Rocket Python Documentation

Release 1.3.3

Derek Stegelman

Mar 24, 2023

Contents

1 Installation 3

2 Main API Documentation 5

3 Detailed Module Docs 15

Python Module Index 17

Index 19

Rocket Python is a Python 2 and 3 library to access the REST API in RocketChat instances. The goal is to implement the entire REST API provided by RocketChat.

CHAPTER 1

Installation

First step is to install via pip:

```
pip install rocket-python
```

You'll need to create an instance of the Rocket API by logging in:

```
from rocketchat.api import RocketChatAPI

 api = RocketChatAPI(settings={'username': 'someuser', 'password': 'somepassword',
 'domain': 'https://myrockethchatdomain.com'})
```

CHAPTER 2

Main API Documentation

2.1 rocketchat package

2.1.1 Subpackages

rocketchat.calls package

Subpackages

rocketchat.calls.auth package

Submodules

rocketchat.calls.auth.get_me module

class rocketchat.calls.auth.get_me.**GetMe**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/me'

post_response(*result*)

Module contents

rocketchat.calls.channels package

Submodules

rocketchat.calls.channels.create_public_room module

class rocketchat.calls.channels.create_public_room.**CreatePublicRoom**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.PostMixin*, *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_payload(***kwargs*)

Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

endpoint = '/api/v1/channels.create'

post_response(*result*)

rocketchat.calls.channels.delete_public_room module

class rocketchat.calls.channels.delete_public_room.**DeletePublicRoom**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.PostMixin*, *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_payload(***kwargs*)

Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

endpoint = '/api/v1/channels.delete'

post_response(*result*)

rocketchat.calls.channels.get_history module

class rocketchat.calls.channels.get_history.**GetRoomHistory**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/channels.history'

post_response(*result*)

rocketchat.calls.channels.get_public_rooms module

class rocketchat.calls.channels.get_public_rooms.**GetPublicRooms**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint()

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/channels.list'

post_response(*result*)

rocketchat.calls.channels.get_room_info module

class rocketchat.calls.channels.get_room_info.**GetRoomInfo**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/channels.info'

post_response(*result*)

Module contents

rocketchat.calls.chat package

Submodules

rocketchat.calls.chat.send_message module

class rocketchat.calls.chat.send_message.**SendMessage**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.PostMixin*, *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_payload(***kwargs*)

Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

endpoint = '/api/v1/chat.postMessage'

Module contents

rocketchat.calls.groups package

Submodules

rocketchat.calls.groups.get_private_room_history module

class rocketchat.calls.groups.get_private_room_history.**GetPrivateRoomHistory**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/groups.history'

post_response(*result*)

rocketchat.calls.groups.get_private_room_info module

class rocketchat.calls.groups.get_private_room_info.**GetPrivateRoomInfo**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/groups.info'

post_response(*result*)

rocketchat.calls.groups.get_private_rooms module

class rocketchat.calls.groups.get_private_rooms.**GetPrivateRooms**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint()

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/groups.list'

post_response(*result*)

rocketchat.calls.groups.get_room_id module

class rocketchat.calls.groups.get_room_id.**GetRoomId**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/groups.info'

post_response(*result*)

rocketchat.calls.groups.upload_file module

class rocketchat.calls.groups.upload_file.**UploadFile**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.PostMixin*, *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_files(***kwargs*)

Build files :param kwargs: :return:

build_payload(***kwargs*)

Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

endpoint = '/api/v1/rooms.upload'

post_response(*result*)

Module contents

rocketchat.calls.users package

Submodules

rocketchat.calls.users.create_user module

class rocketchat.calls.users.create_user.**CreateUser**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.PostMixin*, *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_payload(***kwargs*)
Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

endpoint = '/api/v1/users.create'

post_response(*result*)

rocketchat.calls.users.delete_user module

class rocketchat.calls.users.delete_user.**DeleteUser**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.PostMixin*, *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)
Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_payload(***kwargs*)
Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

endpoint = '/api/v1/users.delete'

post_response(*result*)

rocketchat.calls.users.get_user_info module

class rocketchat.calls.users.get_user_info.**GetUserInfo**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)
Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/users.info'

post_response(*result*)

rocketchat.calls.users.get_users module

class rocketchat.calls.users.get_users.**GetUsers**(*settings=None*, **args*, ***kwargs*)

Bases: *rocketchat.calls.base.RocketChatBase*

build_endpoint(***kwargs*)
Build the endpoint for the user given some kwargs from the initial calling.

Returns

endpoint = '/api/v1/users.list'

post_response(*result*)

Module contents

Submodules

rocketchat.calls.base module

class rocketchat.calls.base.**PostMixin**

Bases: object

method = 'post'

class rocketchat.calls.base.**RocketChatBase**(*settings=None*, **args*, ***kwargs*)

Bases: object

auth_token = None

auth_user_id = None

build_endpoint(***kwargs*)

Build the endpoint for the user given some kwargs from the initial calling.

Returns

build_files(***kwargs*)

Build files :param kwargs: :return:

build_payload(***kwargs*)

Build a payload dict that will be passed directly to the endpoint. If you need to pass this as plain text or whatever you'll need to the dumping here.

Returns

call(**args*, ***kwargs*)

Parameters

- **args** –
- **kwargs** –

Returns

endpoint = None

files = None

headers = {}

method = 'get'

post_response(*result*)

set_auth_headers()

set_auth_token()

settings = None

Module contents

2.1.2 Submodules

2.1.3 rocketchat.api module

class `rocketchat.api.`**`RocketChatAPI`**(*settings=None*, **args*, ***kwargs*)

Bases: `object`

`close_im_room`(*room_id*, ***kwargs*)

Close direct message room

Parameters **`room_id`** –

Returns

`create_im_room`(*username*, ***kwargs*)

Create direct message room with user

Parameters **`username`** –

Returns

`create_public_room`(*name*, ***kwargs*)

Create room with given name :param name: Room name :param kwargs: members: The users to add to the channel when it is created.

Optional; Ex.: ["rocket.cat"], Default: []

read_only: Set if the channel is read only or not. Optional; Ex.: True, Default: False

Returns

`create_user`(*email*, *name*, *password*, *username*, ***kwargs*)

Create user :param email: E-mail :param name: Full name :param password: Password :param username: Username :param kwargs: active: roles: join_default_channels: require_password_change: send_welcome_email: verified: custom_fields: :return:

`delete_public_room`(*room_id*, ***kwargs*)

Delete room with given ID :param room_id: Room ID :param kwargs: :return:

`delete_user`(*user_id*, ***kwargs*)

Delete user :param user_id: User ID :param kwargs: :return:

`get_im_room_history`(*room_id*, *oldest=None*, *latest=datetime.datetime(2023, 3, 24, 18, 58, 37, 495925)*, *inclusive=False*, *count=20*, *unreads=False*, ***kwargs*)

Get various history of specific direct message room

Parameters

- **`room_id`** –
- **`kwargs`** –

Returns

`get_im_rooms`(***kwargs*)

Get direct message rooms

Returns

`get_my_info`(***kwargs*)

get_private_room_history(*room_id*, *oldest=None*, ***kwargs*)
Get various history of specific private group in this case private

Parameters

- **room_id** –
- **kwargs** –

Returns

get_private_room_info(*room_id*, ***kwargs*)
Get various information about a specific private group

Parameters

- **room_id** –
- **kwargs** –

Returns

get_private_rooms(***kwargs*)
Get a listing of all private rooms with their names and IDs

get_public_rooms(***kwargs*)
Get a listing of all public rooms with their names and IDs

get_room_history(*room_id*, *oldest=None*, *latest=datetime.datetime(2023, 3, 24, 18, 58, 37, 495897)*, *inclusive=False*, *count=20*, *unreads=False*, ***kwargs*)
Get various history of specific channel/room

Parameters

- **room_id** –
- **kwargs** –

Returns

get_room_id(*room_name*, ***kwargs*)
Get room ID :param room_name: :param kwargs: :return:

get_room_info(*room_id*, ***kwargs*)
Get various information about a specific channel/room

Parameters

- **room_id** –
- **kwargs** –

Returns

get_user_info(*user_id*, ***kwargs*)
Retrieves information about a user, the result is only limited to what the callee has access to view. :param user_id: :param kwargs: :return:

get_users(***kwargs*)
Gets all of the users in the system and their information :param kwargs: :return:

logout(***kwargs*)
Logout

Returns

open_im_room(*room_id*, ***kwargs*)
Open direct message room

Parameters `room_id` –

Returns

`send_message`(*message*, *room_id*, ***kwargs*)
Send a message to a given room

`set_room_topic`(*room_id*, *topic*, ***kwargs*)

`settings = None`

`upload_file`(*room_id*, *description*, *file*, *message*, *mime_type='text/plain'*, ***kwargs*)
Upload file to room :param room_id: :param description: :param file: :param kwargs: :return:

2.1.4 Module contents

CHAPTER 3

Detailed Module Docs

3.1 Licensing

MIT License

Copyright (c) 2016 - 2018 Derek Stegelman
Copyright (c) 2018 Peter Gonda

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Python Module Index

r

rocketchat, 14
rocketchat.api, 12
rocketchat.calls, 12
rocketchat.calls.auth, 5
rocketchat.calls.auth.get_me, 5
rocketchat.calls.base, 11
rocketchat.calls.channels, 7
rocketchat.calls.channels.create_public_room, 6
rocketchat.calls.channels.delete_public_room, 6
rocketchat.calls.channels.get_history, 6
rocketchat.calls.channels.get_public_rooms, 7
rocketchat.calls.channels.get_room_info, 7
rocketchat.calls.chat, 8
rocketchat.calls.chat.send_message, 7
rocketchat.calls.groups, 9
rocketchat.calls.groups.get_private_room_history, 8
rocketchat.calls.groups.get_private_room_info, 8
rocketchat.calls.groups.get_private_rooms, 8
rocketchat.calls.groups.get_room_id, 9
rocketchat.calls.groups.upload_file, 9
rocketchat.calls.users, 11
rocketchat.calls.users.create_user, 9
rocketchat.calls.users.delete_user, 10
rocketchat.calls.users.get_user_info, 10
rocketchat.calls.users.get_users, 10

Index

A

auth_token (*rocketchat.calls.base.RocketChatBase attribute*), 11
auth_user_id (*rocketchat.calls.base.RocketChatBase attribute*), 11

B

build_endpoint() (*rocketchat.calls.auth.get_me.GetMe method*), 5
build_endpoint() (*rocketchat.calls.base.RocketChatBase method*), 11
build_endpoint() (*rocketchat.calls.channels.create_public_room.CreatePublicRoom method*), 6
build_endpoint() (*rocketchat.calls.channels.delete_public_room.DeletePublicRoom method*), 6
build_endpoint() (*rocketchat.calls.channels.get_history.GetRoomHistory method*), 6
build_endpoint() (*rocketchat.calls.channels.get_public_rooms.GetPublicRooms method*), 7
build_endpoint() (*rocketchat.calls.channels.get_room_info.GetRoomInfo method*), 7
build_endpoint() (*rocketchat.calls.chat.send_message.SendMessage method*), 7
build_endpoint() (*rocketchat.calls.groups.get_private_room_history.GetPrivateRoomHistory method*), 8
build_endpoint() (*rocketchat.calls.groups.get_private_room_info.GetPrivateRoomInfo method*), 8
build_endpoint() (*rocketchat.calls.groups.get_private_rooms.GetPrivateRooms method*), 8
build_endpoint() (*rocketchat.calls.groups.get_room_id.GetRoomId method*), 9
build_endpoint() (*rocketchat.calls.groups.upload_file.UploadFile method*), 9
build_endpoint() (*rocketchat.calls.users.create_user.CreateUser method*), 9
build_endpoint() (*rocketchat.calls.users.delete_user.DeleteUser method*), 10
build_endpoint() (*rocketchat.calls.users.get_user_info.GetUserInfo method*), 10
build_endpoint() (*rocketchat.calls.users.get_users.GetUsers method*), 10
build_files() (*rocketchat.calls.base.RocketChatBase method*), 11
build_files() (*rocketchat.calls.groups.upload_file.UploadFile method*), 9
build_payload() (*rocketchat.calls.base.RocketChatBase method*), 11
build_payload() (*rocketchat.calls.channels.create_public_room.CreatePublicRoom method*), 6
build_payload() (*rocketchat.calls.channels.delete_public_room.DeletePublicRoom method*), 6
build_payload() (*rocketchat.calls.chat.send_message.SendMessage method*), 7
build_payload() (*rocketchat.calls.groups.upload_file.UploadFile method*), 9

build_payload() (*rocketchat.calls.users.create_user.CreateUser method*), 9
build_payload() (*rocketchat.calls.users.delete_user.DeleteUser method*), 10

C

call() (*rocketchat.calls.base.RocketChatBase method*), 11
close_im_room() (*rocketchat.api.RocketChatAPI method*), 12
create_im_room() (*rocketchat.api.RocketChatAPI method*), 12
create_public_room() (*rocketchat.api.RocketChatAPI method*), 12
create_user() (*rocketchat.api.RocketChatAPI method*), 12
CreatePublicRoom (*class in rocketchat.calls.channels.create_public_room*), 6
CreateUser (*class in rocketchat.calls.users.create_user*), 9

D

delete_public_room() (*rocketchat.api.RocketChatAPI method*), 12
delete_user() (*rocketchat.api.RocketChatAPI method*), 12
DeletePublicRoom (*class in rocketchat.calls.channels.delete_public_room*), 6
DeleteUser (*class in rocketchat.calls.users.delete_user*), 10

E

endpoint (*rocketchat.calls.auth.get_me.GetMe attribute*), 5
endpoint (*rocketchat.calls.base.RocketChatBase attribute*), 11
endpoint (*rocketchat.calls.channels.create_public_room.CreatePublicRoom attribute*), 6
endpoint (*rocketchat.calls.channels.delete_public_room.DeletePublicRoom attribute*), 6
endpoint (*rocketchat.calls.channels.get_history.GetRoomHistory attribute*), 6
endpoint (*rocketchat.calls.channels.get_public_rooms.GetPublicRooms attribute*), 7
endpoint (*rocketchat.calls.channels.get_room_info.GetRoomInfo attribute*), 7
endpoint (*rocketchat.calls.chat.send_message.SendMessage attribute*), 7
endpoint (*rocketchat.calls.groups.get_private_room_history.GetPrivateRoomHistory attribute*), 8
endpoint (*rocketchat.calls.groups.get_private_room_info.GetPrivateRoo attribute*), 8
endpoint (*rocketchat.calls.groups.get_private_rooms.GetPrivateRooms attribute*), 8
endpoint (*rocketchat.calls.groups.get_room_id.GetRoomId attribute*), 9
endpoint (*rocketchat.calls.groups.upload_file.UploadFile attribute*), 9
endpoint (*rocketchat.calls.users.create_user.CreateUser attribute*), 10
endpoint (*rocketchat.calls.users.delete_user.DeleteUser attribute*), 10
endpoint (*rocketchat.calls.users.get_user_info.GetUserInfo attribute*), 10
endpoint (*rocketchat.calls.users.get_users.GetUsers attribute*), 10

F

files (*rocketchat.calls.base.RocketChatBase attribute*), 11

G

get_im_room_history() (*rocketchat.api.RocketChatAPI method*), 12
get_im_rooms() (*rocketchat.api.RocketChatAPI method*), 12
get_my_info() (*rocketchat.api.RocketChatAPI method*), 12
get_private_room_history() (*rocketchat.api.RocketChatAPI method*), 12
get_private_room_info() (*rocketchat.api.RocketChatAPI method*), 13
get_private_rooms() (*rocketchat.api.RocketChatAPI method*), 13
get_public_rooms() (*rocketchat.api.RocketChatAPI method*), 13
get_room_history() (*rocketchat.api.RocketChatAPI method*), 13
get_room_id() (*rocketchat.api.RocketChatAPI method*), 13
get_room_info() (*rocketchat.api.RocketChatAPI method*), 13
get_public_info() (*rocketchat.api.RocketChatAPI method*), 13
get_users() (*rocketchat.api.RocketChatAPI method*), 13
GetMe (*class in rocketchat.calls.auth.get_me*), 5
GetPrivateRoomHistory (*class in rocketchat.calls.groups.get_private_room_history*), 8
GetPrivateRoomInfo (*class in rocketchat.calls.groups.get_private_room_info*), 8

GetPrivateRooms (*class in rocketchat.calls.groups.get_private_rooms*), 8
GetPublicRooms (*class in rocketchat.calls.channels.get_public_rooms*), 7
GetRoomHistory (*class in rocketchat.calls.channels.get_history*), 6
GetRoomId (*class in rocketchat.calls.groups.get_room_id*), 9
GetRoomInfo (*class in rocketchat.calls.channels.get_room_info*), 7
GetUserInfo (*class in rocketchat.calls.users.get_user_info*), 10
GetUsers (*class in rocketchat.calls.users.get_users*), 10

H

headers (*rocketchat.calls.base.RocketChatBase attribute*), 11

L

logout() (*rocketchat.api.RocketChatAPI method*), 13

M

method (*rocketchat.calls.base.PostMixin attribute*), 11
method (*rocketchat.calls.base.RocketChatBase attribute*), 11

O

open_im_room() (*rocketchat.api.RocketChatAPI method*), 13

P

post_response() (*rocketchat.calls.auth.get_me.GetMe method*), 5
post_response() (*rocketchat.calls.base.RocketChatBase method*), 11
post_response() (*rocketchat.calls.channels.create_public_room.CreatePublicRoom method*), 6
post_response() (*rocketchat.calls.channels.delete_public_room.DeletePublicRoom method*), 6
post_response() (*rocketchat.calls.channels.get_history.GetRoomHistory method*), 6
post_response() (*rocketchat.calls.channels.get_public_rooms.GetPublicRooms method*), 7
post_response() (*rocketchat.calls.channels.get_room_info.GetRoomInfo method*), 7
post_response() (*rocketchat.calls.groups.get_private_room_history.GetPrivateRoomHi method*), 8
post_response() (*rocketchat.calls.groups.get_private_room_info.GetPrivateRoomInfo method*), 8
post_response() (*rocketchat.calls.groups.get_private_rooms.GetPrivateRooms method*), 8
post_response() (*rocketchat.calls.groups.get_room_id.GetRoomId method*), 9
post_response() (*rocketchat.calls.groups.upload_file.UploadFile method*), 9
post_response() (*rocketchat.calls.users.create_user.CreateUser method*), 10
post_response() (*rocketchat.calls.users.delete_user.DeleteUser method*), 10
post_response() (*rocketchat.calls.users.get_user_info.GetUserInfo method*), 10
post_response() (*rocketchat.calls.users.get_users.GetUsers method*), 10
PostMixin (*class in rocketchat.calls.base*), 11

R

rocketchat (*module*), 14
rocketchat.api (*module*), 12
rocketchat.calls (*module*), 12
rocketchat.calls.auth (*module*), 5
rocketchat.calls.auth.get_me (*module*), 5
rocketchat.calls.base (*module*), 11
rocketchat.calls.channels (*module*), 7
rocketchat.calls.channels.create_public_room (*module*), 6
rocketchat.calls.channels.delete_public_room (*module*), 6
rocketchat.calls.channels.get_history (*module*), 6
rocketchat.calls.channels.get_public_rooms (*module*), 7
rocketchat.calls.channels.get_room_info (*module*), 7
rocketchat.calls.chat (*module*), 8
rocketchat.calls.chat.send_message (*module*), 7
rocketchat.calls.groups (*module*), 9
rocketchat.calls.groups.get_private_room_history (*module*), 8

rocketchat.calls.groups.get_private_room_info (*module*), 8
rocketchat.calls.groups.get_private_rooms (*module*), 8
rocketchat.calls.groups.get_room_id (*module*), 9
rocketchat.calls.groups.upload_file (*module*), 9
rocketchat.calls.users (*module*), 11
rocketchat.calls.users.create_user (*module*), 9
rocketchat.calls.users.delete_user (*module*), 10
rocketchat.calls.users.get_user_info (*module*), 10
rocketchat.calls.users.get_users (*module*), 10
RocketChatAPI (*class in rocketchat.api*), 12
RocketChatBase (*class in rocketchat.calls.base*), 11

S

send_message() (*rocketchat.api.RocketChatAPI method*), 14
SendMessage (*class in rocketchat.calls.chat.send_message*), 7
set_auth_headers() (*rocketchat.calls.base.RocketChatBase method*), 11
set_auth_token() (*rocketchat.calls.base.RocketChatBase method*), 11
set_room_topic() (*rocketchat.api.RocketChatAPI method*), 14
settings (*rocketchat.api.RocketChatAPI attribute*), 14
settings (*rocketchat.calls.base.RocketChatBase attribute*), 11

U

upload_file() (*rocketchat.api.RocketChatAPI method*), 14
UploadFile (*class in rocketchat.calls.groups.upload_file*), 9