
vero Documentation

Release 1.2.0

Wave Accounting Inc.

July 07, 2015

1	Features	3
2	Installation	5
3	Usage	7
3.1	Create instance	7
3.2	Add user	7
3.3	Edit user	7
3.4	Add user tags	8
3.5	Remove user tags	8
3.6	Unsubscribe user	8
3.7	Add event	8
3.8	Re-identify a user	8

Vero is an API wrapper for event logging in your Python application. Fetch your auth token from your [Vero](#) account and use the python interface instead of [API](#) web hooks.

```
>>> from vero import VeroEventLogger
>>> logger = VeroEventLogger(auth_token)
>>> user_id = 42
>>> user_data = {
    'full_name': 'Jane Doe'
}
>>> response = logger.add_user(user_id, user_data)
>>> response.status_code
200
```

Features

Modify user data and log events. Run in live or test mode.

- Add user
- Edit user
- Add user tags
- Remove user tags
- Unsubscribe user
- Add event

Installation

Install the package from PyPI

```
pip install vero
```

Usage

3.1 Create instance

Use the authorization token from your Vero account page to create a VeroEventLogger object.

```
>>> from vero import VeroEventLogger
>>> auth_token = "foobar"
>>> logger = VeroEventLogger(auth_token)
```

After creating an instance of VeroEventLogger as `logger` use any of the following methods to access Vero. All methods will accept the keyword argument `development_mode=True` to enable logging in test mode.

3.2 Add user

Create a new user with the information in `user_data`. `user_email` is optional but is needed to trigger emails to the user.

```
>>> user_id = 1
>>> user_email = 'johndoe@example.com'
>>> user_data = {
    'first name': 'John',
    'last name': 'Doe'
}
>>> logger.add_user(user_id, user_data, user_email=user_email)
```

3.3 Edit user

Add or change fields in `user_data` for the user.

```
>>> user_id = 1
>>> user_data = {
    'first name': 'Jane'
}
>>> logger.edit_user(user_id, user_data)
```

3.4 Add user tags

Add each tag in `tag_list` to the user.

```
>>> user_id = 1
>>> tag_list = ['blue', 'red', 'yellow']
>>> logger.add_tags(user_id, tag_list)
```

3.5 Remove user tags

Remove each tag in `tag_list` from the user.

```
>>> user_id = 1
>>> tag_list = ['yellow']
>>> logger.remove_tags(user_id, tag_list)
```

3.6 Unsubscribe user

Unsubscribe the user from triggering future events.

```
>>> user_id = 1
>>> logger.unsubscribe_user(user_id)
```

3.7 Add event

Note: adding an event with a user id that doesn't exist will create the user.

Event data can contain whatever fields are needed.

```
>>> user_id = 2
>>> user_email = 'janedoe@example.com'
>>> event_name = 'Visited Website'
>>> event_data = {
    'date': 'today',
    'visited': 'front page'
}
>>> logger.add_event(event_name, event_data, user_id, user_email=user_email)
```

3.8 Re-identify a user

Change a user's identifier (or ID) in Vero. This method accept their current (old) identifier and the identifier to replace it.

```
>>> user_id = 1
>>> new_user_id = 2
>>> logger.reidentify_user(user_id, new_user_id)
```