

---

# ActiveCampaign

*Remote API*

---

## Remote API

### Usage

In order to use the API, you will need to write an application that can connect to the server where your Email Marketing is located, send a request to it and handle the response it receives from your Email Marketing software. Usually (if using PHP) this can be done by using CURL library.

In order to use the API, you will need a valid user account for accessing your Email Marketing software. The API username and password are the same as your login access to the Admin panel.

Requests

Every API request needs to have an action; that way the system knows what action to perform and what result to return.

Actions that send data to the server (add/edit actions, but not delete actions) have to be requested as POST, while actions that retrieve the data need to be requested as GET. (The exception are "delete" actions, since inputs are IDs only. Those should be submitted as GET.)

At the moment (new actions as added often), Email Marketing software lets you run the following actions:

- list\_view - returns the list info. Input: List ID
- list\_list - returns a list of lists. Input: comma-separated list of List IDs
- list\_add - adds the list. Input: complete list settings
- list\_edit - edits the list. Input: complete list settings and List ID to edit
- list\_delete - deletes the list. Input: List ID
- list\_delete\_list - deletes a list of lists. Input: comma-separated list of List IDs
- subscriber\_view - returns the subscriber info. Input: Subscriber ID or Subscriber Email
- subscriber\_view\_email - returns the subscriber info. Input: Subscriber Email
- subscriber\_list - returns a list of subscribers. Input: comma-separated list of Subscriber IDs
- subscriber\_add - adds the subscriber. Input: complete subscriber info with all lists he should belong to from now on
- subscriber\_edit - edits the subscriber. Input: complete subscriber info with all lists he should belong to from now on and Subscriber ID to edit
- subscriber\_delete - deletes the list. Input: Subscriber ID
- subscriber\_delete\_list - deletes a list of subscribers. Input: comma-separated list of Subscriber IDs
- user\_view - returns the user info. Input: User ID
- user\_view\_email - returns the user info. Input: User email address
- user\_view\_username - returns the user info. Input: User username
- user\_list - Returns a list of lists. Input: comma-separated list of User IDs
- user\_add - adds the user. Input: complete user info
- user\_edit - edits the user. Input: complete user info and User ID to edit
- user\_delete - deletes the list. Input: User ID
- user\_delete\_list - deletes a list of users. Input: comma-separated list of User IDs
- message\_view - returns the message info. Input: Message ID
- message\_list - returns a list of messages. Input: comma-separated list of Message IDs
- message\_add - adds the message. Input: complete message settings
- message\_edit - edits the message. Input: complete message settings and Message ID to edit
- message\_delete - deletes the list. Input: Message ID
- message\_delete\_list - deletes a list of lists. Input: comma-separated list of Message IDs
- campaign\_create - adds a new campaign.
- campaign\_send - sends an existing campaign. Input: Campaign ID, Subscriber Email
- campaign\_status - changes the campaign status. Input: Campaign ID

---

campaign\_delete - deletes the campaign. Input: Campaign ID

campaign\_delete\_list - deletes a list of campaigns. Input: comma-separated list of Campaign IDs

form\_view - returns the subscription form info. Input: Subscription Form ID

form\_list - returns a list of subscription forms. Input: comma-separated list of Subscription Form IDs

form\_add - adds the subscription form. Input: complete subscription form settings

form\_edit - edits the subscription form. Input: complete subscription form settings and Subscription Form ID to edit

form\_delete - deletes the subscription form. Input: Subscription Form ID

form\_delete\_list - deletes a list of subscription forms. Input: comma-separated list of Subscription Form IDs

## Result

API can return the results in:

serialized format - data is returned in a serialized format and can be decoded with PHP's native unserialize() function

JSON format - data is returned in JSON format and can be decoded with json\_decode() function (included in PHP since 5.2.0)

XML format - write your own XML parser to handle the output

In order to set your favorite output format, change the 'api\_output' variable. Possible values are 'serialize', 'xml' and 'json'.

## Examples

Examples on how to use each API request can be found in your /docs/api-examples/ folder.