
Vinely Documentation

Release 0.9

Vinely

June 20, 2013

CONTENTS

1	Authentication and Authorization	3
1.1	Signup	3
1.2	Login	3
1.3	Logout	3
2	Accounts: User and profile management	5
2.1	User	5
2.2	Profile	5
3	Parties and Events	7
3.1	Events	7
3.2	Party	7
3.3	Party Invitations and RSVP	7
4	Personality and Wine Ratings	9
4.1	Wine Personality	9
4.2	Wine Ratings	9
5	Indices and tables	11
	HTTP Routing Table	13

Contents:

AUTHENTICATION AND AUTHORIZATION

1.1 Signup

Vinely only allows you to sign up as a Host, Pro or as a Vinely Club Member.

POST /api/v1/auth/signup/

Note: The current implementation allows you to signup but without an unassigned role. Clarity is still required as to what role signups are allowed.

1.2 Login

Allows you to log into the system. All API calls expect the user to be logged in.

POST /api/v1/auth/login/

A successful login returns an `api_key` for that session. This `api_key` must be sent along with all subsequent requests

Note: All API calls will only allow you to view the information pertaining to the logged in user.

To use the API Key, you can specify and pass the `username/api_key` combination either via an `Authorization` http header or as `GET/POST` parameters:

Examples:

```
# As Authorization header
Authorization: ApiKey attendee1@example.com:6f6f5ac1a65a6c9ae78ed8a9c7bc59d0430cb89e
```

```
# As GET Parameters
http://dev.vinely.com/api/v1/user/?username=attendee1@example.com&api_key=6f6f5ac1a65a6c9ae78ed8a9c7bc59d0430cb89e
```

Warning: Use of the `Authorization` header is preferred for security reasons.

1.3 Logout

POST /api/v1/auth/logout/

ACCOUNTS: USER AND PROFILE MANAGEMENT

Handle all user and profile management requests

2.1 User

Fetching user information.

GET `/api/v1/user/`

Detailed profile information is included in the returned information but it can also be fetched in a separate API call if required. *See below.*

Note: This exists for the purposes of convention and can be used if you don't have the `user_id`. It returns a list with only one item i.e. the profile of the person currently logged in.

Get information by id.

GET `/api/v1/profile/{id}/`

This returns information about the user with the provided `id`.

Note: For security reasons, the `id` must be the `user_id` of the currently logged in person.

Update user information

PUT `/api/v1/user/{id}/`

Note: For security reasons, the `id` must be the `user_id` of the currently logged in person.

2.2 Profile

Returns user profile information

GET `/api/v1/profile/`

Note: This exists for the purposes of convention. It returns a list with only one item i.e. the profile of the person currently logged in.

Get information by id.

GET `/api/v1/profile/{id}/`

This returns information about the profile with the provided `id`.

Note: For security reasons, the `id` must be the `profile_id` of the currently logged in person.

Update profile information

PUT `/api/v1/profile/{id}/`

Note: For security reasons, the `id` must be the `profile_id` of the currently logged in person.

PARTIES AND EVENTS

3.1 Events

Vinely Events are public parties hosted by Vinely and users can come to the site and RSVP to attend. Generally no invitations are sent out for Events.

Fetch event list

GET /api/v1/event/

This returns the list of all upcoming Vinely events.

Note: Past events are not returned

Fetch information about a specific event

GET /api/v1/event/{id}/

3.2 Party

Vinely Parties are private parties where someone has to be invited in order to attend. Invitations are sent out via mail which contains an RSVP link.

Fetch a list of upcoming parties that a user has been invited to

GET /api/v1/party/

Note: Past parties are not returned

Fetch information about a specific party

GET /api/v1/party/{id}/

3.3 Party Invitations and RSVP

Every Vinely Party/Event has party invites that associate users with a party.

Fetch a list of all party invitations

GET /api/v1/partyinvite/

Fetch information about a specific party invite.

GET `/api/v1/partyinvite/{id}/`

Creating a party invitation.

Hosts and Pro's can invite people to a party at any time. Other tasters can only invite people to a party if the Host allows it.

POST `/api/v1/partyinvite/`

Warning: Unlike the other calls where `relation` fields require the uri, you should provide the names and email of the `invitee` when creating an invitation.
If the user does not exist, a new account will be created for them.

Example:

Only the invitee and party fields are required, all others are optional.

```
{
  "invitee": {
    "first_name": "First",
    "last_name": "Last",
    "email": "someuser@example.com"
  },
  "party": "/api/v1/party/123/"
}
```

Updating a party invitation

PUT `/api/v1/partyinvite/{id}/`

A party invitation can be updated at any time up until the day of the event.

RSVP'ing to an event:

```
RESPONSE_CHOICES = (
    (0, '--'),
    (1, 'No'),
    (2, 'Maybe'),
    (3, 'Yes'),
    (4, 'Under Age'),
)
```

Note: You rsvp to an event by setting the `response` field of the party invite object to any of the integer values in `RESPONSE_CHOICES` above.

PERSONALITY AND WINE RATINGS

Every user that participates in a Vinely Tasting event or Joins the VIP club can have their Personalities revealed by having their wine ratings for the 6 wines filled in.

4.1 Wine Personality

There are 6 different Wine Personalities and the mystery personality (for those that have not had their personality revealed yet).

Fetch a list of the Wine Personalities defined by Vinely.

GET `/api/v1/personality/`

Get information about a specific wine personality.

GET `/api/v1/personality/{id}/`

4.2 Wine Ratings

Each wine has a number of different values that need to be rated.

The ratings choices are:

```
LIKENESS_CHOICES = (  
    (1, 'Hate'),  
    (2, 'Dislike'),  
    (3, 'Neutral'),  
    (4, 'Like'),  
    (5, 'Love'),  
)
```

```
DNL_CHOICES = (  
    (1, 'Too Little'),  
    (2, 'Just Right'),  
    (3, 'Too Much'),  
)
```

```
SWEET_CHOICES = (  
    (1, 'Tart'),  
    (2, 'Semi Tart'),  
    (3, 'Neutral'),  
    (4, 'Semi Sweet'),  
    (5, 'Sweet'),  
)
```

```
)

WEIGHT_CHOICES = (
    (1, 'Light'),
    (2, 'Semi Light'),
    (3, 'Medium'),
    (4, 'Semi Heavy'),
    (5, 'Heavy'),
)

TEXTURE_CHOICES = (
    (1, 'Silky'),
    (2, 'Semi Silky'),
    (3, 'Neutral'),
    (4, 'Semi Furry'),
    (5, 'Furry'),
)

SIZZLE_CHOICES = (
    (1, 'None'),
    (2, 'Somewhat'),
    (3, 'Tingle'),
    (4, 'Semi Burn'),
    (5, 'Burn')
)
```

Fetch a list of Wine rating data for the user currently logged in. It should return up to 6 objects for the 6 wines that were tasted

GET /api/v1/rating/

Get information about a particular wine rating

GET /api/v1/rating/{id}/

Update wine rating information

PUT /api/v1/rating/{id}/

Create a new wine rating

POST /api/v1/rating/

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

HTTP ROUTING TABLE

/api

```
POST /api/v1/auth/login/,3
POST /api/v1/auth/logout/,3
POST /api/v1/auth/signup/,3
GET /api/v1/event/,7
GET /api/v1/event/{id}/,7
GET /api/v1/party/,7
GET /api/v1/party/{id}/,7
GET /api/v1/partyinvite/,7
POST /api/v1/partyinvite/,8
GET /api/v1/partyinvite/{id}/,7
PUT /api/v1/partyinvite/{id}/,8
GET /api/v1/personality/,9
GET /api/v1/personality/{id}/,9
GET /api/v1/profile/,5
GET /api/v1/profile/{id}/,5
PUT /api/v1/profile/{id}/,6
GET /api/v1/rating/,10
POST /api/v1/rating/,10
GET /api/v1/rating/{id}/,10
PUT /api/v1/rating/{id}/,10
GET /api/v1/user/,5
PUT /api/v1/user/{id}/,5
```