

cPanel API 2 Functions - Email::storefilter

Guide to cPanel API 2

[Return Data](#)

[Filter and Sort Output](#)

[Paginate Output](#)

[Output Columns](#)

[Calls from the WHM API](#)

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

[AddonDomain:: add addondomain](#)

[AddonDomain:: dela ddondomain](#)

[AddonDomain:: lista ddondomains](#)

[Backups:: listfullbackups](#)

[Bandwidth:: addhttp bandwidth](#)

[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 creates an email filter.

For more information about Exim filters, read [Exim's documentation](#).



Warning:

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

- [Email::store_filter](#) — This function creates a new email filter.

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=storefilter&filtername=coffee&account=user@example.com&action1=deliver&dest1=cheesecloth%40example.com&part1=%24header_subject%3A&match1=contains&vall=curds&opt1=or&part2=%24message_body&match2=is&val2=whey
```



Note:

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

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

// Create an email filter.
$create_mail_filter = $cpanel->api2(
    'Email', 'storefilter',
    array(
        'filtername' => 'coffee',
        'account' => 'user@example.com',
        'action1' => 'deliver',
        'dest1' => 'cheesecloth@example.com',
        'part1' => '$header_subject:',
        'match1' => 'contains',
        'vall' => 'curds',
        'opt1' => 'or',
        'part2' => '$message_body',
        'match2' => 'is',
        'val2' => 'whey',
    )
);
```



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: action*, filter name, match*, part*, val*
Return Format: JSON, XML



Notes:

- XML output: [call this](#)
- cPanel deprecated XML output

Legacy Equivalents:

- [Email::addfilter](#)

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:: [resolvocalcss](#)

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

# Create an email filter.
my $create_mail_filter = $spliveapi->api2(
    'Email', 'storefilter',
    {
        'filtername' => 'coffee',
        'account' => 'user@example.com',
        'action1' => 'deliver',
        'dest1' => 'cheesecloth@example.com',
        'part1' => '$header_subject:',
        'match1' => 'contains',
        'val1' => 'curds',
        'opt1' => 'or',
        'part2' => '$message_body',
        'match2' => 'is',
        'val2' => 'whey',
    }
);
```



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.

- These functions may require a feature list. To view the available features on a cPanel installation, view the WHM's [Feature Manifest Generator](#) interface. • Unless you can

CustInfo: [displaycontactinfo](#)

CustInfo: [savecontactinfo](#)

DBmap: [status](#)

DBmap: [version](#)

DenyIp: [listdenyips](#)

DiskCounter: [diskcounter](#)

DiskUsage: [buildcache](#)

DiskUsage: [clearcache](#)

DiskUsage: [fetch_raw_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: [checkmaindiscard](#)

Email: [clearpopcache](#)

Email: [deletefilter](#)

```
cpapi2 --user=username Email storefilter
filtername=coffee account=user%40example.com
action1=deliver dest1=cheesecloth@example.com
part1=$header_subject: match1=contains vall=curds opt1=or
part2=$message_body match2=is val2=whey
```



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
```

```
{
  "cpanelresult": {
    "apiversion": 2,
    "func": "storefilter",
    "data": [
      {
        "ok": 1,
        "account": "user@example.com",
        "error": 0,
        "result": "Filter Saved."
      }
    ],
    "event": {
      "result": 1
    },
    "module": "Email"
  }
}
```



Note:

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

Parameters



Important:

For additional use information, read the [Create multiple rules](#) section.

Parameter	Type	Description	Possible values	Example
-----------	------	-------------	-----------------	---------

ll
thi
s
A
P
I
v
i
a
th
e
W
H
M
A
P
I
,
yo
u
mu
st
u
se
th
e
a
p
pr
o
pr
iat
e
c
P
a
n
el
p
or
ts
(2
0
82
o
r
2
0
83
).
Y
o
u
ca
n
al
so
ca
ll
so
me
fu
nc
tio
ns
vi
a
c
P
a
n
el'
s
W
e
b
m
ail
p
or

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_g_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](#)

account	<i>string</i>	The email address, for user-level filters. If you do not use this parameter, the function creates an account-level filter.	A valid email address on the account.	user@example.com
filtername	<i>string</i>	Required The filter's name.	A string.	coffee
oldfiltername	<i>string</i>	An existing filter's name, to rename it. If you do not use this parameter, the function creates a new filter.	A filter on the account.	pool
action*	<i>string</i>	Required The filter's action.	<ul style="list-style-type: none"> deliver — Deliver the message to the <code>dest*</code> address. fail — Force a delivery failure. finish — Stop processing the message. save — Save the message to the <code>dest*</code> file. pipe — Pipe the message to the <code>dest*</code> application. 	deliver
dest*	<i>string</i>	The destination for filtered mail. <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p>Note:</p> <p>You must use this parameter if the action value is <code>deliver</code>, <code>save</code>, or <code>pipe</code>.</p> </div>	<ul style="list-style-type: none"> A file path. An application path. A valid email address. 	cheeseclot@example.com
match*	<i>string</i>	Required The filter's match type. <ul style="list-style-type: none"> If the <code>val*</code> parameter is a string, use a string operator. If the <code>val*</code> parameter is an integer, use a numeric operator. 	String operators: <ul style="list-style-type: none"> is matches contains does not contain begins does not begin ends does not end does not match Numeric operators: <ul style="list-style-type: none"> is above is not above is below is not below 	contains

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

Find a function

Related functions

- [cPanel API 2 Modules - Email](#) — The Email module modifies cPanel email account settings.
- [cPanel API 2 Functions - Email: accountname](#) — This function retrieves an email account's username.
- [cPanel API 2 Functions - Email: adddomainforward](#) — This function creates a domain-level email forwarder.
- [cPanel API 2 Functions - Email: addforward](#) — This function creates an email forwarder.
- [cPanel API 2 Functions - Email: addmx](#) — This function creates an MX record.

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_signing](#)

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:: [autocompletedir](#)

Fileman:: [fileop](#)

Fileman:: [getabsdir](#)

Fileman:: [getdir](#)

Fileman:: [getdiractions](#)

Fileman:: [getdiskinfo](#)

Fileman:: [getedittype](#)

Fileman:: [getfileactions](#)

opt*	<i>string</i>	The connection between multiple conditions.	<ul style="list-style-type: none"> and or 	and
part*	<i>string</i>	Required The email section to query.	<ul style="list-style-type: none"> \$header_from: — Match against the From: section. \$header_subject: — Match against the Subject: section. \$header_to: — Match against the To: section. \$reply_address: — Match against the Reply To: section. \$message_body — Match against the message's body. \$message_headers — Match against the message's headers. foranyaddress \$h_to:,\$h_cc: — Match against all message recipients. <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p>Note:</p> <p>Generally, the recipient does not receive the BCC field in an email's header. For this reason, you cannot use the BCC field in a filter.</p> </div> <ul style="list-style-type: none"> not delivered — Match if the message is not queued for delivery. error_message — Match if the incoming message bounced. 	\$message_body
val*	<i>string</i> or <i>integer</i>	Required The value to match.	<ul style="list-style-type: none"> Any string value. A positive integer. 	coconut

Create multiple rules

You may create up to 4,096 separate sets of conditions in one filter. To do this, append numbers to the parameter names.

To create a filter with two sets of actions and conditions, use the following parameters:

- Assign the information for the first filter rule to the `action1`, `dest1`, `match1`, `opt1`, `part1`, and `val1` parameters.
- Assign the information for the second filter rule to the `action2`, `dest2`, `match2`, `opt2`, `part2`, and `val2` parameters.

To create a filter that uses one set of actions but two sets of conditions, use the following parameters:

- Assign the actions to the `action1` and `dest1` parameters.
- Assign the first set of conditions to the `match1`, `opt1`, `part1`, and `val1` parameters.

- Assign the second set of conditions to the match2, opt2, part2, and val2 parameters.

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](#)

Returns

Return	Type	Description	Possible values	Example
ok	<i>Boolean</i>	Whether the function succeeded.	<ul style="list-style-type: none"> • 1 — The function succeeded. • 0 — The function failed. 	1
account	<i>string</i>	The email address or cPanel account to which the filter applies.	<ul style="list-style-type: none"> • An email address, for account-level filters. • The cPanel username, for user-level filters. 	user@example.com
error	<i>Boolean</i>	Whether the function encountered an error.	<ul style="list-style-type: none"> • 1 — Errors. • 0 — No errors. 	0
result	<i>string</i>	A message of success or a reason for failure.	<ul style="list-style-type: none"> • Filter Saved. — The function succeeded. • An error message. — The function failed. 	Filter Saved.
reason	<i>string</i>	A reason for failure. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Note: This function only returns a <code>reason</code> value if it failed.</p> </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

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:: [listparkedomains](#)

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