## *Assume all POST parameters and responses are JSON-encoded*

## Error Response: { ‘error’: ‘…’ }

# User Accounts and Friends

## **POST** /register/

### Required Parameters:

* ‘email’ (string)
* ‘password’ (MD5 string)

On Success:

* { }
* User is sent an email with an /activate link

### Possible Errors:

* BadRequest if email or password absent
* Malformed email address
* Email is already registered

## **GET** /activate/<token>

### URL is sent as a link in activation email

On Success:

* activated (string) - the email address verified

Possible Errors:

* BadRequest if token not found
* Account already activated

## **POST** /login/

Required Parameters:

* ‘email’ (string)
* ‘password’ (MD5 string)

On Success:

* { }

### Possible Errors:

* BadRequest if email or password absent
* Malformed email address
* User nonexistent
* Account not activated
* Wrong password

## **GET** /logout

### Logs out the current user if possible

### Always:

* { }

## **GET** /friends/list

On Success:

* ‘friends’ (list; int) - list of users who have reciprocated your friend request

### Possible Errors:

* Not logged in

## **GET** /friends/list/<user>

Required Parameters:

* ‘user’ (integer)

On Success:

* ‘friends’ (list; int) - list of users who are friends of the specified user

### Possible Errors:

* Not logged in
* Not friends with user

## **GET** /friends/list/friends

On Success:

* ‘friends\_of\_friends’ (list; int) - list of users who are friends of the your friends

### Possible Errors:

* Not logged in

## **GET** /friends/pending

On Success:

* ‘pending’ (list; int) - list of users whom you have added, but have not accepted yet

### Possible Errors:

* Not logged in

## **GET** /friends/add/<user>

Required Parameters:

* ‘user’ (integer)

On Success:

* { }

### Possible Errors:

* Not logged in
* Cannot add yourself as a friend
* User does not exist

## **GET** /friends/remove/<user>

Required Parameters:

* ‘user’ (integer)

On Success:

* { }

### Possible Errors:

* Not logged in

# Content and Ratings

## **GET** /content/<type>/<uri>

Required Parameters:

* ‘type’ (string) - one of: track, album, artist, playlist
* ‘uri’ - (string) - the content’s Spotify ID, for example 2noRn2Aes5aoNVsU6iWThc

On Success:

* ‘content\_id’ (integer) - an ID number that can be used to identify the content later
* ‘name’ (string) - the content’s name as returned by Spotify’s API
* ‘created’ (boolean) - whether the content record was created by this request

### Possible Errors:

* Not logged in
* Not authenticated with Spotify
* Spotify API errors as a result of fetching the content

## **GET** /content/<id>

Required Parameters:

* ‘id’ (integer) - content ID

On Success:

* ‘type’ (string) - one of: track, album, artist, playlist
* ‘uri’ (string) - the content’s Spotify ID
* ‘name’ (string) - the content’s name as returned by Spotify’s API

### Possible Errors:

* Not logged in
* Content ID not found

## **GET** /content/<id>/rating

Required Parameters:

* ‘id’ (integer) - content ID

On Success:

* ‘rating’ (integer) - your numeric rating of this piece of content

### Possible Errors:

* Not logged in
* Rating not found

## **GET** /content/<id>/rating/<user>

Required Parameters:

* ‘id’ (integer) - content ID
* ‘user’ (integer)

On Success:

* ‘rating’ (integer) - user’s numeric rating of this piece of content

### Possible Errors:

* Not logged in
* Not friends with user
* Rating not found

## **GET** /content/<id>/rating/set/<rating>

Required Parameters:

* ‘id’ (integer) - content ID
* ‘rating’ (integer) - in range: 0 <= n <= 10

On Success:

* { }

### Possible Errors:

* Not logged in
* Content ID not found
* Rating out of range

## **GET** /content/<id>/rating/reset

Required Parameters:

* ‘id’ (integer) - content ID

On Success:

* { }

### Possible Errors:

* Not logged in
* Content ID not found

## **GET** /ratings/list

On Success:

* ratings (list; each element contains
  + - ‘content\_id’ (integer)
    - ‘rating’ (integer) )

### Possible Errors:

* Not logged in

## **GET** /ratings/list/<user>

Required Parameters:

* ‘user’ (integer)

On Success:

* ratings (list; each element contains
  + - ‘content\_id’ (integer)
    - ‘rating’ (integer) )

### Possible Errors:

* Not logged in
* Not friends with user

## **GET** /ratings/list/friends

On Success:

* ratings (list; each element contains
  + - ‘friend\_id’ (integer)
    - ‘content\_id’ (integer)
    - ‘rating’ (integer) )

### Possible Errors:

* Not logged in

# Spotify Authentication

## **GET** /spotify/auth

### Redirects to Spotify login flow

Possible Errors:

* Not logged in

## **GET** /spotify/auth/callback

### Sets the user’s auth and refresh tokens if successful

### Redirected to by /spotify/auth by Spotify after the user logs in there

Possible Errors:

* Not logged in

# Spotify Data

For All, Possible Errors:

* Not logged in
* Not authenticated with Spotify
* Problem accessing Spotify API

Parameters passed in GET request

## **GET** /spotify/user/profile

## **GET** /spotify/user/playing\_track

## **GET** /spotify/user/recent\_tracks

Optional Parameters:

* limit
* before
* after

## **GET** /spotify/user/top\_tracks

Optional Parameters:

* limit
* offset
* timespan

## **GET** /spotify/user/followed\_artists

Optional Parameters:

* limit
* after

## **GET** /spotify/user/playlists

Optional Parameters:

* limit
* offset

## **GET** /spotify/user/saved\_albums

Optional Parameters:

* limit
* offset

## **GET** /spotify/user/saved\_tracks

Optional Parameters:

* limit
* offset

## **GET** /spotify/user/search

Required Parameters:

* q - search query

Optional Parameters:

* limit
* offset
* type

## **GET** /spotify/user/user\_playlists

Required Parameters:

* user

Optional Parameters:

* limit
* offset

## **GET** /spotify/user/fetch\_tracks

Required Parameters:

* tracks - some Spotify track URIs, comma separated

## **GET** /spotify/user/fetch\_albums

Required Parameters:

* albums - some Spotify album URIs, comma separated

## **GET** /spotify/user/fetch\_artists

Required Parameters:

* artists - some Spotify artist URIs, comma separated

## **GET** /spotify/user/create\_playlist

Required Parameters:

* name (string)

Optional Parameters:

* description (string)

## **GET** /spotify/user/add\_playlist\_custom\_image

Required Parameters:

* playlist
* image\_b64

## **GET** /spotify/user/follow\_playlist

Required Parameters:

* playlist

## **GET** /spotify/user/is\_following\_playlist

Required Parameters:

* playlist
* users - some Spotify user IDs, comma separated

## **GET** /spotify/user/add\_playlist\_tracks

Required Parameters:

* playlist
* tracks - some Spotify track URIs, comma separated

Optional Parameters:

* position (integer)

## **GET** /spotify/user/edit\_playlist\_details

Required Parameters:

* playlist

Optional Parameters:

* name (string)
* description (string)