

User api v1.0.0

last modified: 2016-10-05

The purpose of this document is to describe the details for integrating with the user api, which stores user accounts and related data.

Base URL:

development: https://staging-api.example.com

production: https://api.example.com

All request should be made relative to this base url.

HTTP Basic:

All requests will need to supply http basic authentication credentials. These credentials allow for access through to the api and are in no way related to individual user authentication.

The credentials are as follows for each environment:

development:

username: devUsername

password: \$*rs9D(

production:

username: prodUsername

password: &&KeXt97&sd

API Endpoints:

Create User:

Description:

This method creates a new user.

HTTP Method: POST

Allowed Content-Type values: application/x-www-form-urlencoded or application/json

URL: {base_url}/user

Parameters:

email – string, user's email address

password – string, user's password

confirm_password – string, password confirmation

first_name – string, user's first name

last_name – string, user's last name

street_number – int, user's street number

apartment_number – (optional), int, user's apartment number
street_name – string, user's street name
city – string, user's city
state – string, user's state (2 letter abbreviation)

Response:

The following http codes are possible:

201 – success, account created
400 – invalid parameters supplied, do not submit with the same parameters
401 – permission denied, invalid http basic credentials
409 – user already exists
500 – internal server error

The response body will be an application/json object and the member variables will vary depending on the http code returned.

For non-201 http codes (unsuccessful):

error – string indicating the error

For 201 http codes (success):

user_id – unique id for the new user

Get User:

Description:

This method returns user data for an individual user.

HTTP Method: GET

Allowed Content-Type values: N/A

URL: {base_url}/user/{user_id}

Parameters:

user_id – int, unique id for a user

Response:

The following http codes are possible:

200 – success, user data returned successfully
401 – permission denied, invalid http basic credentials
404 – user not found
500 – internal server error

The response body will be an application/json object and the member variables will vary depending on the http code returned.

For non-200 http codes (unsuccessful):

error – string indicating the error

For 200 http codes (success):

account – object with the following member variables:

email – string, user's email address
password – string, user's password
first_name – string, user's first name
last_name – string, user's last name
street_number – int, user's street number
apartment_number – int, user's apartment number
street_name – string, user's street name
city – string, user's city
state – string, user's state (2 letter abbreviation)

Delete User:

Description:

This method deletes and exiting user.

HTTP Method: DELETE

Allowed Content-Type values: N/A

URL: {base_url}/user/{user_id}

Parameters:

user_id – int, unique id for a user

Response:

The following http codes are possible:

204 – user deleted successfully

401 – permission denied, invalid http basic credentials

404 – user not found

500 – internal server error

The response body will be an application/json object (or empty) and the member variables will vary depending on the http code returned.

For non-204 http codes (unsuccessful):

error – string indicating the error

For 204 http codes (success):

no body

Update User:

Description:

This method updates an existing user account.

HTTP Method: PUT

Allowed Content-Type values: application/x-www-form-urlencoded or application/json

URL: {base_url}/user/{user_id}

Parameters:

user_id – existing unique id for a user

email – (optional) string, user's email address

password – (optional) string, user's password

confirm_password – (optional) string password confirmation, required when updating user's password

first_name – (optional) string, user's first name

last_name – (optional) string, user's last name

street_number – (optional) int, user's street number

apartment_number – (optional), int, user's apartment number

street_name – (optional) string, user's street name

city – (optional) string, user's city

state – (optional) string, user's state (2 letter abbreviation)

Response:

The following http codes are possible:

204 – success, account updated

400 – invalid parameters supplied, do not submit with the same parameters

401 – permission denied, invalid http basic credentials

404 – user not found

500 – internal server error

The response body will be an application/json object (or empty) and the member variables will vary depending on the http code returned.

For non-204 http codes (unsuccessful):

error – string indicating the error

For 204 http codes (success):

no body