

# API Reference Guide



Plantalytics Team  
Portland State University Computer Science Capstone

August 29<sup>th</sup>, 2016

# Table of Contents

1 Introduction.....	2
2 Errors.....	2
2.1 Error Codes.....	2
2.2 Error Responses.....	2
3 Endpoints.....	3
3.1 /admin/user.....	3
3.2 /admin/user/disable.....	4
3.3 /admin/user/edit.....	5
3.4 /admin/user/new.....	6
3.5 /admin/user/subscription.....	7
3.6 /admin/vineyard.....	8
3.7 /admin/vineyard/disable.....	9
3.8 /admin/vineyard/edit.....	10
3.9 /admin/vineyard/new.....	11
3.10 /email_change.....	12
3.11 /env_data.....	13
3.12 /health_check.....	14
3.13 /hub_data.....	15
3.14 /login.....	16
3.15 /password/change.....	17
3.16 /password/reset.....	18
3.17 /vineyard.....	19

# 1 Introduction

The Plantalytics backend API provides an abstraction layer to the user, vineyard, and environmental data. There are a number of endpoints available to provide administration for users, such as creating and updating users, to allow users to manage their own accounts, such as changing their password and logging in to their dashboard, and to provide environmental data for visualization. The API has been developed to accept a number of different JSON payloads and respond to requests in JSON format. Payloads structures for specific endpoints are detailed in this guide.

## 2 Errors

The API has been outfitted with particular response codes for expected errors. In addition, custom error messages are formatted with a particular response structure. These features allow the requesting entity to anticipate errors and perform appropriate actions when errors are encountered.

### 2.1 Error Codes

Expected errors encountered while serving a request respond with the following codes:

**Table 2.1: Response codes for expected errors.**

Response Code	Origin
400	<ul style="list-style-type: none"><li>Required argument is missing.</li></ul>
403	<ul style="list-style-type: none"><li>User is not logged in or does not have appropriate permissions.</li></ul>

### 2.2 Error Responses

Contained within the JSON response object is an error entry containing an object of the form:

**Figure 2.2: Example JSON response for error codes.**

```
{
  "errors": {
    "<error_code>": "English error message. (string)",
  }
}
```

## 3 Endpoints

The API endpoints provide an abstraction layer to the database. Request bodies are expected to be in JSON format and follow the structures provided in the following sections. In addition, response bodies are provided in JSON format.

Any required request fields are indicated with an asterisk, similar to the following example:

**Figure 3: Example JSON structure for a required request field.**

```
{
  "<a_field>": <some required field (string)>*
}
```

### 3.1 /admin/user

Retrieves user information.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.1: Payload structures for /admin/user endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "request_username": &lt;username to retrieve info for (string)&gt;*,   "auth_token": &lt;authentication token (string)&gt;* }</pre>
Response Body
<pre>{   "vineyards": [     &lt;vineyard id (integer)&gt;,   ],   "email": &lt;email address (string)&gt;,   "user_id": &lt;user id (integer)&gt;, }</pre>

```

    "is_admin": <indicates if admin (boolean)>,
    "is_enable": <indicates if user account is enabled (boolean)>,
    "sub_end_date": <subscription end date (string)>
}

```

## 3.2 /admin/user/disable

Disables a user. User will no longer be allowed to login, auth tokens will be disabled, but associated data will remain.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.2: Payload structures for /admin/user/disable endpoint response and request bodies.**

Payloads	
Request Body	
<pre> {     "auth_token": &lt;authentication token (string)&gt;,*     "request_username": &lt;username to disable (string)&gt;* } </pre>	
Response Body	
<pre> {     "errors": {} } </pre>	

### 3.3 /admin/user/edit

Edits an existing user in the database.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.3: Payload structures for /admin/user/edit endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "edit_user_info": {     "username": &lt;existing username (string)&gt;,*     "password": &lt;password (string)&gt;,     "email": &lt;email address (string)&gt;,     "admin": &lt;set as admin (boolean)&gt;,     "enable": &lt;enable the user (boolean)&gt;,     "subenddate": &lt;subscription end date (string)&gt;,     "userid": &lt;user id (integer)&gt;,     "vineyards": [       &lt;vineyard id (integer)&gt;,     ]   } }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.4 /admin/user/new

Creates a new user in the database.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.4: Payload structures for /admin/user/new endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "new_user_info": {     "username": &lt;username (string)&gt;,*     "password": &lt;password (string)&gt;,*     "email": &lt;email address (string)&gt;,*     "admin": &lt;set as admin (boolean)&gt;,*     "enable": &lt;enable the user (boolean)&gt;,*     "subenddate": &lt;subscription end date (string)&gt;,*     "userid": &lt;user id (integer)&gt;,*     "vineyards": [       &lt;vineyard id (integer)&gt;,*     ]*   }* }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.5 /admin/user/subscription

Update the subscription end date for a user.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.5: Payload structures for /admin/user/subscription endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "request_username": &lt;username to update (string)&gt;,*   "sub_end_date": &lt;new subscription end date (string)&gt;* }</pre>
Response Body
<pre>{   "errors": {} }</pre>



### 3.6 /admin/vineyard

Retrieves extra vineyard information like the owner(s) and other authenticated users.  
/admin APIs must be called by an admin who is logged in.

**Methods :** POST  
**Content Type:** application/json

Table 3.6: Payload structures for /admin/vineyard endpoint response and request bodies.

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "vineyard_id": &lt;vineyard id (integer)&gt;* }</pre>
Response Body
<pre>{   "name": &lt;vineyard name (string)&gt;,   "is_enable": &lt;indicates if the vineyard is enabled(boolean)&gt;,   "owners": [     &lt;owner name (string)&gt;,   ],   "users": [     &lt;user name (string)&gt;,   ] }</pre>

### 3.7 /admin/vineyard/disable

Disables a vineyard. Vineyard will no longer be accessible but associated data will remain.  
/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.7: Payload structures for /admin/vineyard/disable endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "vineyard_id": &lt;vineyard id (integer)&gt;* }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.8 /admin/vineyard/edit

Edits an existing vineyard in the database.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.1: Payload structures for /admin/vineyard/edit endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "edit_vineyard_info": {     "vineyard_id": &lt;vineyard id (integer)&gt;,*     "enable": &lt;enable the vineyard (boolean)&gt;,     "name": &lt;vineyard name (string)&gt;,     "owners": [       &lt;owner name (string)&gt;,     ],     "boundaries": [       {         "lat": &lt;latitude coordinate (float)&gt;,         "lon": &lt;longitude coordinate (float)&gt;       },     ],     "center": {       "lat": &lt;latitude coordinate (float)&gt;,       "lon": &lt;longitude coordinate (float)&gt;     }   } }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.9 /admin/vineyard/new

Creates a new vineyard in the database.

/admin APIs must be called by an admin who is logged in.

**Methods : POST**

**Content Type: application/json**

**Table 3.9: Payload structures for /admin/vineyard/new endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "new_vineyard_info": {     "vineyard_id": &lt;vineyard id (integer)&gt;,*     "name": &lt;vineyard name (string)&gt;,*     "enable": &lt;enable the vineyard (boolean)&gt;,*     "owners": [       &lt;owner name (string)&gt;,*     ],*     "boundaries": [       {         "lat": &lt;latitude coordinate (float)&gt;,*         "lon": &lt;longitude coordinate (float)&gt;*       },     ],*     "center": {       "lat": &lt;latitude coordinate (float)&gt;,*       "lon": &lt;longitude coordinate (float)&gt;*     }*   }* }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.10 /email\_change

Changes the email address in the database of an existing (currently logged in) user.

**Methods : POST**

**Content Type: application/json**

**Table 3.10: Payload structures for /email\_change endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;*,   "new_email": &lt;new email address (string)&gt;* }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.11 /env\_data

Retrieves most recent environmental data.

Currently supported environmental variables:

- "temperature"
- "humidity"
- "leafwetness"

**Methods : POST**

**Content Type: application/json**

**Table 3.11: Payload structures for /env\_data endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "vineyard_id": &lt;vineyard id (integer)&gt;,*   "env_variable": &lt;environmental variable (string)&gt;,*   "auth_token": &lt;authentication token (string)&gt;* }</pre>
Response Body
<pre>{   "env_data": [     {       "&lt;environmental variable&gt;": &lt;value (float)&gt;,       "latitude": &lt;value (float)&gt;,       "longitude": &lt;value (float)&gt;,     },   ] }</pre>

## 3.12 /health\_check

A simple endpoint to indicate that the backend is operational.

**Methods : GET**

**Table 3.12: Payload structures for /health\_check endpoint response and request bodies.**

Payloads
Response Body
<pre>{   "isAlive": true }</pre>

## 3.13 /hub\_data

Stores data supplied by a hub into the database.

**Methods :** PUT, POST

**Content Type:** application/json

**Table 3.13: Payload structures for /hub\_data endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "key": &lt;unique value to validate a hub (string)&gt;,*   "vine_id": &lt;vineyard id (integer)&gt;,*   "hub_id": &lt;hub id (integer)&gt;,*   "hub_data": [     {       "node_id": &lt;node id (integer)&gt;,*       "temperature": &lt;value (float)&gt;,*       "humidity": &lt;value (float)&gt;,*       "leafwetness": &lt;value (float)&gt;,*       "data_sent": &lt;sent date/time from node(integer)&gt;*     },   ],   "batch_sent": &lt;date and time of transmission from hub (integer)&gt;* }</pre>
Response Body
<pre>{   "errors": {} }</pre>



## 3.14 /login

Verifies a user's credentials and supplies them with an authentication token. Also retrieves the vineyards they are authorized to view.

**Methods : POST**

**Content Type: application/json**

**Table 3.14: Payload structures for /login endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "username": &lt;username (string)&gt;*,   "password": &lt;password (string)&gt;* }</pre>
Response Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,   "authorized_vineyards": [     {       "vineyard_id": &lt;vineyard id (integer)&gt;,       "vineyard_name": &lt;vineyard name (string)&gt;,     },   ] }</pre>

## 3.15 /password/change

Changes the password for an existing user.

NOTE: “token” is different from the “auth\_token”.

There are 3 use cases anticipated:

- 1) The current user is logged in and trying to change their own password.
  - a) An “old” password, new “password” & a valid “auth\_token” must be given.
  - b) The “token” and “username” are ignored.
- 2) The current user is the administrator.
  - a) The admin’s valid “auth\_token”, the “username”, and the new “password” must be supplied.
  - b) “old” and “token” are ignored.
- 3) The user forgot the password and has used the /reset endpoint below.
  - a) a “username” must be specified and a matching password reset “token” must be given, as well as a new “password”
  - b) The “old” and “auth\_token” will be ignored.

**Methods : POST**

**Content Type: application/json**

**Table 3.15: Payload structures for /password/change endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "username": &lt;username (string)&gt;,*   "password": &lt;new password (string)&gt;,*   "old": &lt;old password (string)&gt;,   "token": &lt;password reset token (string)&gt;,   "auth_token": &lt;auth token given with every POST (string)&gt; }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.16 /password/reset

Generates a reset password token and sends email to user's stored address.

Request a password reset email for submitted username. Password will not be changed at the time of request, it will simply generate a password reset token and dispatch an email to the user, allowing the user to continue the process.

**Methods : POST**

**Content Type: application/json**

**Table 3.16: Payload structures for /password/reset endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "username": &lt;username (string)&gt;* }</pre>
Response Body
<pre>{   "errors": {} }</pre>

## 3.17 /vineyard

Retrieves the center point and boundary points of an existing vineyard.

**Methods : POST**

**Content Type: application/json**

**Table 3.17: Payload structures for /vineyard endpoint response and request bodies.**

Payloads
Request Body
<pre>{   "auth_token": &lt;authentication token (string)&gt;,*   "vineyard_id": &lt;vineyard id (integer)&gt;* }</pre>
Response Body
<pre>{   "boundary": [     {       "lat": &lt;latitude coordinate (float)&gt;,       "lon": &lt;longitude coordinate (float)&gt;     },   ],   "center": {     "lat": &lt;latitude coordinate (float)&gt;,     "lon": &lt;longitude coordinate (float)&gt;   } }</pre>