo:: [fetchinfo](#)

StatCache:: [clearcache](#)

Stats:: [getmonthlybandwidth](#)

Stats:: [getmonthlydomainbandwidth](#)

Stats:: [getthismonthsbwusage](#)

Stats:: [lastapachehits](#)

Stats:: [lastvisitors](#)

Stats:: [listanalog](#)

Stats:: [listawstats](#)

Stats:: [listlastvisitors](#)

Stats:: [listrawlogs](#)

Stats:: [listurchin](#)

Stats:: [listwebalizer](#)

StatsBar:: [getrowcounter](#)

StatsBar:: [rowcounter](#)

StatsBar:: [setrowcounter](#)

StatsBar:: [stat](#)

SubDomain:: [addsubdomain](#)

SubDomain:: [changedocroot](#)

SubDomain:: [delsubdomain](#)

SubDomain:: [getreservedsubdomains](#)

SubDomain:: [listsubdomains](#)

SubDomain:: [validregex](#)

Themes:: [apply_new_theme](#)

Themes:: [does_panel_theme_exist](#)

Themes:: [get_available_themes](#)

Themes:: [get_themes_list](#)

Themes:: [restore_saved_theme](#)

UI:: [dynamicincludelist](#)

UI:: [includelist](#)

UI:: listform

UI:: paginate

UI:: paginate_list

UserHttpUtils:: getdirindices

WebDisk:: addwebdisk

WebDisk:: delwebdisk

WebDisk:: hasdigest

WebDisk:: listwebdisks

WebDisk:: passworddisk

WebDisk:: set_digest_auth

WebDisk:: set_home_dir

WebDisk:: set_perms

WebDisk:: setstatus

WebmailApps:: listwebmailapps

ZoneEdit:: add_zone_record

ZoneEdit:: edit_zone_record

ZoneEdit:: fetch_cpanel_generated_domains

ZoneEdit:: fetchzone

ZoneEdit:: fetchzone_records

ZoneEdit:: fetchzones

ZoneEdit:: get_zone_record

ZoneEdit:: remove_zone_record

ZoneEdit:: resetzone