
LoginRadius Documentation

Release 2.0

LoginRadius

November 14, 2014

1	Introduction	3
2	Quickstart guide	5
2.1	Install	5
2.2	Using the LoginRadius Object	6
2.3	Breaking Down the LoginRadius Object	6
3	Exceptions	9
3.1	Handling Exceptions	9
4	Examples and Use Cases	11
4.1	Accessing Raw Provider Data	11
4.2	Preloading Data	11
4.3	Changing HTTP Libraries	12
5	Reference Manual	13
5.1	Data Points	13
5.2	REST API	13
5.3	LoginRadius Object	13
6	Feedback	17
7	Indices and tables	19
	Python Module Index	21

Release 2.0

Date November 14, 2014

Contents:

Introduction

LoginRadius offers a complete social infrastructure solution combining 30 major social platforms into one unified API. With LoginRadius, websites and mobile apps can integrate social login, enable social sharing, capture user profiles and social data, create a single sign-on experience for their users, and get comprehensive social analytics. Our social solution helps websites engage, understand, and leverage their users.

This module provides a wrapper for `urllib2` or the `requests` library to easily access the API from <https://api.loginradius.com/> in a more “pythonic” way. Providing easier access to essential data in a few lines of code. This will work with 1.0 or 2.0 API specifications.

For more information, visit: <http://loginradius.com/>

Quickstart guide

This guide should help you install and utilize LoginRadius and its feature set quickly and painlessly. For more examples, please see examples.

2.1 Install

2.1.1 Prerequisites

You will need at least Python - 2.7 or greater. LoginRadius module utilizes the `namedtuple` from the collections library and the `import_module` from importlib.

2.1.2 From Package

Using pip

```
pip install loginradius
```

or with easy_install

```
easy_install loginradius
```

2.1.3 From Source

You can download the latest version from [PyPI](#)

- Unzip/untar the files.
- Browse to the directory that you extracted the files to.
- Run:

```
python setup.py install
```

To install the LoginRadius module.

2.2 Using the LoginRadius Object

import the class

```
from LoginRadius import LoginRadius
```

When you initialise your application, you will need to set your API Secret. This can be found in your control panel under your dashboard:

<https://secure.loginradius.com/account#dashboard>

When your Python application first starts up, you should set your API Secret once and only once

```
LoginRadius.API_SECRET = "some-really-obscure-string"
```

When you receive a callback from LoginRadius after a user has authenticated with their provider it will come with a token parameter. For example, with python in CGI we can do:

```
import cgi
arguments = cgi.FieldStorage()
token = arguments.getvalue('token')
```

This is highly dependent on your framework. You should browse some of our *Examples and Use Cases* for one that fits with your web framework and code flow.

We can now construct a LoginRadius object with respect to this token.

```
login = LoginRadius(token)
```

Doing so will make an HTTP request to LoginRadius for the access token. Not to be confused with the token we just received from our callback. This new token is the basis for all API calls to LoginRadius.

With our newly constructed object, we can perform any API call listed in the online API help document. As long as your plan is setup to do so! For example, the basic profile data is available to anyone.

<http://api.loginradius.com/help/>

We can access the profile information from the login object as a namedtuple.

```
if login.access.valid():
    print ("Hello there, " + login.user.profile["FirstName"] + ".")
```

We first check to see if our token hasn't expired. Be wary that storing a LoginRadius object for too long may cause the token to become invalid. The default time, at the time of writing this guide, is fifteen minutes.

The next line of code embedded in the if statement prints the user's first name.

This data is normalized across all data providers and is stored as a python dictionary to adapt to the API if changes are made at a later date. To view all up to date items available to you in the dictionary, visit:

<http://www.loginradius.com/datapoints/>

The thought process behind the LoginRadius object is to provide easily accessible functions with minimal arguments. It only loads data when needed.

2.3 Breaking Down the LoginRadius Object

Like our previous example, profile is not the only attribute available to you.

api

This should only be used in cases where you want direct access to the LoginRadiusAPI object without our wrapper. It is highly advised against using this attribute at all.

The api attribute is class of: `LoginRadius.LoginRadiusAPI`

access

Contains useful information about the access token.

expire

A string containing the date when this access token is no longer valid.

raw

A dictionary containing the raw JSON response from LoginRadius for the token.

token

This is our access token. Essentially for every API call.

valid()

Invoking this will return a True or False boolean for if the access token is still valid. It is useful to check before making an API call. This method will only compare the stored expired attribute date and current UTC time. It will not make a remote call to the server to validate.

error

A string of the last error message encountered.

settings

HTTP library settings for encoding requests.

library

Stores a string of the HTTP library currently being used with the LoginRadius object. Currently, this value can either be 'urllib2' or 'requests'. Please call `_settings(library)` and not this if you intend to change it.

urllib

imported module of urllib.

urllib2

imported module of urllib2.

json

imported module of json.

requests

imported module of requests

token

The original token from the callback.

user

Some of these attributes may not pull any data. Please check your endpoints at <http://www.loginradius.com/datapoints/>

All of these attributes, by default, set raw=False to normalize data.

profile

Information from the user's profile with respect to the current provider.

photo

Photos with regards to the album_id attribute.

album_id

A string of the album_id to be set before retrieval.

check_in

Get a list of checked in places from this profile.

album

Get a list of albums available on this profile.

audio

Get a list of audio available on this profile.

video

Get a list of uploaded videos on this profile

contacts

Get a list of contacts, you can also specify a `next_cursor` attribute.

next_cursor

An optional string for the next cursor.

status

Get a list of status updated on this profile.

group

Get a list of groups from this profile.

event

Get a list of events from this profile.

mention

Get a list of mentions from this profile.

activity

Get a list of activities from this profile.

following

Get a list of following from this profile.

page

Get page data with respect to what this user's profile can see.

page_name

A string containing the unique identifier for the page to fetch data from.

like

Get a list of likes from this profile.

status_update (*status*, *title*='', *url*='', *imageurl*='', *caption*='', *description*='')

Perform a status update on the profile on behalf of the user. Some of these arguments may be ignored depending on the provider.

direct_message (*to*, *subject*, *message*)

Direct message another user on behalf of this user.

Exceptions

3.1 Handling Exceptions

A list of specific exceptions are in `LoginRadius.Exceptions`

However, there is a catch all class from which all LoginRadius exceptions inherit from. `LoginRadius.LoginRadiusExceptions`

```
from LoginRadius import LoginRadiusExceptions

try:
    print login.user.profile['FirstName']

except LoginRadiusExceptions:
    print "Something went wrong!"
    print login.error
```

The error attribute to your LoginRadius object contains a string that may help you solve why this exceptions was raised in the first place.

Examples and Use Cases

You can find fully functional examples in `examples/` in the [repository](#).

4.1 Accessing Raw Provider Data

If you do not want normalized data, but instead the raw return with respect to the provider, you must enable it.

```
#Let's get their first name but with the raw response from Facebook
login = LoginRadius(token)
login.user.profile.set_raw(True)
print "Hello there," + login.user.profile['first_name']
```

Warning: This will of course depend on the provider each user logs in with, and it is only recommended for advanced users where normalized data is not applicable.
This will only set profile requests to raw, and not other attributes, such as status.

4.2 Preloading Data

To reduce load times, you probably don't want to make an external HTTP request while populating your templates. If you know what data you need beforehand, we should fetch it.

First we should understand what happens in a normal use case.

```
login = LoginRadius(token)

#Your code here

#This makes a GET request and will be blocking.
print "Hello there," + login.user.profile['FirstName']

#We already called profile, and all this data is already loaded into the dictionary.
#This is non-blocking now.
print "What can I do for you today, " + login.user.profile['FirstName'] + "?"
```

The first time you call any attribute will block and fetch the data remotely from LoginRadius. After doing so, it is saved with respect to the attribute of our object.

However, this is not always optimal. You could call this in a thread and prepare the response in the meantime. While threading is beyond the scope of this document, here is how you load the data preemptively.

```
login = LoginRadius(token)

#Let's make sure the data is available to us.
login.user.profile.load()

#Your code here.

#This makes a GET request and will be non-blocking now.
print "Hello there," + login.user.profile['FirstName']

#We already called profile, and all this data is already loaded into the dictionary.
#This is non-blocking now.
print "What can I do for you today, " + login.user.profile['FirstName'] + "?"
```

Invoking load on any user attribute will fetch the data.

4.3 Changing HTTP Libraries

By default, the requests module will be picked if it is version 2.0 or greater. However, you can override this and change it as you see fit.

Checking current library after constructing a LoginRadius object.

```
login = LoginRadius(token)
print login.settings.library
```

Changing the library for one LoginRadius object.

```
login = LoginRadius(token)

#Let's opt in for urllib2 instead of requests
login.change_library("urllib2")
```

Valid options are only 'urllib2' or 'requests'.

Changing the library for all future constructed LoginRadius objects.

```
from LoginRadius import LoginRadius

#Changing to the urllib2 library
LoginRadius.LIBRARY = "urllib2"
```

Reference Manual

5.1 Data Points

When you access the python dictionary with respect to each attribute of the object, it is parsing the information reflected in this document.

<http://www.loginradius.com/datapoints>

Warning: These data points are subject to change and vary with providers.

For example

```
print login.user.profile['NickName']
```

May work for some providers like Mixi, but will not work for Facebook. Instead, you would get a key value error.

5.2 REST API

You can take advantage of the raw API if you wish and learn how it works.

<http://api.loginradius.com/help/>

5.3 LoginRadius Object

class LoginRadius.**Exceptions**

Common exceptions used by LoginRadius.

exception APIKeyInvalid

Raised if you entered your API wrong, or not at all.

exception Exceptions.**APISecretInvalid**

Raised if you your API Secret is invalid.

exception Exceptions.**EndPointNotSupported**

Raised if a the endpoint is not supported by the provider which correlates to the token.

exception Exceptions.**FeatureNotSupported**

Raised if the request cannot be completed because your account/API access does not include this.

exception `Exceptions.InvalidAccessToken`

Raised if you access token is invalid.

exception `Exceptions.InvalidLibrary` (*library*)

Raised on trying to change library to through `_settings` on invalid library argument.

exception `Exceptions.InvalidRequestToken`

Raised if you your request token is invalid from the POST request.

exception `Exceptions.MissingJsonResponseParameter` (*missing_parameter, raw=None*)

Raised if construction of namedtuple would fail because missing expected response from LoginRadius API.

exception `Exceptions.NoAPISecret` (*version*)

Raised on construction of the LoginRadius object, if no `API_SECRET` has been set for the class.

exception `Exceptions.ParameterMissing`

Raised if a parameter in the GET or POST request is missing.

exception `Exceptions.ParameterNotFormatted`

Raised if a parameter in the GET or POST request is not formatted properly for the provider.

exception `Exceptions.RequestTokenExpired`

Raised if you your request token has expired from the POST request.

exception `Exceptions.RequestsLibraryDated` (*version*)

Raise exception if module requests is outdated. By default 0.12 is included in Python. We need at least version 2.0

exception `Exceptions.TokenExpired` (*time*)

Raised if the request cannot be completed because the access token has expired.

exception `Exceptions.UnknownJsonError` (*result*)

Raised if cannot determine error number from Json

class `LoginRadius.LazyLoad` (*lr_object, raw=False*)

“lazy load” the details when needed. This methodology is inspired by SQLAlchemy.

flush ()

Clears the local data stored so that the next request doesn’t default to the local cache and instead grabs it from the LoginRadius servers.

get ()

This will get the JSON API and return it as a dictionary.

load ()

Promptly load the data instead of loading it on first access.

set_raw (*state*)

Change the state of the raw parameter to determine what data set is grabbed from the provider.

class `LoginRadius.LoginRadius` (*token*)

LoginRadius Class. Use this to obtain social data and other information from the LoginRadius API. Requires Python 2.7 or greater.

class `LoginRadius.LoginRadiusAPI` (*lr_object*)

Where all the API commands can be invoked locally.

direct_message (*to, subject, message*)

Direct message another user on behalf of this user.

get_activity ()

Get activity from profile.

get_activity_raw()
Get activity from profile but in raw format based on provider.

get_album()
Get albums from profile.

get_album_raw()
Get albums from profile but in raw format based on provider.

get_audio()
Get audio from the profile.

get_audio_raw()
Get audio from the profile but in raw format based on provider.

get_checkin()
Get check ins from profile.

get_checkin_raw()
Get check ins but in raw format based on provider.

get_company()
Get company from profile.

get_company_raw()
Get company from profile but in raw format based on provider.

get_contacts (*next_cursor*='')
Get a list of contacts from the profile.

get_contacts_raw (*next_cursor*='')
Get a list of contacts from the profile but in raw format based on provider.

get_event()
Get events from profile.

get_event_raw()
Get events from profile but in raw format based on provider.

get_following()
Get following/followers from profile.

get_following_raw()
Get following/followers from profile but in raw format based on provider.

get_group()
Get group data from profile.

get_group_raw()
Get group data from profile but in raw format based on provider.

get_like()
Get likes from profile.

get_like_raw()
Get likes from profile but in raw format based on provider.

get_mention()
Get mentions from profile.

get_mention_raw()
Get mentions from profile but in raw format based on provider.

get_page (*page_name*)

Get page information from profile.

get_page_raw (*page_name*)

Get page information from profile but in raw format based on provider.

get_photo (*album_id*='')

Get photos based on the album_id retrieved.

get_photo_raw (*album_id*='')

Get photos based on the album_id retrieved but unformatted based on the provider.

get_post ()

Get posts from profile.

get_post_raw ()

Get posts from profile but in raw format based on provider.

get_status ()

Get status updates from profile.

get_status_raw ()

Get status updates from profile but in raw format based on provider.

get_user_profile ()

Retrieve basic profile information.

get_user_profile_raw ()

Retrieve basic profile information but unformatted based on the provider.

get_video ()

Get videos from the profile.

get_video_raw ()

Get videos from the profile but in raw format based on provider.

status_update (*status*, *title*='', *url*='', *imageurl*='', *caption*='', *description*='')

Perform a status update on the profile on behalf of the user. Some of these arguments may be ignored depending on the provider. For what is and is not supported, please refer to: <http://www.loginradius.com/datapoints>

exception LoginRadius.LoginRadiusExceptions

This is the base for all LoginRadius Exceptions. Makes dealing with exceptions easy!

Feedback

If you have any questions, concerns, or feature requests, please send an email to the maintainers: [developers \(at \) loginradius.com](mailto:developers@loginradius.com)

Indices and tables

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

I

LoginRadius, [13](#)

A

access, 7
activity, 8
album, 8
album_id, 7
api, 6
audio, 8

C

check_in, 7
contacts, 8

D

direct_message(), 8
direct_message() (LoginRadius.LoginRadiusAPI
method), 14

E

error, 7
event, 8
Exceptions (class in LoginRadius), 13
Exceptions.APIKeyInvalid, 13
Exceptions.APISecretInvalid, 13
Exceptions.EndPointNotSupported, 13
Exceptions.FeatureNotSupported, 13
Exceptions.InvalidAccessToken, 13
Exceptions.InvalidLibrary, 14
Exceptions.InvalidRequestToken, 14
Exceptions.MissingJsonResponseParameter, 14
Exceptions.NoAPISecret, 14
Exceptions.ParameterMissing, 14
Exceptions.ParameterNotFormatted, 14
Exceptions.RequestsLibraryDated, 14
Exceptions.RequestTokenExpired, 14
Exceptions.TokenExpired, 14
Exceptions.UnknownJsonError, 14
expire, 7

F

flush() (LoginRadius.LazyLoad method), 14

following, 8

G

get() (LoginRadius.LazyLoad method), 14
get_activity() (LoginRadius.LoginRadiusAPI method),
14
get_activity_raw() (LoginRadius.LoginRadiusAPI
method), 14
get_album() (LoginRadius.LoginRadiusAPI method), 15
get_album_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_audio() (LoginRadius.LoginRadiusAPI method), 15
get_audio_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_checkin() (LoginRadius.LoginRadiusAPI method),
15
get_checkin_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_company() (LoginRadius.LoginRadiusAPI method),
15
get_company_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_contacts() (LoginRadius.LoginRadiusAPI method),
15
get_contacts_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_event() (LoginRadius.LoginRadiusAPI method), 15
get_event_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_following() (LoginRadius.LoginRadiusAPI method),
15
get_following_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_group() (LoginRadius.LoginRadiusAPI method), 15
get_group_raw() (LoginRadius.LoginRadiusAPI
method), 15
get_like() (LoginRadius.LoginRadiusAPI method), 15
get_like_raw() (LoginRadius.LoginRadiusAPI method),
15
get_mention() (LoginRadius.LoginRadiusAPI method),
15

`get_mention_raw()` (LoginRadius.LoginRadiusAPI method), 15

`get_page()` (LoginRadius.LoginRadiusAPI method), 15

`get_page_raw()` (LoginRadius.LoginRadiusAPI method), 16

`get_photo()` (LoginRadius.LoginRadiusAPI method), 16

`get_photo_raw()` (LoginRadius.LoginRadiusAPI method), 16

`get_post()` (LoginRadius.LoginRadiusAPI method), 16

`get_post_raw()` (LoginRadius.LoginRadiusAPI method), 16

`get_status()` (LoginRadius.LoginRadiusAPI method), 16

`get_status_raw()` (LoginRadius.LoginRadiusAPI method), 16

`get_user_profile()` (LoginRadius.LoginRadiusAPI method), 16

`get_user_profile_raw()` (LoginRadius.LoginRadiusAPI method), 16

`get_video()` (LoginRadius.LoginRadiusAPI method), 16

`get_video_raw()` (LoginRadius.LoginRadiusAPI method), 16

`group`, 8

J

`json`, 7

L

`LazyLoad` (class in LoginRadius), 14

`library`, 7

`like`, 8

`load()` (LoginRadius.LazyLoad method), 14

`LoginRadius` (class in LoginRadius), 14

`LoginRadius` (module), 13

`LoginRadiusAPI` (class in LoginRadius), 14

`LoginRadiusExceptions`, 16

M

`mention`, 8

N

`next_cursor`, 8

P

`page`, 8

`page_name`, 8

`photo`, 7

`profile`, 7

R

`raw`, 7

`requests`, 7

S

`set_raw()` (LoginRadius.LazyLoad method), 14

`settings`, 7

`status`, 8

`status_update()`, 8

`status_update()` (LoginRadius.LoginRadiusAPI method), 16

T

`token`, 7

U

`urllib`, 7

`urllib2`, 7

`user`, 7

V

`valid()`, 7

`video`, 8