slackly Documentation

Release 1.0.6

Hunter Senft-Grupp

May 25, 2017

Contents:

CHAPTER 1

Slackly - A Slack Toolkit

Use Cases:

- Use the *slackly.SlackClient* to make calls to the Web API
- Use the *slackly.SlackRTMClient* to listen to the Real Time Messaging API (via a websocket connection)
- Use the *slackly.events* and *slackly.schema.types* in this library to build higher level abstractions on the Slack API

> **Danger:** This library is very much in alpha and under very active development to flesh out the API. I'll do my best to not make breaking changes, and hide most of these changes behind feature toggles that can become default in future versions. I want users and feedback, so I will do my utmost to make usage of this library as safe as can be.

Installing

```
$ pip install slackly
```

Simple Examples

Creating a Client and Making an API Call

```
from slackly import SlackClient
import os

client = SlackClient(token=os.environ['SLACK_TOKEN'])

```

```
# Call the api.test endpoint
client.api_call('api.test')
```

Creating the API object and preparing a call

```
from slackly import SlackAPI, SlackClient
import os
slack = SlackAPI()

# Prepare an api call. Pass "api_call" a client to get actually make the call
api_call = slack.api.test()

client = SlackClient(token=os.environ['SLACK_TOKEN'])
result = api_call(client)
```

Printing Events off of the Real Time Messaging API

```
import os
from slackly import SlackRTMClient

rtm_client = SlackRTMClient.from_token(token=os.environ['SLACK_TOKEN'])
for event in rtm_client.get_events_forever():
    print(event)
```

Using Event Types

```
import os
from slackly import SlackRTMClient, SlackEventParsed
from slackly.events import Message, UserTyping
rtm_client = SlackRTMClient.from_token(token=os.environ['SLACK_TOKEN'])
rtm_client.event_factory = SlackEventParsed  # Tell the RTM client to make␣
↪events using this class

for event in rtm_client.get_events_forever():
    if isinstance(event, Message):
        print("We're dealing with a message")

    elif isinstance(event, UserTyping):
        print("Someone's typing")

    else:
        print("I don't care about this event")
```

Development

```
$ git clone https://github.com/huntcsg/slackly.git
$ cd slackly
$ tox    # Runs test suites against all python versions, pypy, does style␣
↪and test coverage analysis
```

1. All pull requests must pass the travis-ci builds
2. All pull requests should include inline (docstring) documentation, updates to built documentation if applicable, and test coverage. This project aspires to be a 100% test coverage library.
3. If integration or regression test coverage is needed, let the project maintainer know and we can work out the best way to do so.

Changes

1.0.4

- Documentation Updates and Release stability

1.0.0

- Initial Release

CHAPTER 2

Quickstart

1. Follow these instructions to get a token from slack
2. Set the environment variable *SLACK_TOKEN* as the token you got in Step 1
3. Install slackly

   ```
   $ pip install slackly
   ```

4. List your team's channels and users:

   ```
   from slackly import SlackClient
   import os
   client = SlackClient(token=os.environ['SLACK_TOKEN'])

   channels = client.api.channels.list()
   users = client.api.users.list()

   print(channels)
   print(users)
   ```

CHAPTER 3

Guides

Setting Up A Token

Making a Bot

CHAPTER 4

Contributing

How to add a New Event

1. Add a new module in the src/slackly/events/ directory. Follow the below template:

```
@register_event('accounts_changed')
class AccountsChanged(BaseEvent):
    """
        .. code-block:: json
            :caption: Example json response

            {
                "type": "accounts_changed"
            }


    For more information see https://api.slack.com/events/accounts_changed
    """
    @property
    def schema(self):
        return {
            'type': types.String,
        }
```

2. Be sure of the following details:

   (a) The event should be "registered" under the string found in event['type'] key

   (b) The class subclasses from `slackly.events.BaseEvent`

   (c) The class includes a rst json code-block with an example event json blob

   (d) Although it is not 100% necessary, please include a *schema* property as shown

3. Add an import in src/slackly/events/__init__.py

```
from .accounts_changed import AccountsChanged
```

4. Add an example json file in tests/data/events/[YOUR EVENT]/example.json, where [YOUR EVENT] is the string you registered
5. Run the test suite

How to add a new Web Method

For code organization and a usable API, two classes make up each endpoint.

There is the `BaseAPIDispatch` class and the `BaseAPIEndpoint`.

For example, if you had a single slack endpoint called foo.bar, that took a required argument baz, and an optional argument foobar. If it requires a token, then indicated it. Here's how it would be written:

```
class Foo(BaseAPIDispatch):
    pass

@Foo.register('bar')
class Bar(BaseAPIEndpoint):
    required_args = {
        'baz',
    }
    optional_args = {
         'foobar',
    }
    options = {
        'requires_token': True
    }s

    def __call__(self,baz, foobar=None):
        optional_args = {}
        if foobar is not None:
            optional_args['foobar'] = foobar

        return BaseAPIEndpoint.__call__(self, baz=baz, **optional_args)

class SlackAPI(object):
    foo = Foo()
```

Adding a new endpoint

1. In the module named like the root of the endpoint (the 'foo.py' in the above example), create a class like so:

```
@Foo.register('bar')
class FooBar(BaseAPIEndpoint):
    ...
```

2. Although I did not go into detail above, please include the description of the endpoint from the docs as well as at least one json code block

3. Add a new directory under tests/data/endpoints (e.g. tests/data/endpoints/foo.bar) with simple.json file with an example json object representative of the response the endpoint produces.

Adding a new dispatch route

1. Add a new module in the *src/slackly/api/endpoints* directory. Within this module create a class:

   ```
   class Foo(BaseAPIEndpoint):
       pass
   ```

2. In the src/api/slackly/__init__.py add the new dispatch object as a class attribute

   ```
   class SlackAPI(object):
       foo = Foo()
   ```

3. Follow the directions above for registering endpoints.

New Type

CHAPTER 5

API Docs

Main Classes

class slackly.**SlackAPI**(*bind=None*)

The API object

Parameters **bind** – Optional. The client to bind to this API object.

class slackly.**SlackClient**(*token=None*, *base_url='https://slack.com/api'*, *user_agent=None*, *response_factory=<class 'slackly.client.response_factory.SlackAPIDictResponse'>*, *include_api=True*, *raise_for_status=True*)

The web client

Parameters

- **token** – Required. A str slack token
- **base_url** – Optional. A str base url for slack
- **user_agent** – Optional. A user agent string. leave this blank unless you know what you are doing
- **response_factory** – A response factory (gets __call__'ed on every response)
- **raise_for_status** – Indicates whether a non 200 http status code cause a python exception

class slackly.**SlackRTMClient**(*token*, *url*, *event_factory=<function SlackEventDict>*, *client=None*, *keep_alive=True*, *ignore_pong=True*, *donotdie=True*)

The RTM Client

Parameters

- **token** – A str token
- **url** – A str the base url for slack
- **event_factory** – A class to process events coming off the real time messaging api

- **`client`** – A *`slackly.SlackClient`* or None.
- **`keep_alive`** – The option to have the websocket automatically written to at least every 3 seconds
- **`ignore_pong`** – Whether to emit or squash the response to pings

Slack API Endpoints

Base Classes and Utilities

Endpoints

class `slackly.api.endpoints.api.`**`ApiTest`**(*bind=None*)

This method helps you test your calling code.

The response includes any supplied arguments:

```
{
    "ok": true,
    "args": {
        "foo": "bar"
    }
}
```

If called with an error argument an error response is returned:

```
{
    "ok": false,
    "error": "my_error",
    "args": {
        "error": "my_error"
    }
}
```

For more information see https://api.slack.com/methods/test

`__call__`(*error=None*, *foo=None*)

Checks API calling code.

Parameters

- **`error`** – Optional. Error response to return e.g. my_error
- **`foo`** – Optional. example property to return e.g. bar

class `slackly.api.endpoints.auth.`**`AuthRevoke`**(*bind=None*)

This method revokes an access token. Use it when you no longer need a token. For example, with a Sign In With Slack app, call this to log a user out.

The response indicates whether the token was actually revoked:

```
{
    "ok": true,
    "revoked": true
}
```

For more information see https://api.slack.com/methods/revoke

__call__(*test=None*)
Revokes a token.

Parameters **test** – Optional. Setting this parameter to 1 triggers a testing mode where the specified token will not actually be revoked. e.g. true

class slackly.api.endpoints.auth.**AuthTest**(*bind=None*)
This method checks authentication and tells you who you are.

```
{
    "ok": true,
    "url": "https:\/\/myteam.slack.com\/",
    "team": "My Team",
    "user": "cal",
    "team_id": "T12345",
    "user_id": "U12345"
}
```

When working against a team within an Enterprise Grid, you'll also find their enterprise_id here.

For more information see https://api.slack.com/methods/test

__call__()
Checks authentication & identity.

class slackly.api.endpoints.bots.**BotsInfo**(*bind=None*)
This method returns information about a bot user.

bot_id is returned from bot_message message events and in the response of methods like channels.history.

Use this method to look up the username and icons for those bot users. Use the app_id field to identify whether a bot belongs to your Slack app.

Returns a bot user:

```
{
    "ok": true,
    "bot": {
        "id": "B12345678",
        "app_id": "A4H1JB4AZ",
        "deleted": false,
        "name": "My Bot",
        "icons": {
            "image_36": "https://...",
            "image_48": "https://...",
            "image_72": "https://..."
        }
    }
}
```

For more information see https://api.slack.com/methods/info

__call__(*bot=None*)
Gets information about a bot user.

Parameters **bot** – Optional. Bot user to get info on e.g. B12345678

class slackly.api.endpoints.channels.**ChannelsArchive**(*bind=None*)
This method archives a channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/archive

__call__(*channel*)

Archives a channel.

Parameters **channel** – Required. Channel to archive e.g. C1234567890

class slackly.api.endpoints.channels.**ChannelsCreate**(*bind=None*)

This method is used to create a channel.

To create a private channel, use groups.create.

If successful, the command returns a channel object, including state information:

```
{
    "ok": true,
    "channel": {
        "id": "C024BE91L",
        "name": "fun",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "is_member": true,
        "is_general": false,
        "last_read": "0000000000.000000",
        "latest": null,
        "unread_count": 0,
        "unread_count_display": 0,
        "members": [ ... ],
        "topic": { ... },
        "purpose": { ... }
    }
}
```

For more information see https://api.slack.com/methods/create

__call__(*name*, *validate=None*)

Creates a channel.

Parameters

- **name** – Required. Name of channel to create e.g. mychannel
- **validate** – Optional. Whether to return errors on invalid channel name instead of modifying it to meet the specified criteria. e.g. true

class slackly.api.endpoints.channels.**ChannelsHistory**(*bind=None*)

This method returns a portion of message events from the specified public channel.

To read the entire history for a channel, call the method with no latest or oldest arguments, and then continue paging using the instructions below.

To retrieve a single message, specify its ts value as latest, set inclusive to true, and dial your count down to 1.

```
{
    "ok": true,
    "latest": "1358547726.000003",
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
            "reactions": [
                {
                    "name": "space_invader",
                    "count": 3,
                    "users": [ "U1", "U2", "U3" ]
                },
                {
                    "name": "sweet_potato",
                    "count": 5,
                    "users": [ "U1", "U2", "U3", "U4", "U5" ]
                }
            ]
                    },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "has_more": false
}
```

Navigating through collections of messages The messages array contains up to 100 messages between latest and oldest. If there were more than 100 messages between those two points, then has_more will be true. If a message has the same timestamp as latest or oldest it will not be included in the list, unless inclusive is true. This allows a client to fetch all messages in a hole in channel history, by calling channels.history with latest set to the oldest message they have after the hole, and oldest to the latest message they have before the hole. If the response includes has_more then the client can make another call, using the ts value of the final messages as the latest param to get the next page of messages. If there are more than 100 messages between the two timestamps then the messages returned are the ones closest to latest. In most cases an application will want the most recent messages and will page backward from there. If oldest is provided but not latest then the messages returned are those closest to oldest, allowing you to page forward through history if desired.

Retrieving a single message channels.history can also be used to pluck a single message from the archive. You'll need a message's ts value, uniquely identifying it within a channel. You'll also need that channel's ID. Provide another message's ts value as the latest parameter. Specify true for the inclusive parameter and set the count to 1. If it exists, you'll receive the queried message in return. GET /api/channels.history?token=TOKEN_WITH_CHANNELS_HISTORY_SCOPE&channel=C2EB2QT8A&latest=1476909142.00

Message types Messages of type "message" are user-entered text messages sent to the channel, while other types are events that happened within the channel. All messages have both a type and a sortable ts, but the other fields depend on the type. For a list of all possible events, see the channel messages documentation. Messages that have been reacted to by team members will have a reactions array delightfully included. If a message has been starred by the calling user, the is_starred property will be present and true. This property is only added for starred items, so is not present in the majority of messages. The is_limited boolean property is only included for free teams that have reached the free message limit. If true, there are messages before the current result set, but they are beyond the message limit.

For more information see https://api.slack.com/methods/history

`__call__`(*channel*, *count=None*, *inclusive=None*, *latest=None*, *oldest=None*, *unreads=None*)

Fetches history of messages and events from a channel.

Parameters

- **`channel`** – Required. Channel to fetch history for. e.g. C1234567890
- **`count`** – Optional, default=100. Number of messages to return, between 1 and 1000. e.g. 100
- **`inclusive`** – Optional, default=0. Include messages with latest or oldest timestamp in results. e.g. true
- **`latest`** – Optional, default=now. End of time range of messages to include in results. e.g. 1234567890.123456
- **`oldest`** – Optional, default=0. Start of time range of messages to include in results. e.g. 1234567890.123456
- **`unreads`** – Optional, default=0. Include unread_count_display in the output? e.g. true

class `slackly.api.endpoints.channels.`**`ChannelsInfo`**(*bind=None*)

This method returns information about a team channel.

To retrieve information on a private channel, use groups.info.

Returns a channel object:

```
{
    "ok": true,
    "channel": {
        "id": "C024BE91L",
        "name": "fun",

        "created": 1360782804,
        "creator": "U024BE7LH",

        "is_archived": false,
        "is_general": false,
        "is_member": true,
        "is_starred": true,

        "members": [ ... ],

        "topic": { ... },
        "purpose": { ... },

        "last_read": "1401383885.000061",
        "latest": { ... },
        "unread_count": 0,
        "unread_count_display": 0
    }
}
```

An is_org_shared attribute may appear set to true when the channel is part of a shared channel between multiple teams of an Enterprise Grid. You'll also find the team's enterprise_id. See the Enterprise Grid shared channels documentation for more detail.

For more information see https://api.slack.com/methods/info

`__call__`(*channel*)

Gets information about a channel.

Parameters **`channel`** – Required. Channel to get info on e.g. C1234567890

class `slackly.api.endpoints.channels.`**`ChannelsInvite`**(*bind=None*)

This method is used to invite a user to a channel. The calling user must be a member of the channel.

```
{
    "ok": true,
    "channel": {
        "id": "C024BE91L",
        "name": "fun",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "is_member": true,
        "is_general": false,
        "last_read": "1401383885.000061",
        "latest": { ... },
        "unread_count": 0,
        "unread_count_display": 0,
        "members": [ ... ],
        "topic": {
            "value": "Fun times",
            "creator": "U024BE7LV",
            "last_set": 1369677212
        },
        "purpose": {
            "value": "This channel is for fun",
            "creator": "U024BE7LH",
            "last_set": 1360782804
        }
    }
}
```

For more information see https://api.slack.com/methods/invite

`__call__`(*channel, user*)

Invites a user to a channel.

Parameters

- **`channel`** – Required. Channel to invite user to. e.g. C1234567890
- **`user`** – Required. User to invite to channel. e.g. U1234567890

class `slackly.api.endpoints.channels.`**`ChannelsJoin`**(*bind=None*)

This method is used to join a channel. If the channel does not exist, it is created.

If successful, the command returns a channel object, including state information:

```
{
    "ok": true,
    "channel": {
        "id": "C024BE91L",
        "name": "fun",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "is_member": true,
        "is_general": false,
        "last_read": "1401383885.000061",
        "latest": { ... },
        "unread_count": 0,
        "unread_count_display": 0,
        "members": [ ... ],
        "topic": {
            "value": "Fun times",
            "creator": "U024BE7LV",
            "last_set": 1369677212
        },
        "purpose": {
            "value": "This channel is for fun",
            "creator": "U024BE7LH",
            "last_set": 1360782804
        }
    }
}
```

If you are already in the channel, the response is slightly different. already_in_channel will be true, and a limited channel object will be returned. This allows a client to see that the request to join GeNERaL is the same as the channel #general that the user is already in:

```
{
    "ok": true,
    "already_in_channel": true,
    "channel": {
        "id": "C024BE91L",
        "name": "fun",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "is_general": false
    }
}
```

For more information see https://api.slack.com/methods/join

`__call__`(*name*, *validate=None*)

Joins a channel, creating it if needed.

Parameters

- **`name`** – Required. Name of channel to join e.g. C1234567890
- **`validate`** – Optional. Whether to return errors on invalid channel name instead of modifying it to meet the specified criteria. e.g. true

class `slackly.api.endpoints.channels.`**`ChannelsKick`**(*bind=None*)

This method allows a user to remove another member from a team channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/kick

`__call__`(*channel*, *user*)
Removes a user from a channel.

Parameters

- **`channel`** – Required. Channel to remove user from. e.g. C1234567890
- **`user`** – Required. User to remove from channel. e.g. U1234567890

class `slackly.api.endpoints.channels.`**`ChannelsLeave`**(*bind=None*)
This method is used to leave a channel.

```
{
    "ok": true
}
```

This method will not return an error if the user was not in the channel before it was called. Instead the response will include a not_in_channel property:

```
{
    "ok": true,
    "not_in_channel": true
}
```

For more information see https://api.slack.com/methods/leave

`__call__`(*channel*)
Leaves a channel.

Parameters **`channel`** – Required. Channel to leave e.g. C1234567890

class `slackly.api.endpoints.channels.`**`ChannelsList`**(*bind=None*)
This method returns a list of all channels in the team. This includes channels the caller is in, channels they are not currently in, and archived channels but does not include private channels. The number of (non-deactivated) members in each channel is also returned.

To retrieve a list of private channels, use groups.list.

Having trouble getting a HTTP 200 response from this method? Try excluding the members list from each channel object using the exclude_members parameter.

Returns a list of limited channel objects:

```
{
    "ok": true,
    "channels": [
        {
            "id": "C024BE91L",
            "name": "fun",
            "created": 1360782804,
            "creator": "U024BE7LH",
            "is_archived": false,
            "is_member": false,
            "num_members": 6,
            "topic": {
                "value": "Fun times",
                "creator": "U024BE7LV",
                "last_set": 1369677212
            },
            "purpose": {
                "value": "This channel is for fun",
                "creator": "U024BE7LH",
                "last_set": 1360782804
            }
        },
        ....
    ]
}
```

To get a full channel object, call the channels.info method. Use the exclude_members parameter to exclude the members collection from each listed channel. This improves performance, especially with larger teams. Use channels.info to retrieve members on a channel-by-channel basis instead. An is_org_shared attribute may appear set to true on channels that are shared channel between multiple teams of an enterprise grid. See the enterprise grid shared channels documentation for more detail.

For more information see https://api.slack.com/methods/list

`__call__`(*exclude_archived=None*, *exclude_members=None*)

Lists all channels in a Slack team.

Parameters

- **`exclude_archived`** – Optional, default=false. Exclude archived channels from the list e.g. true
- **`exclude_members`** – Optional, default=false. Exclude the members collection from each channel e.g. true

class `slackly.api.endpoints.channels.`**`ChannelsMark`**(*bind=None*)

This method moves the read cursor in a channel.

```
{
    "ok": true
}
```

After making this call, the mark is saved to the database and broadcast via the message server to all open connections for the calling user. Clients should try to avoid making this call too often. When needing to mark a read position, a client should set a timer before making the call. In this way, any further updates needed during the timeout will not generate extra calls (just one per channel). This is useful for when reading scroll-back history, or following a busy live channel. A timeout of 5 seconds is a good starting point. Be sure to flush these calls on shutdown/logout.

For more information see https://api.slack.com/methods/mark

`__call__`(*channel*, *ts*)

Sets the read cursor in a channel.

Parameters

- **`channel`** – Required. Channel to set reading cursor in. e.g. C1234567890
- **`ts`** – Required. Timestamp of the most recently seen message. e.g. 1234567890.123456

class `slackly.api.endpoints.channels.`**`ChannelsRename`**(*bind=None*)

This method renames a team channel.

The only people who can rename a channel are Team Admins, or the person that originally created the channel. Others will receive a “not_authorized” error.

```
{
    "ok": true,
    "channel": {
        "id": "C024BE91L",
        "is_channel": true,
        "name": "new_name",
        "created": 1360782804
    }
}
```

Returns the channel ID, name and date created (as a unix timestamp).

For more information see https://api.slack.com/methods/rename

__call__(*channel*, *name*, *validate=None*)
Renames a channel.

Parameters

- **channel** – Required. Channel to rename e.g. C1234567890
- **name** – Required. New name for channel. e.g.
- **validate** – Optional. Whether to return errors on invalid channel name instead of modifying it to meet the specified criteria. e.g. true

class slackly.api.endpoints.channels.**ChannelsReplies**(*bind=None*)
This method returns an entire thread (a message plus all the messages in reply to it).

```
{
    "ok": true,
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "thread_info": [
        "complete": true,
        "count": 3,
    ]
}
```

For more information see https://api.slack.com/methods/replies

__call__(*channel*, *thread_ts*)
Retrieve a thread of messages posted to a channel

Parameters

- **channel** – Required. Channel to fetch thread from e.g. C1234567890
- **thread_ts** – Required. Unique identifier of a thread's parent message e.g. 1234567890.123456

class slackly.api.endpoints.channels.**ChannelsSetPurpose**(*bind=None*)

This method is used to change the purpose of a channel. The calling user must be a member of the channel.

```
{
    "ok": true,
    "purpose": "This is the new purpose!"
}
```

For more information see https://api.slack.com/methods/setPurpose

__call__(*channel*, *purpose*)

Sets the purpose for a channel.

Parameters

- **channel** – Required. Channel to set the purpose of e.g. C1234567890
- **purpose** – Required. The new purpose e.g. My Purpose

class slackly.api.endpoints.channels.**ChannelsSetTopic**(*bind=None*)

This method is used to change the topic of a channel. The calling user must be a member of the channel.

```
{
    "ok": true,
    "topic": "This is the new topic!"
}
```

For more information see https://api.slack.com/methods/setTopic

__call__(*channel*, *topic*)

Sets the topic for a channel.

Parameters

- **channel** – Required. Channel to set the topic of e.g. C1234567890
- **topic** – Required. The new topic e.g. My Topic

class slackly.api.endpoints.channels.**ChannelsUnarchive**(*bind=None*)

This method unarchives a channel. The calling user is added to the channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/unarchive

__call__(*channel*)

Unarchives a channel.

Parameters **channel** – Required. Channel to unarchive e.g. C1234567890

class slackly.api.endpoints.chat.**ChatDelete**(*bind=None*)

This method deletes a message from a channel.

```
{
    "ok": true,
    "channel": "C024BE91L",
    "ts": "1401383885.000061"
}
```

The response includes the channel and timestamp properties of the deleted message.

For more information see https://api.slack.com/methods/delete

__call__(*channel*, *ts*, *as_user=None*)

Deletes a message.

Parameters

- **channel** – Required. Channel containing the message to be deleted. e.g. C1234567890
- **ts** – Required. Timestamp of the message to be deleted. e.g. 1405894322.002768
- **as_user** – Optional. Pass true to delete the message as the authed user. Bot users in this context are considered authed users. e.g. true

class slackly.api.endpoints.chat.**ChatMeMessage**(*bind=None*)

This method sends a me message to a channel from the calling user.

```
{
    "ok": true,
    "channel": "C024BE7LR",
    "ts": "1417671948.000006"
}
```

For more information see https://api.slack.com/methods/meMessage

__call__(*channel*, *text*)

Share a me message into a channel.

Parameters

- **channel** – Required. Channel to send message to. Can be a public channel, private group or IM channel. Can be an encoded ID, or a name. e.g. C1234567890
- **text** – Required. Text of the message to send. e.g. Hello world

class slackly.api.endpoints.chat.**ChatPostMessage**(*bind=None*)

This method posts a message to a public channel, private channel, or direct message/IM channel.

```
{
    "ok": true,
    "ts": "1405895017.000506",
    "channel": "C024BE91L",
    "message": {
        ...
    }
}
```

The response includes the timestamp (ts) and channel for the posted message. It also includes the complete message object, as it was parsed by our servers. This may differ from the provided arguments as our servers sanitize links, attachments and other properties.

For more information see https://api.slack.com/methods/postMessage

__call__(*channel*, *text*, *as_user=None*, *attachments=None*, *icon_emoji=None*, *icon_url=None*, *link_names=None*, *parse=None*, *reply_broadcast=None*, *thread_ts=None*, *unfurl_links=None*, *unfurl_media=None*, *username=None*)

Sends a message to a channel.

Parameters

- **`channel`** – Required. Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name. See below for more details. e.g. C1234567890
- **`text`** – Required. Text of the message to send. See below for an explanation of formatting. This field is usually required, unless you're providing only attachments instead. e.g. Hello world
- **`as_user`** – Optional. Pass true to post the message as the authed user, instead of as a bot. Defaults to false. See authorship below. e.g. true
- **`attachments`** – Optional. Structured message attachments. e.g. [{"pretext": "pre-hello", "text": "text-world"}]
- **`icon_emoji`** – Optional. Emoji to use as the icon for this message. Overrides icon_url. Must be used in conjunction with as_user set to false, otherwise ignored. See authorship below. e.g. :chart_with_upwards_trend:
- **`icon_url`** – Optional. URL to an image to use as the icon for this message. Must be used in conjunction with as_user set to false, otherwise ignored. See authorship below. e.g. http://lorempixel.com/48/48
- **`link_names`** – Optional. Find and link channel names and usernames. e.g. true
- **`parse`** – Optional. Change how messages are treated. Defaults to none. See below. e.g. full
- **`reply_broadcast`** – Optional. Used in conjunction with thread_ts and indicates whether reply should be made visible to everyone in the channel or conversation. Defaults to false. e.g. true
- **`thread_ts`** – Optional. Provide another message's ts value to make this message a reply. Avoid using a reply's ts value; use its parent instead. e.g. 1234567890.123456
- **`unfurl_links`** – Optional. Pass true to enable unfurling of primarily text-based content. e.g. true
- **`unfurl_media`** – Optional. Pass false to disable unfurling of media content. e.g. false
- **`username`** – Optional. Set your bot's user name. Must be used in conjunction with as_user set to false, otherwise ignored. See authorship below. e.g. My Bot

class `slackly.api.endpoints.chat.`**`ChatUnfurl`**(*bind=None*)

This method attaches Slack app unfurl behavior to a specified and relevant message. A user token is required as this method does not support bot user tokens.

The first time this method is executed with a particular ts and channel combination, the valid unfurls attachments you provide will be attached to the message. Subsequent attempts with the same ts and channel values will modify the same attachments, rather than adding more.

```
{
    "ok": true
}
```

As you can see, we provide a minimal positive response when your unfurl attempt is successful. When it is not, you'll receive one of the errors below and ok will not be false.

For more information see https://api.slack.com/methods/unfurl

`__call__`(*channel*, *ts*, *unfurls*, *user_auth_required=None*)

Unfurl a URL that a user posted

Parameters

- **`channel`** – Required. Channel ID of the message e.g. C1234567890
- **`ts`** – Required. Timestamp of the message to add unfurl behavior to e.g.
- **`unfurls`** – Required. JSON mapping a set of URLs from the message to their unfurl attachments e.g.
- **`user_auth_required`** – Optional, default=0. Set to true or 1 to indicate the user must install your Slack app to trigger unfurls for this domain e.g. true

class `slackly.api.endpoints.chat.`**`ChatUpdate`**`(`*bind=None*`)`

This method updates a message in a channel. Though related to chat.postMessage, some parameters of chat.update are handled differently.

```
{
        "ok": true,
        "channel": "C024BE91L",
        "ts": "1401383885.000061",
        "text": "Updated Text"
    }
```

The response includes the text, channel and timestamp properties of the updated message so clients can keep their local copies of the message in sync.

Bot users To use chat.update with a bot user token, you'll need to think of your bot user as a user, and pass as_user set to true while editing a message created by that same bot user.

Interactive messages with buttons If you're posting message with buttons, you may use chat.update to continue updating ongoing state changes around a message. Provide the ts field the message you're updating and follow the bot user instructions above to update message text, remove or add attachments and actions.

For more information see https://api.slack.com/methods/update

`__call__``(`*channel*, *text*, *ts*, *as_user=None*, *attachments=None*, *link_names=None*, *parse=None*`)`

Updates a message.

Parameters

- **`channel`** – Required. Channel containing the message to be updated. e.g. C1234567890
- **`text`** – Required. New text for the message, using the default formatting rules. e.g. Hello world
- **`ts`** – Required. Timestamp of the message to be updated. e.g. 1405894322.002768
- **`as_user`** – Optional. Pass true to update the message as the authed user. Bot users in this context are considered authed users. e.g. true
- **`attachments`** – Optional. Structured message attachments. e.g. [{"pretext": "pre-hello", "text": "text-world"}]
- **`link_names`** – Optional. Find and link channel names and usernames. Defaults to none. This parameter should be used in conjunction with parse. To set link_names to 1, specify a parse mode of full. e.g. true
- **`parse`** – Optional. Change how messages are treated. Defaults to client, unlike chat.postMessage. See below. e.g. none

class `slackly.api.endpoints.dnd.`**`DndEndDnd`**`(`*bind=None*`)`

Ends the user's currently scheduled Do Not Disturb session immediately.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/endDnd

`__call__`()
Ends the current user's Do Not Disturb session immediately.

class `slackly.api.endpoints.dnd.`**`DndEndSnooze`**(*bind=None*)
Ends the current user's snooze mode immediately.

```
{
    "ok": true,
    "dnd_enabled": true,
    "next_dnd_start_ts": 1450418400,
    "next_dnd_end_ts": 1450454400,
    "snooze_enabled": false
}
```

For more information see https://api.slack.com/methods/endSnooze

`__call__`()
Ends the current user's snooze mode immediately.

class `slackly.api.endpoints.dnd.`**`DndInfo`**(*bind=None*)
Provides information about a user's current Do Not Disturb settings.

```
{
    "ok": true,
    "dnd_enabled": true,
    "next_dnd_start_ts": 1450416600,
    "next_dnd_end_ts": 1450452600,
    "snooze_enabled": true,
    "snooze_endtime": 1450416600,
    "snooze_remaining": 1196
}
```

Snooze properties All of the snooze_* properties will only be visible if the user being queried is also the current user. The snooze_endtime and snooze_remaining properties will only be returned if snooze_enabled is true.

For more information see https://api.slack.com/methods/info

`__call__`(*user=None*)
Retrieves a user's current Do Not Disturb status.

Parameters **`user`** – Optional. User to fetch status for (defaults to current user) e.g. U1234

class `slackly.api.endpoints.dnd.`**`DndSetSnooze`**(*bind=None*)
Adjusts the snooze duration for a user's Do Not Disturb settings. If a snooze session is not already active for the user, invoking this method will begin one for the specified duration.

```
{
    "ok": true,
    "snooze_enabled": true,
    "snooze_endtime": 1450373897,
    "snooze_remaining": 60,
}
```

The snooze_remaining field is expressed in seconds. If your request presents a num_minutes value of 1, the response's snooze_remaining will be 60.

For more information see https://api.slack.com/methods/setSnooze

`__call__`(*num_minutes*)

Turns on Do Not Disturb mode for the current user, or changes its duration.

Parameters **`num_minutes`** – Required. Number of minutes, from now, to snooze until. e.g. 60

class `slackly.api.endpoints.dnd.`**`DndTeamInfo`**(*bind=None*)

Provides information about the current Do Not Disturb settings for users of a Slack team.

```
{
    "ok": true,
    "users": {
        "U023BECGF": {
            "dnd_enabled": true,
            "next_dnd_start_ts": 1450387800,
            "next_dnd_end_ts": 1450423800
        },
        "U058CJVAA": {
            "dnd_enabled": false,
            "next_dnd_start_ts": 1,
            "next_dnd_end_ts": 1
        }
    }
}
```

For more information see https://api.slack.com/methods/teamInfo

`__call__`(*users=None*)

Retrieves the Do Not Disturb status for users on a team.

Parameters **`users`** – Optional. Comma-separated list of users to fetch Do Not Disturb status for e.g. U1234,U4567

class `slackly.api.endpoints.emoji.`**`EmojiList`**(*bind=None*)

This method lists the custom emoji for a team.

```
{
    "ok": true,
    "emoji": {
        "bowtie": "https:\/\/my.slack.com\/emoji\/bowtie\/46ec6f2bb0.png",
        "squirrel": "https:\/\/my.slack.com\/emoji\/squirrel\/f35f40c0e0.png",
        "shipit": "alias:squirrel",
        ...
    }
}
```

The emoji property contains a map of name/url pairs, one for each custom emoji used by the team. The alias: pseudo-protocol will be used where the emoji is an alias, the string following the colon is the name of the other emoji this emoji is an alias to.

For more information see https://api.slack.com/methods/list

`__call__`()

Lists custom emoji for a team.

class `slackly.api.endpoints.files.`**`CommentsAdd`**`(`*bind=None*`)`

Add a comment to an existing file.

If successful, the response will include a file comment object.

```
{
    "ok": true,
    "comment": {
        "id": "Fc1234567890",
        "created": 1356032811,
        "timestamp": 1356032811,
        "user": "U1234567890",
        "comment": "Everyone should take a moment to read this file.",
        "channel": "C1234467890"
    }
}
```

For more information see https://api.slack.com/methods/add

`__call__``(`*comment*, *file*`)`

Add a comment to an existing file.

Parameters

- **`comment`** – Required. Text of the comment to add. e.g. Everyone should take a moment to read this file.
- **`file`** – Required. File to add a comment to. e.g. F1234467890

class `slackly.api.endpoints.files.`**`CommentsDelete`**`(`*bind=None*`)`

Delete an existing comment on a file. Only the original author of the comment or a Team Administrator may delete a file comment.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/delete

`__call__``(`*file*, *id*`)`

Deletes an existing comment on a file.

Parameters

- **`file`** – Required. File to delete a comment from. e.g. F1234567890
- **`id`** – Required. The comment to delete. e.g. Fc1234567890

class `slackly.api.endpoints.files.`**`CommentsEdit`**`(`*bind=None*`)`

Edit an existing comment on a file. Only the user who created a comment may make edits. Teams may configure a limited time window during which file comment edits are allowed.

If successful, the response will include a file comment object.

```
{
    "ok": true,
    "comment": {
        "id": "Fc1234567890",
        "created": 1356032811,
        "timestamp": 1356032811,
        "user": "U1234567890",
        "comment": "Everyone should take a moment to read this file, seriously."
    }
}
```

For more information see https://api.slack.com/methods/edit

__call__(*comment*, *file*, *id*)

Edit an existing file comment.

Parameters

- **comment** – Required. Text of the comment to edit. e.g. Everyone should take a moment to read this file, seriously.
- **file** – Required. File containing the comment to edit. e.g. F1234567890
- **id** – Required. The comment to edit. e.g. Fc1234567890

class slackly.api.endpoints.files.**FilesDelete**(*bind=None*)

This method deletes a file from your team.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/delete

__call__(*file*)

Deletes a file.

Parameters **file** – Required. ID of file to delete. e.g. F1234567890

class slackly.api.endpoints.files.**FilesInfo**(*bind=None*)

This method returns information about a file in your team.

The response contains a file object, and a list of comment objects followed by paging information.

```
{
    "ok": true,
    "file": {
        "id" : "F2147483862",
        "timestamp" : 1356032811,

        "name" : "file.htm",
        "title" : "My HTML file",
        "mimetype" : "text\/plain",
        "filetype" : "text",
        "pretty_type": "Text",
        "user" : "U2147483697",

        "mode" : "hosted",
        "editable" : true,
        "is_external": false,
        "external_type": "",

        "size" : 12345,

        "url": "https:\/\/slack-files.com\/files-pub\/T024BE7LD-F024BERPE-
↪09acb6\/1.png",
        "url_download": "https:\/\/slack-files.com\/files-pub\/T024BE7LD-
↪F024BERPE-09acb6\/download\/1.png",
        "url_private": "https:\/\/slack.com\/files-pri\/T024BE7LD-
↪F024BERPE\/1.png",
        "url_private_download": "https:\/\/slack.com\/files-pri\/
↪T024BE7LD-F024BERPE\/download\/1.png",

        "thumb_64": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_64.png",
        "thumb_80": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_80.png",
        "thumb_360": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_360.png",
        "thumb_360_gif": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_360.gif",
        "thumb_360_w": 100,
        "thumb_360_h": 100,

        "permalink": "https:\/\/tinyspeck.slack.com\/files\/cal\/
↪F024BERPE\/1.png",
        "edit_link": "https:\/\/tinyspeck.slack.com\/files\/cal\/
↪F024BERPE\/1.png/edit",
        "preview": "&lt;!DOCTYPE html&gt;
```

&lt;html&gt; &lt;meta charset='utf-8'&gt;",

"preview_highlight": "&lt;div class="sssh-code"&gt;&lt;div class="sssh-line"&gt;&lt;pre&gt;&lt;!DOCTYPE html...", "lines" : 123, "lines_more": 118,

"is_public": true, "public_url_shared": false, "channels": ["C024BE7LT", ...], "groups": ["G12345", ...], "initial_comment": {...}, "num_stars": 7, "is_starred": true

}, "comments": [

{ "id": "Fc027BN9L9", "timestamp": 1356032811, "user": "U2147483697", "comment": "This is a comment"

], "paging": {

"count": 100, "total": 2, "page": 1, "pages": 0

}

}

The file object contains information about the uploaded file. Each comment object in the comments array contains details about a single comment. Comments are returned oldest first. The paging information contains the count of comments returned, the total number of comments, the page of results returned in this response and the total number of pages available. Please note that the max count value is 1000 and the max page value is 100. Bot user tokens may use this method to access information about files appearing in the channels they belong to.

For more information see https://api.slack.com/methods/info

`__call__` (*file*, *count=None*, *page=None*)
Gets information about a team file.

Parameters

- **`file`** – Required. Specify a file by providing its ID. e.g. F2147483862
- **`count`** – Optional, default=100. Number of items to return per page. e.g. 20
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2

class `slackly.api.endpoints.files.`**`FilesList`**(*bind=None*)

This method returns a list of files within the team. It can be filtered and sliced in various ways.

```
{
    "ok": true,
    "files": [
        {...},
        {...},
        {...},
        ...
    ],
    "paging": {
        "count": 100,
        "total": 295,
        "page": 1,
        "pages": 3
    }
}
```

The response contains a list of file objects, followed by paging information. Files are always returned with the most recent first. The paging information contains the count of files returned, the total number of files matching the filter (if any was supplied), the page of results returned in this response and the total number of pages available.

For more information see https://api.slack.com/methods/list

`__call__`(*channel=None, count=None, page=None, ts_from=None, ts_to=None, types=None, user=None*)

Lists & filters team files.

param channel Optional. Filter files appearing in a specific channel, indicated by its ID. e.g. C1234567890

param count Optional, default=100. Number of items to return per page. e.g. 20

param page Optional, default=1. Page number of results to return. e.g. 2

param ts_from Optional, default=0. Filter files created after this timestamp (inclusive). e.g. 123456789

param ts_to Optional, default=now. Filter files created before this timestamp (inclusive). e.g. 123456789

param types Optional, default=all. Filter files by type:

all - All files spaces - Posts snippets - Snippets images - Image files gdocs - Google docs zips - Zip files pdfs - PDF files

You can pass multiple values in the types argument, like types=spaces,snippets.The default value is all, which does no

param user Optional. Filter files created by a single user. e.g. U1234567890

class `slackly.api.endpoints.files.`**`FilesRevokePublicURL`**(*bind=None*)

This method disables public/external sharing for a file.

The response contains a file object.

For more information see https://api.slack.com/methods/revokePublicURL

`__call__`(*file*)

Revokes public/external sharing access for a file

Parameters **`file`** – Required. File to revoke e.g. F1234567890

class `slackly.api.endpoints.files.`**`FilesSharedPublicURL`**(*bind=None*)

This method enables public/external sharing for a file.

The response contains a file object, including the permalink_public url.

```
{
    "ok": true,
    "file": {
        "id" : "F2147483862",
        "timestamp" : 1356032811,

        "name" : "file.htm",
        "title" : "My HTML file",
        "mimetype" : "text\/plain",
        "filetype" : "text",
        "pretty_type": "Text",
        "user" : "U2147483697",

        "mode" : "hosted",
        "editable" : true,
        "is_external": false,
        "external_type": "",

        "size" : 12345,

        "url": "https:\/\/slack-files.com\/files-pub\/T024BE7LD-F024BERPE-
↪09acb6\/1.png",
        "url_download": "https:\/\/slack-files.com\/files-pub\/T024BE7LD-
↪F024BERPE-09acb6\/download\/1.png",
        "url_private": "https:\/\/slack.com\/files-pri\/T024BE7LD-
↪F024BERPE\/1.png",
        "url_private_download": "https:\/\/slack.com\/files-pri\/
↪T024BE7LD-F024BERPE\/download\/1.png",

        "thumb_64": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_64.png",
        "thumb_80": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_80.png",
        "thumb_360": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_360.png",
        "thumb_360_gif": "https:\/\/slack-files.com\/files-tmb\/T024BE7LD-
↪F024BERPE-c66246\/1_360.gif",
        "thumb_360_w": 100,
        "thumb_360_h": 100,

        "permalink": "https:\/\/tinyspeck.slack.com\/files\/cal\/
↪F024BERPE\/1.png",
        "permalink_public": "https:\/\/slack-files.com\/T024BE7LD-
↪F024BERPE-8004f909b1",
        "edit_link": "https:\/\/tinyspeck.slack.com\/files\/cal\/
↪F024BERPE\/1.png/edit",
        "preview": "&lt;!DOCTYPE html&gt;
```

&lt;html&gt; &lt;meta charset='utf-8'&gt;",

"preview_highlight": "&lt;div class="sssh-code"&gt;&lt;div class="sssh-line"&gt;&lt;pre&gt;&lt;!DOCTYPE html...", "lines" : 123, "lines_more": 118,

"is_public": true, "public_url_shared": false, "channels": ["C024BE7LT",

...], “groups”: [”G12345”, ...], “initial_comment”: {...}, “num_stars”: 7, “is_starred”: true

}, “comments”: [

{ “id”: “Fc027BN9L9”, “timestamp”: 1356032811, “user”: “U2147483697”, “comment”: “This is a comment”

], “paging”: {

“count”: 100, “total”: 2, “page”: 1, “pages”: 0

}

}

The file object contains information about the uploaded file. Each comment object in the comments array contains details about a single comment. Comments are returned oldest first. The paging information contains the count of comments returned, the total number of comments, the page of results returned in this response and the total number of pages available. Please note that the max count value is 1000 and the max page value is 100.

For more information see https://api.slack.com/methods/sharedPublicURL

`__call__`(*file*)

Enables a file for public/external sharing.

Parameters **`file`** – Required. File to share e.g. F1234567890

class `slackly.api.endpoints.files.`**`FilesUpload`**(*bind=None*)

This method allows you to create or upload an existing file.

If successful, the response will include a file object.

```
{
    "ok": true,
    "file": {...}
}
```

By default all newly-uploaded files are private and only visible to the owner. They become public once they are shared into a public channel (which can happen at upload time via the channels argument).

Examples Upload “dramacat.gif” and share it in channel, using multipart/form-data: curl -F file=@dramacat.gif -F channels=C024BE91L,#general -F token=xxxx-xxxxxxxxx-xxxx https://slack.com/api/files.upload

Create an editable file containing the text “Hello”: curl -F content=”Hello” -F token=xxxx-xxxxxxxxx-xxxx https://slack.com/api/files.upload

For more information see https://api.slack.com/methods/upload

`__call__`(*channels=None*, *content=None*, *file=None*, *filename=None*, *filetype=None*, *initial_comment=None*, *title=None*)

Uploads or creates a file.

Parameters

- **`channels`** – Optional. Comma-separated list of channel names or IDs where the file will be shared. e.g. C1234567890,C2345678901,C3456789012
- **`content`** – Optional. File contents via a POST variable. If omitting this parameter, you must provide a file. e.g. ...
- **`file`** – Optional. File contents via multipart/form-data. If omitting this parameter, you must submit content. e.g. ...

- **filename** – Optional. Filename of file. e.g. foo.txt
- **filetype** – Optional. A file type identifier. e.g. php
- **initial_comment** – Optional. Initial comment to add to file. e.g. Best!
- **title** – Optional. Title of file. e.g. My File

class slackly.api.endpoints.groups.**GroupsArchive**(*bind=None*)

This method archives a private channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/archive

__call__(*channel*)

Archives a private channel.

Parameters **channel** – Required. Private channel to archive e.g. G1234567890

class slackly.api.endpoints.groups.**GroupsClose**(*bind=None*)

This method closes a private channel.

```
{
    "ok": true
}
```

If the private channel was already closed the response will include no_op and already_closed properties:

```
{
    "ok": true,
    "no_op": true,
    "already_closed": true
}
```

For more information see https://api.slack.com/methods/close

__call__(*channel*)

Closes a private channel.

Parameters **channel** – Required. Private channel to close. e.g. G1234567890

class slackly.api.endpoints.groups.**GroupsCreate**(*bind=None*)

This method creates a private channel.

If successful, the command returns a group object, including state information:

```
{
    "ok": true,
    "group": {
        "id": "G024BE91L",
        "name": "secretplans",
        "is_group": "true",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "is_open": true,
        "last_read": "0000000000.000000",
        "latest": null,
        "unread_count": 0,
        "unread_count_display": 0,
        "members": [
            "U024BE7LH"
        ],
        "topic": {
            "value": "Secret plans on hold",
            "creator": "U024BE7LV",
            "last_set": 1369677212
        },
        "purpose": {
            "value": "Discuss secret plans that no-one else should know",
            "creator": "U024BE7LH",
            "last_set": 1360782804
        }
    }
}
```

For more information see https://api.slack.com/methods/create

__call__(*name*, *validate=None*)

Creates a private channel.

Parameters

- **name** – Required. Name of private channel to create e.g.
- **validate** – Optional. Whether to return errors on invalid channel name instead of modifying it to meet the specified criteria. e.g. true

class slackly.api.endpoints.groups.**GroupsCreateChild**(*bind=None*)

This method takes an existing private channel and performs the following steps:

<ul> <li>Renames the existing private channel (from "example" to "example-archived").</li> <li>Archives the existing private channel.</li> <li>Creates a new private channel with the name of the existing private channel.</li> <li>Adds all members of the existing private channel to the new private channel.</li> </ul>

This is useful when inviting a new member to an existing private channel while hiding all previous chat history from them. In this scenario you can call groups.createChild followed by groups.invite.

The new private channel will have a special parent_group property pointing to the original archived private channel. This will only be returned for members of both private channels, so will not be visible to any newly invited members.

If successful, the command returns the new group object:

```
{
    "ok": true,
    "group": {
        "id": "G024BE91L",
        "name": "secretplans",
        "is_group": "true",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "members": [
            "U024BE7LH"
        ],
        ...
    }
}
```

For more information see https://api.slack.com/methods/createChild

__call__(*channel*)

Clones and archives a private channel.

Parameters **channel** – Required. Private channel to clone and archive. e.g. G1234567890

class slackly.api.endpoints.groups.**GroupsHistory**(*bind=None*)

This method returns a portion of messages/events from the specified private channel. To read the entire history for a private channel, call the method with no latest or oldest arguments, and then continue paging using the instructions below.

```
{
    "ok": true,
    "latest": "1358547726.000003",
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "has_more": false
}
```

The messages array up to 100 messages between latest and oldest. If there were more than 100 messages between those two points, then has_more will be true. If a message has the same timestamp as latest or oldest it will not be included in the list, unless inclusive is true. This allows a client to fetch all messages in a hole in channel history, by calling channels.history with latest set to the oldest message they have after the hole, and oldest to the latest message they have before the hole. If the response includes has_more then the client can make another call, using the ts value of the final messages as the latest param to get the next page of messages. If there are more than 100 messages between the two timestamps then the messages returned are the ones closest to latest. In most cases an application will want the most recent messages and will page backward from there. If oldest is provided but not latest then the messages returned are those closest to oldest, allowing you to page forward through history if desired. Messages of type "message" are user-entered text messages sent to the group, while other types are events that happened within the group. All messages have both a type and a sortable ts, but the other fields depend on the type. For a list of all possible events, see the channel messages documentation. If a message has been starred by the calling user, the is_starred property will be present and true. This property is only added for starred items, so is not present in the majority of messages. The is_limited boolean property is only included for free teams that have reached the free message limit. If true, there are messages before the current result set, but they are beyond the message limit.

For more information see https://api.slack.com/methods/history

__call__(*channel*, *count=None*, *inclusive=None*, *latest=None*, *oldest=None*, *unreads=None*)

Fetches history of messages and events from a private channel.

Parameters

- **channel** – Required. Private channel to fetch history for. e.g. G1234567890
- **count** – Optional, default=100. Number of messages to return, between 1 and 1000. e.g. 100
- **inclusive** – Optional, default=0. Include messages with latest or oldest timestamp in results. e.g. true
- **latest** – Optional, default=now. End of time range of messages to include in results. e.g. 1234567890.123456
- **oldest** – Optional, default=0. Start of time range of messages to include in results. e.g. 1234567890.123456
- **unreads** – Optional, default=0. Include unread_count_display in the output? e.g. true

class slackly.api.endpoints.groups.**GroupsInfo**(*bind=None*)

This method returns information about a private channel.

Returns a group object:

```
{
    "ok": true,
    "group": {
        "id": "G024BE91L",
        "name": "secretplans",
        "is_group": "true",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "members": [
            "U024BE7LH"
        ],

        "topic": { ... },
        "purpose": { ... },

        "last_read": "1401383885.000061",
        "latest": { ... }
        "unread_count": 0,
        "unread_count_display": 0

    },
}
```

For more information see https://api.slack.com/methods/info

__call__(*channel*)

Gets information about a private channel.

Parameters **channel** – Required. Private channel to get info on e.g. G1234567890

class slackly.api.endpoints.groups.**GroupsInvite**(*bind=None*)

This method is used to invite a user to a private channel. The calling user must be a member of the private channel.

To invite a new member to a private channel without giving them access to the archives of the private channel, call the groups.createChild method before inviting.

If successful, the API response includes a group object:

```
{
    "ok": true,
    "group": {
        ...
    },
}
```

If the invited user is already in the private channel, the response will include an already_in_group property:

```
{
    "ok": true,
    "already_in_group": true,
    "group": {
        ...
    },
}
```

For more information see https://api.slack.com/methods/invite

`__call__`(*channel*, *user*)
Invites a user to a private channel.

Parameters

- **`channel`** – Required. Private channel to invite user to. e.g. G1234567890
- **`user`** – Required. User to invite. e.g. U1234567890

class `slackly.api.endpoints.groups.`**`GroupsKick`**(*bind=None*)
This method allows a user to remove another member from a private channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/kick

`__call__`(*channel*, *user*)
Removes a user from a private channel.

Parameters

- **`channel`** – Required. Private channel to remove user from. e.g. G1234567890
- **`user`** – Required. User to remove from private channel. e.g. U1234567890

class `slackly.api.endpoints.groups.`**`GroupsLeave`**(*bind=None*)
This method is used to leave a private channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/leave

`__call__`(*channel*)
Leaves a private channel.

Parameters **`channel`** – Required. Private channel to leave e.g. G1234567890

class `slackly.api.endpoints.groups.`**`GroupsList`**`(`*bind=None*`)`

This method returns a list of private channels in the team that the caller is in and archived groups that the caller was in. The list of (non-deactivated) members in each private channel is also returned.

Returns a list of group objects (also known as "private channel objects"):

```
{
    "ok": true,
    "groups": [
        {
            "id": "G024BE91L",
            "name": "secretplans",
            "created": 1360782804,
            "creator": "U024BE7LH",
            "is_archived": false,
            "members": [
                "U024BE7LH"
            ],
            "topic": {
                "value": "Secret plans on hold",
                "creator": "U024BE7LV",
                "last_set": 1369677212
            },
            "purpose": {
                "value": "Discuss secret plans that no-one else should know",
                "creator": "U024BE7LH",
                "last_set": 1360782804
            }
        },
        ....
    ]
}
```

For more information see https://api.slack.com/methods/list

`__call__``(`*exclude_archived=None*`)`

Lists private channels that the calling user has access to.

Parameters **`exclude_archived`** – Optional, default=0. Don't return archived private channels. e.g. true

class `slackly.api.endpoints.groups.`**`GroupsMark`**`(`*bind=None*`)`

This method moves the read cursor in a private channel.

```
{
    "ok": true
}
```

After making this call, the mark is saved to the database and broadcast via the message server to all open connections for the calling user. Clients should try to avoid making this call too often. When needing to mark a read position, a client should set a timer before making the call. In this way, any further updates needed during the timeout will not generate extra calls (just one per channel). This is useful for when reading scroll-back history, or following a busy live channel. A timeout of 5 seconds is a good starting point. Be sure to flush these calls on shutdown/logout.

For more information see https://api.slack.com/methods/mark

`__call__``(`*channel, ts*`)`

Sets the read cursor in a private channel.

Parameters

- **channel** – Required. Private channel to set reading cursor in. e.g. G1234567890
- **ts** – Required. Timestamp of the most recently seen message. e.g. 1234567890.123456

class slackly.api.endpoints.groups.**GroupsOpen**(*bind=None*)

This method opens a private channel.

```
{
    "ok": true
}
```

If the private channel was already open the response will include no_op and already_open properties:

```
{
    "ok": true,
    "no_op": true,
    "already_open": true
}
```

For more information see https://api.slack.com/methods/open

__call__(*channel*)

Opens a private channel.

Parameters **channel** – Required. Private channel to open. e.g. G1234567890

class slackly.api.endpoints.groups.**GroupsRename**(*bind=None*)

This method renames a private channel.

```
{
    "ok": true,
    "channel": {
        "id": "C024BE91L",
        "is_group": true,
        "name": "new_name",
        "created": 1360782804
    }
}
```

Returns the channel ID, name and date created (as a unix timestamp).

For more information see https://api.slack.com/methods/rename

__call__(*channel*, *name*, *validate=None*)

Renames a private channel.

Parameters

- **channel** – Required. Private channel to rename e.g. G1234567890
- **name** – Required. New name for private channel. e.g.
- **validate** – Optional. Whether to return errors on invalid channel name instead of modifying it to meet the specified criteria. e.g. true

class slackly.api.endpoints.groups.**GroupsReplies**(*bind=None*)

This method returns an entire thread (a message plus all the messages in reply to it).

```
{
    "ok": true,
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "thread_info": [
        "complete": true,
        "count": 3,
    ]
}
```

For more information see https://api.slack.com/methods/replies

`__call__`(*channel*, *thread_ts*)

Retrieve a thread of messages posted to a private channel

Parameters

- **`channel`** – Required. Private channel to fetch thread from e.g. C1234567890
- **`thread_ts`** – Required. Unique identifier of a thread's parent message e.g. 1234567890.123456

class `slackly.api.endpoints.groups.`**`GroupsSetPurpose`**(*bind=None*)

This method is used to change the purpose of a private channel. The calling user must be a member of the private channel.

```
{
    "ok": true,
    "purpose": "This is the new purpose!"
}
```

For more information see https://api.slack.com/methods/setPurpose

`__call__`(*channel*, *purpose*)

Sets the purpose for a private channel.

Parameters

- **`channel`** – Required. Private channel to set the purpose of e.g. G1234567890
- **`purpose`** – Required. The new purpose e.g. My Purpose

class `slackly.api.endpoints.groups.`**`GroupsSetTopic`**(*bind=None*)

This method is used to change the topic of a private channel. The calling user must be a member of the private channel.

```
{
    "ok": true,
    "topic": "This is the new topic!"
}
```

For more information see https://api.slack.com/methods/setTopic

`__call__`(*channel*, *topic*)

Sets the topic for a private channel.

Parameters

- **`channel`** – Required. Private channel to set the topic of e.g. G1234567890
- **`topic`** – Required. The new topic e.g. My Topic

class `slackly.api.endpoints.groups.`**`GroupsUnarchive`**(*bind=None*)

This method unarchives a private channel.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/unarchive

`__call__`(*channel*)

Unarchives a private channel.

Parameters **`channel`** – Required. Private channel to unarchive e.g. G1234567890

class `slackly.api.endpoints.im.`**`ImClose`**(*bind=None*)

This method closes a direct message channel.

```
{
    "ok": true
}
```

If the channel was already closed the response will include a no_op property:

```
{
    "ok": true,
    "no_op": true,
    "already_closed": true
}
```

For more information see https://api.slack.com/methods/close

`__call__`(*channel*)

Close a direct message channel.

Parameters **`channel`** – Required. Direct message channel to close. e.g. D1234567890

class `slackly.api.endpoints.im.`**`ImHistory`**(*bind=None*)

This method returns a portion of messages/events from the specified direct message channel. To read the entire history for a direct message channel, call the method with no latest or oldest arguments, and then continue paging using the instructions below.

```
{
    "ok": true,
    "latest": "1358547726.000003",
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "has_more": false
}
```

The messages array up to 100 messages between latest and oldest. If there were more than 100 messages between those two points, then has_more will be true. If a message has the same timestamp as latest or oldest it will not be included in the list, unless inclusive is true. This allows a client to fetch all messages in a hole in channel history, by calling channels.history with latest set to the oldest message they have after the hole, and oldest to the latest message they have before the hole. If the response includes has_more then the client can make another call, using the ts value of the final messages as the latest param to get the next page of messages. If there are more than 100 messages between the two timestamps then the messages returned are the ones closest to latest. In most cases an application will want the most recent messages and will page backward from there. If oldest is provided but not latest then the messages returned are those closest to oldest, allowing you to page forward through history if desired. Messages of type "message" are user-entered text messages sent to the direct message channel, while other types are events that happened within the direct message channel. All messages have both a type and a sortable ts, but the other fields depend on the type. For a list of all possible events, see the channel messages documentation. If a message has been starred by the calling user, the is_starred property will be present and true. This property is only added for starred items, so is not present in the majority of messages. The is_limited boolean property is only included for free teams that have reached the free message limit. If true, there are messages before the current result set, but they are beyond the message limit.

For more information see https://api.slack.com/methods/history

`__call__`(*channel*, *count=None*, *inclusive=None*, *latest=None*, *oldest=None*, *unreads=None*)

Fetches history of messages and events from direct message channel.

Parameters

- **`channel`** – Required. Direct message channel to fetch history for. e.g. D1234567890
- **`count`** – Optional, default=100. Number of messages to return, between 1 and 1000. e.g. 100
- **`inclusive`** – Optional, default=0. Include messages with latest or oldest timestamp in results. e.g. true
- **`latest`** – Optional, default=now. End of time range of messages to include in results. e.g. 1234567890.123456
- **`oldest`** – Optional, default=0. Start of time range of messages to include in results. e.g. 1234567890.123456

- **unreads** – Optional, default=0. Include unread_count_display in the output? e.g. true

class `slackly.api.endpoints.im.`**`ImList`**(*bind=None*)

This method returns a list of all im channels that the user has.

Returns a list of IM objects:

```
{
    "ok": true,
    "ims": [
        {
           "id": "D024BFF1M",
           "is_im": true,
           "user": "USLACKBOT",
           "created": 1372105335,
           "is_user_deleted": false
        },
        {
           "id": "D024BE7RE",
           "is_im": true,
           "user": "U024BE7LH",
           "created": 1356250715,
           "is_user_deleted": false
        },
        ...
    ]
}
```

For more information see https://api.slack.com/methods/list

`__call__`()

Lists direct message channels for the calling user.

class `slackly.api.endpoints.im.`**`ImMark`**(*bind=None*)

This method moves the read cursor in a direct message channel.

```
{
    "ok": true
}
```

After making this call, the mark is saved to the database and broadcast via the message server to all open connections for the calling user. Clients should try to avoid making this call too often. When needing to mark a read position, a client should set a timer before making the call. In this way, any further updates needed during the timeout will not generate extra calls (just one per channel). This is useful for when reading scroll-back history, or following a busy live channel. A timeout of 5 seconds is a good starting point. Be sure to flush these calls on shutdown/logout.

For more information see https://api.slack.com/methods/mark

`__call__`(*channel*, *ts*)

Sets the read cursor in a direct message channel.

Parameters

- **channel** – Required. Direct message channel to set reading cursor in. e.g. D1234567890
- **ts** – Required. Timestamp of the most recently seen message. e.g. 1234567890.123456

class `slackly.api.endpoints.im.`**`ImOpen`**(*bind=None*)

This method opens a direct message channel with another member of your Slack team.

```
{
    "ok": true,
    "channel": {
        "id": "D024BFF1M"
    }
}
```

If the channel was already open the response will include no_op and already_open properties:

```
{
    "ok": true,
    "no_op": true,
    "already_open": true,
    "channel": {
        "id": "D024BFF1M"
    }
}
```

In either case, if the return_im argument was passed, the channel object will contain the full channel definition:

```
{
    "ok": true,
    "channel": {
        "id":"D024BE91L",
        "is_im":true,
        "user":"U024BE7LH",
        "created":1434412652,
        "last_read":"1442525627.000002",
        "latest":{
            "type":"message",
            "user":"U024BE7LH",
            "text":"hello",
            "ts":"1442525627.000002"
        },
        "unread_count":0,
        "unread_count_display":0,
        "is_open":true
    }
}
```

For more information see https://api.slack.com/methods/open

`__call__`(*user*, *return_im=None*)

Opens a direct message channel.

Parameters

- **`user`** – Required. User to open a direct message channel with. e.g. U1234567890
- **`return_im`** – Optional. Boolean, indicates you want the full IM channel definition in the response. e.g. true

class `slackly.api.endpoints.im.`**`ImReplies`**(*bind=None*)

This method returns an entire thread (a message plus all the messages in reply to it).

```
{
    "ok": true,
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "thread_info": [
        "complete": true,
        "count": 3,
    ]
}
```

For more information see https://api.slack.com/methods/replies

__call__(*channel*, *thread_ts*)

Retrieve a thread of messages posted to a direct message conversation

Parameters

- **channel** – Required. Direct message channel to fetch thread from e.g. C1234567890
- **thread_ts** – Required. Unique identifier of a thread's parent message e.g. 1234567890.123456

class slackly.api.endpoints.mpim.**MpimClose**(*bind=None*)

This method closes a multiparty direct message channel.

```
{
    "ok": true
}
```

If the mpim was already closed the response will include no_op and already_closed properties:

```
{
    "ok": true,
    "no_op": true,
    "already_closed": true
}
```

For more information see https://api.slack.com/methods/close

__call__(*channel*)

Closes a multiparty direct message channel.

Parameters **channel** – Required. MPIM to close. e.g. G1234567890

class slackly.api.endpoints.mpim.**MpimHistory**(*bind=None*)

This method returns a portion of messages/events from the specified multiparty direct message channel. To read the entire history for a multiparty direct message, call the method with no latest or oldest arguments, and then continue paging using the instructions below.

```
{
    "ok": true,
    "latest": "1358547726.000003",
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "has_more": false
}
```

The messages array up to 100 messages between latest and oldest. If there were more than 100 messages between those two points, then has_more will be true. If a message has the same timestamp as latest or oldest it will not be included in the list, unless inclusive is true. This allows a client to fetch all messages in a hole in channel history, by calling channels.history with latest set to the oldest message they have after the hole, and oldest to the latest message they have before the hole. If the response includes has_more then the client can make another call, using the ts value of the final messages as the latest param to get the next page of messages. If there are more than 100 messages between the two timestamps then the messages returned are the ones closest to latest. In most cases an application will want the most recent messages and will page backward from there. If oldest is provided but not latest then the messages returned are those closest to oldest, allowing you to page forward through history if desired. Messages of type “message” are user-entered text messages sent to the multiparty direct message channel, while other types are events that happened within the multiparty direct message channel. All messages have both a type and a sortable ts, but the other fields depend on the type. For a list of all possible events, see the channel messages documentation. If a message has been starred by the calling user, the is_starred property will be present and true. This property is only added for starred items, so is not present in the majority of messages. The is_limited boolean property is only included for free teams that have reached the free message limit. If true, there are messages before the current result set, but they are beyond the message limit.

For more information see https://api.slack.com/methods/history

`__call__`(*channel*, *count=None*, *inclusive=None*, *latest=None*, *oldest=None*, *unreads=None*)

Fetches history of messages and events from a multiparty direct message.

Parameters

- **`channel`** – Required. Multiparty direct message to fetch history for. e.g. G1234567890
- **`count`** – Optional, default=100. Number of messages to return, between 1 and 1000. e.g. 100

- **`inclusive`** – Optional, default=0. Include messages with latest or oldest timestamp in results. e.g. true
- **`latest`** – Optional, default=now. End of time range of messages to include in results. e.g. 1234567890.123456
- **`oldest`** – Optional, default=0. Start of time range of messages to include in results. e.g. 1234567890.123456
- **`unreads`** – Optional, default=0. Include unread_count_display in the output? e.g. true

class `slackly.api.endpoints.mpim.`**`MpimList`**(*bind=None*)

This method returns a list of all multiparty direct message channels that the user has.

Returns a list of group objects:

```
{
    "ok": true,
    "groups": [
        {
            "id": "G024BE91L",
            "name": "dm-messaging-user-1",
            "created": 1360782804,
            "creator": "U024BE7LH",
            "is_archived": false,
            "is_mpim": true
            "members": [
                "U024BE7LH",
                "U1234567890",
                "U2345678901",
                "U3456789012"
            ],
            "topic": {
                "value": "Group messaging.",
                "creator": "U024BE7LH",
                "last_set": 1360782804
            },
            "purpose": {
                "value": "Group messaging with: @user @user_a @user_b @user_c",
                "creator": "U024BE7LH",
                "last_set": 1360782804
            }
        },
        ....
    ]
}
```

For more information see https://api.slack.com/methods/list

`__call__`()

Lists multiparty direct message channels for the calling user.

class `slackly.api.endpoints.mpim.`**`MpimMark`**(*bind=None*)

This method moves the read cursor in a multiparty direct message channel.

```
{
    "ok": true
}
```

After making this call, the mark is saved to the database and broadcast via the message server to all open connections for the calling user. Clients should try to avoid making this call too often. When needing to mark a read position, a client should set a timer before making the call. In this way, any further updates needed during the timeout will not generate extra calls (just one per channel). This is useful for when reading scroll-back history, or following a busy live channel. A timeout of 5 seconds is a good starting point. Be sure to flush these calls on shutdown/logout.

For more information see https://api.slack.com/methods/mark

__call__(*channel*, *ts*)

Sets the read cursor in a multiparty direct message channel.

Parameters

- **channel** – Required. multiparty direct message channel to set reading cursor in. e.g. G1234567890
- **ts** – Required. Timestamp of the most recently seen message. e.g. 1234567890.123456

class slackly.api.endpoints.mpim.**MpimOpen**(*bind=None*)

This method opens a multiparty direct message.

Opening a multiparty direct message takes a list of up-to 8 encoded user ids. If there is no MPIM already created that includes that exact set of members, a new MPIM will be created. Subsequent calls to mpim.open with the same set of users will return the already existing MPIM conversation.

If successful, the command returns a mpim object, including state information:

```
{
    "ok": true,
    "group": {
        "id": "G024BE91L",
        "name": "mpdm-user--user_a--user_b--user_c-1",
        "is_group": "true",
        "created": 1360782804,
        "creator": "U024BE7LH",
        "is_archived": false,
        "is_open": false,
        "is_mpim": true,
        "last_read": "0000000000.000000",
        "latest": null,
        "unread_count": 0,
        "unread_count_display": 0,
        "members": [
            "U024BE7LH",
            "U1234567890",
            "U2345678901",
            "U3456789012"
        ],
        "topic": {
            "value": "Group messaging.",
            "creator": "U024BE7LH",
            "last_set": 1360782804
        },
        "purpose": {
            "value": "Group messaging with: @user @user_a @user_b @user_c",
            "creator": "U024BE7LH",
            "last_set": 1360782804
        }
    }
}
```

For more information see https://api.slack.com/methods/open

`__call__`(*users*)

This method opens a multiparty direct message.

Parameters **`users`** – Required. Comma separated lists of users. The ordering of the users is preserved whenever a MPIM group is returned. e.g. U1234567890,U2345678901,U3456789012

class `slackly.api.endpoints.mpim.`**`MpimReplies`**(*bind=None*)

This method returns an entire thread (a message plus all the messages in reply to it).

```
{
    "ok": true,
    "messages": [
        {
            "type": "message",
            "ts": "1358546515.000008",
            "user": "U2147483896",
            "text": "Hello"
        },
        {
            "type": "message",
            "ts": "1358546515.000007",
            "user": "U2147483896",
            "text": "World",
            "is_starred": true,
        },
        {
            "type": "something_else",
            "ts": "1358546515.000007",
            "wibblr": true
        }
    ],
    "thread_info": [
        "complete": true,
        "count": 3,
    ]
}
```

For more information see https://api.slack.com/methods/replies

`__call__`(*channel*, *thread_ts*)

Retrieve a thread of messages posted to a direct message conversation from a multiparty direct message.

Parameters

- **`channel`** – Required. Multiparty direct message channel to fetch thread from. e.g. C1234567890
- **`thread_ts`** – Required. Unique identifier of a thread's parent message. e.g. 1234567890.123456

class `slackly.api.endpoints.oauth.`**`OauthAccess`**(*bind=None*)

This method allows you to exchange a temporary OAuth code for an API access token. This is used as part of the OAuth authentication flow.

As discussed in RFC 6749 it is possible to supply the Client ID and Client Secret using the HTTP Basic authentication scheme. If HTTP Basic authentication is used you do not need to supply the client_id and client_secret parameters as part of the request.

Keep your tokens secure. Do not share tokens with users or anyone else.

```
{
    "access_token": "xoxp-23984754863-2348975623103",
    "scope": "read"
}
```

You can use the returned token to call protected API methods on behalf of the user.

For more information see https://api.slack.com/methods/access

`__call__`(*client_id*, *client_secret*, *code*, *redirect_uri=None*)
Exchanges a temporary OAuth code for an API token.

Parameters

- **`client_id`** – Required. Issued when you created your application. e.g. 4b39e9-752c4
- **`client_secret`** – Required. Issued when you created your application. e.g. 33fea0113f5b1
- **`code`** – Required. The code param returned via the OAuth callback. e.g. ccdaa72ad
- **`redirect_uri`** – Optional. This must match the originally submitted URI (if one was sent). e.g. http://example.com

class `slackly.api.endpoints.pins.`**`PinsAdd`**(*bind=None*)
This method pins an item (file, file comment, channel message, or group message) to a particular channel. The channel argument is required and one of file, file_comment, or timestamp must also be specified.

```
{
    "ok": true
}
```

After making this call the pin is saved to the database and a pin_added event is broadcast via the RTM API.

For more information see https://api.slack.com/methods/add

`__call__`(*channel*, *file=None*, *file_comment=None*, *timestamp=None*)
Pins an item to a channel.

Parameters

- **`channel`** – Required. Channel to pin the item in. e.g. C1234567890
- **`file`** – Optional. File to pin. e.g. F1234567890
- **`file_comment`** – Optional. File comment to pin. e.g. Fc1234567890
- **`timestamp`** – Optional. Timestamp of the message to pin. e.g. 1234567890.123456

class `slackly.api.endpoints.pins.`**`PinsList`**(*bind=None*)
This method lists the items pinned to a channel.

The response contains a list of pinned items in a channel.

```
{
    "ok": true,
    "items": [
        {
            "type": "message",
            "channel": "C2147483705",
            "message": {...},
            "created": 1456335673,
            "created_by": "U07BVMD97"
        },
        {
            "type": "file",
            "file": { ... },
            "created": 1456335673,
            "created_by": "U07BVMD97"
        },
        {
            "type": "file_comment",
            "file": { ... },
            "comment": { ... },
            "created": 1456335673,
            "created_by": "U07BVMD97"
        }
    ]
}
```

Different item types can be pinned. Every item in the list has a type property, and the other properties depend on the type of item. The possible types are:

message: the item will have a message property containing a message object and a channel property containing the channel ID for the message. file: this item will have a file property containing a file object. file_comment: the item will have a file property containing the file object and a comment property containing the file comment.

The created property on each item is a unix timestamp representing when the item was pinned. The created_by property on each item is a string representing the encoded user id of the user who pinned the item.

For more information see https://api.slack.com/methods/list

`__call__`(*channel*)

Lists items pinned to a channel.

Parameters **`channel`** – Required. Channel to get pinned items for. e.g. C1234567890

class `slackly.api.endpoints.pins.`**`PinsRemove`**(*bind=None*)

This method un-pins an item (file, file comment, channel message, or group message) from a channel. The channel argument is required and one of file, file_comment, or timestamp must also be specified.

```
{
    "ok": true
}
```

After making this call the pin is removed from the database and a pin_removed event is broadcast via the RTM API.

For more information see https://api.slack.com/methods/remove

`__call__`(*channel*, *file=None*, *file_comment=None*, *timestamp=None*)

Un-pins an item from a channel.

Parameters

- **`channel`** – Required. Channel where the item is pinned to. e.g. C1234567890
- **`file`** – Optional. File to un-pin. e.g. F1234567890
- **`file_comment`** – Optional. File comment to un-pin. e.g. Fc1234567890
- **`timestamp`** – Optional. Timestamp of the message to un-pin. e.g. 1234567890.123456

class `slackly.api.endpoints.reactions.`**`ReactionsAdd`**(*bind=None*)

This method adds a reaction (emoji) to an item (file, file comment, channel message, group message, or direct message). One of file, file_comment, or the combination of channel and timestamp must be specified.

```
{
    "ok": true
}
```

After making this call, the reaction is saved and a reaction_added event is broadcast through the RTM API for the calling user.

For more information see https://api.slack.com/methods/add

`__call__`(*name*, *channel=None*, *file=None*, *file_comment=None*, *timestamp=None*)

Adds a reaction to an item.

Parameters

- **`name`** – Required. Reaction (emoji) name. e.g. thumbsup
- **`channel`** – Optional. Channel where the message to add reaction to was posted. e.g. C1234567890
- **`file`** – Optional. File to add reaction to. e.g. F1234567890
- **`file_comment`** – Optional. File comment to add reaction to. e.g. Fc1234567890
- **`timestamp`** – Optional. Timestamp of the message to add reaction to. e.g. 1234567890.123456

class `slackly.api.endpoints.reactions.`**`ReactionsGet`**(*bind=None*)

This method returns a list of all reactions for a single item (file, file comment, channel message, group message, or direct message).

The response contains the item with reactions.

```
{
    "type": "message",
    "channel": "C2147483705",
    "message": {
        ...
        "reactions": [
            {
                "name": "astonished",
                "count": 3,
                "users": [ "U1", "U2", "U3" ]
            },
            {
                "name": "clock1"
                "count": 2,
                "users": [ "U1", "U2", "U3" ]
            }
        ]
    },
},
```

Different item types can be reacted to. The item returned has a type property, the other properties depend on the type of item. The possible types are:

message: the item will have a message property containing a message object and a channel property containing the channel ID for the message. file: this item will have a file property containing a file object. file_comment: the item will have a file property containing the file object and a comment property containing the file comment.

The users array in the reactions property might not always contain all users that have reacted (we limit it to X users, and X might change), however count will always represent the count of all users who made that reaction (i.e. it may be greater than users.length). If the authenticated user has a given reaction then they are guaranteed to appear in the users array, regardless of whether count is greater than users.length or not. If the complete list of users is required they can still be obtained by specifying the full argument.

For more information see https://api.slack.com/methods/get

`__call__`(*channel=None*, *file=None*, *file_comment=None*, *full=None*, *timestamp=None*)

Gets reactions for an item.

Parameters

- **`channel`** – Optional. Channel where the message to get reactions for was posted. e.g. C1234567890
- **`file`** – Optional. File to get reactions for. e.g. F1234567890
- **`file_comment`** – Optional. File comment to get reactions for. e.g. Fc1234567890
- **`full`** – Optional. If true always return the complete reaction list. e.g. true
- **`timestamp`** – Optional. Timestamp of the message to get reactions for. e.g. 1234567890.123456

class `slackly.api.endpoints.reactions.`**`ReactionsList`**(*bind=None*)

This method returns a list of all items (file, file comment, channel message, group message, or direct message) reacted to by a user.

The response contains a list of items with reactions followed by pagination information.

```
{
    "ok": true,
    "items": [
        {
            "type": "message",
            "channel": "C2147483705",
            "message": {
                ...
                "reactions": [
                    {
                        "name": "astonished",
                        "count": 3,
                        "users": [ "U1", "U2", "U3" ]
                    },
                    {
                        "name": "clock1"
                        "count": 2,
                        "users": [ "U1", "U2", "U3" ]
                    }
                ]
            },
        },
        {
            "type": "file",
            "file": { ... },
            "reactions": [
                {
                    "name": "thumbsup",
                    "count": 1,
                    "users": [ "U1" ]
                }
            ]
        }
        {
            "type": "file_comment",
            "file": { ... },
            "comment": { ... },
            "reactions": [
                {
                    "name": "facepalm",
                    "count": 1034,
                    "users": [ "U1", "U2", "U3", "U4", "U5" ]
                }
            ]
        }
    ],
    "paging": {
        "count": 100,
        "total": 4,
        "page": 1,
        "pages": 1
    }
}
```

Different item types can be reacted to. Every item in the list has a type property, the other properties depend on the type of item. The possible types are:

message: the item will have a message property containing a message object and a channel property containing the channel ID for the message. file: this item will have a file property containing a file object. file_comment: the item will have a file property containing the file object and a comment property containing the file comment.

The users array in the reactions property might not always contain all users that have reacted (we limit it to X users, and X might change), however count will always represent the count of all users who made that reaction (i.e. it may be greater than users.length). If the authenticated user has a given reaction then they are guaranteed to appear in the users array, regardless of whether count is greater than users.length or not. If the complete list of users is required they can still be obtained by specifying the full argument. The paging information contains the count of items returned, the total number of items reacted to, the page of results returned in this response and the total number of pages available. Please note that the max count value is 1000 and the max page value is 100.

For more information see https://api.slack.com/methods/list

`__call__`(*count=None*, *full=None*, *page=None*, *user=None*)
Lists reactions made by a user.

Parameters

- **`count`** – Optional, default=100. Number of items to return per page. e.g. 20
- **`full`** – Optional. If true always return the complete reaction list. e.g. true
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2
- **`user`** – Optional. Show reactions made by this user. Defaults to the authed user. e.g. U1234567890

class `slackly.api.endpoints.reactions.`**`ReactionsRemove`**(*bind=None*)
This method removes a reaction (emoji) from an item (file, file comment, channel message, group message, or direct message). One of file, file_comment, or the combination of channel and timestamp must be specified.

```
{
    "ok": true
}
```

After making this call, the reaction is removed and a reaction_removed event is broadcast through the RTM API for the calling user.

For more information see https://api.slack.com/methods/remove

`__call__`(*name*, *channel=None*, *file=None*, *file_comment=None*, *timestamp=None*)

Removes a reaction from an item.

Parameters

- **`name`** – Required. Reaction (emoji) name. e.g. thumbsup
- **`channel`** – Optional. Channel where the message to remove reaction from was posted. e.g. C1234567890
- **`file`** – Optional. File to remove reaction from. e.g. F1234567890
- **`file_comment`** – Optional. File comment to remove reaction from. e.g. Fc1234567890
- **`timestamp`** – Optional. Timestamp of the message to remove reaction from. e.g. 1234567890.123456

class `slackly.api.endpoints.reminders.`**`RemindersAdd`**(*bind=None*)

This method creates a reminder.

(Note: only non-recurring reminders will have time and complete_ts field.)

```
{
    "ok": true,
    "reminder": {
        "id": "Rm12345678",
        "creator": "U18888888",
        "user": "U18888888",
        "text": "eat a banana",
        "recurring": false,
        "time": 1602288000,
        "complete_ts": 0
    }
}
```

For more information see https://api.slack.com/methods/add

`__call__`(*text*, *time*, *user=None*)

Creates a reminder.

Parameters

- **`text`** – Required. The content of the reminder e.g. eat a banana
- **`time`** – Required. When this reminder should happen: the Unix timestamp (up to five years from now), the number of seconds until the reminder (if within 24 hours), or a natural language description (Ex. “in 15 minutes,” or “every Thursday”) e.g. 1602288000
- **`user`** – Optional. The user who will receive the reminder. If no user is specified, the reminder will go to user who created it. e.g. U18888888

class `slackly.api.endpoints.reminders.`**`RemindersComplete`**(*bind=None*)

This method completes a reminder.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/complete

__call__(*reminder*)
Marks a reminder as complete.

Parameters **reminder** – Required. The ID of the reminder to be marked as complete e.g. Rm12345678

class slackly.api.endpoints.reminders.**RemindersDelete**(*bind=None*)
This method deletes a reminder.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/delete

__call__(*reminder*)
Deletes a reminder.

Parameters **reminder** – Required. The ID of the reminder e.g. Rm12345678

class slackly.api.endpoints.reminders.**RemindersInfo**(*bind=None*)
This method returns information about a reminder.

(Note: only non-recurring reminders will have time and complete_ts field.)

```
{
    "ok": true,
    "reminder": {
        "id": "Rm12345678",
        "creator": "U18888888",
        "user": "U18888888",
        "text": "eat a banana",
        "recurring": false,
        "time": 1458678068,
        "complete_ts": 1458678200
    }
}
```

For more information see https://api.slack.com/methods/info

__call__(*reminder*)
Gets information about a reminder.

Parameters **reminder** – Required. The ID of the reminder e.g. Rm23456789

class slackly.api.endpoints.reminders.**RemindersList**(*bind=None*)
This method lists all reminders created by or for a given user.

(Note: only non-recurring reminders will have time and complete_ts field.)

```
{
    "ok": true,
    "reminders": [
        {
            "id": "Rm12345678",
            "creator": "U18888888",
            "user": "U18888888",
            "text": "eat a banana",
            "recurring": false,
            "time": 1458678068,
            "complete_ts": 0
        },
        {
            "id": "Rm23456789",
            "creator": "U18888888",
            "user": "U18888888",
            "text": "drink water",
            "recurring": true
        }
    ]
}
```

For more information see https://api.slack.com/methods/list

__call__()

Lists all reminders created by or for a given user.

class slackly.api.endpoints.rtm.**RtmConnect**(*bind=None*)

This method begins a Real Time Messaging API session and reserves your application a specific URL with which to connect via websocket.

Unlike rtm.start, this method is focused only on connecting to the RTM API.

Use this method in conjunction with other Web API methods like channels.list, users.list, and team.info to build a full picture of the team or workspace you're connecting on behalf of.

Please consult the RTM API documentation for full details on using the RTM API.

This method returns a WebSocket Message Server URL and limited information about the team:

```
{
    "ok": true,
    "url": "wss:\/\/ms9.slack-msgs.com\/websocket\/2I5yBpcvk",
    "team": {
        "id": "T654321",
        "name": "Librarian Society of Soledad",
        "domain": "libsocos",
        "enterprise_id": "E234567",
        "enterprise_name": "Intercontinental Librarian Society"
    },
    "self": {
        "id": "W123456",
        "name": "brautigan"
    }
}
```

The url property contains a WebSocket Message Server URL. Connecting to this URL will initiate a Real Time Messaging session. These URLs are only valid for 30 seconds, so connect quickly! The self property contains a small amount of information concerning the connecting user &mdash; an id and their name. The team attribute also houses brief information about the team, including its id, name, domain, and if it's part of an Enterprise Grid, the corresponding enteprise_id.

For more information see https://api.slack.com/methods/connect

__call__()

Starts a Real Time Messaging session.

class slackly.api.endpoints.rtm.**RtmStart**(*bind=None*)

This method begins a Real Time Messaging API session and reserves your application a specific URL with which to connect via websocket.

It's user-centric and team-centric: your app connects as a specific user or bot user on a specific team. Many apps will find the Events API's subscription model more scalable when working against multiple teams.

This method also returns a smorgasbord of data about the team, its channels, and members. Some times more information than can be provided in a timely or helpful manner.

Please use rtm.connect instead, especially when connecting on behalf of an Enterprise Grid customer.

Consult the RTM API documentation for full details on using the RTM API. You'll also find our changelog entry useful.

This method returns lots of data about the current state of a team, along with a WebSocket Message Server URL:

```
{
    "ok": true,
    "url": "wss:\/\/ms9.slack-msgs.com\/websocket\/7I5yBpcvk",

    "self": {
        "id": "U023BECGF",
        "name": "bobby",
        "prefs": {
            ...
        },
        "created": 1402463766,
        "manual_presence": "active"
    },
    "team": {
        "id": "T024BE7LD",
        "name": "Example Team",
        "email_domain": "",
        "domain": "example",
        "icon": {
            ...
        },
        "msg_edit_window_mins": -1,
        "over_storage_limit": false
        "prefs": {
            ...
        },
        "plan": "std"
    },
    "users": [ ... ],

    "channels": [ ... ],
    "groups": [ ... ],
    "mpims": [ ... ],
    "ims": [ ... ],

    "bots": [ ... ],
}
```

The url property contains a WebSocket Message Server URL. Connecting to this URL will initiate a Real Time Messaging session. These URLs are only valid for 30 seconds, so connect quickly! The self property contains details on the authenticated user. The team property contains details on the authenticated user's team. If a team has not yet set a custom icon, the value of team.icon.image_default will be true. The users property contains a list of user objects, one for every member of the team. The channels property is a list of channel objects, one for every channel visible to the authenticated user. For regular or administrator accounts this list will include every team channel. The is_member property indicates if the user is a member of this channel. If true then the channel object will also include the topic, purpose, member list and read-state related information. The latest attribute is deprecated and will soon be removed from this method's response. See this changelog entry. The groups property is a list of group objects, one for every group the authenticated user is in. The mpims property is a list of mpims objects, one for every group the authenticated user is in. MPIMs are only returned to the client if mpim_aware is set when calling rtm.start. Otherwise, mpims are emulated using the groups API. The ims property is a list of IM objects, one for every direct message channel visible to the authenticated user. The bots property gives details of the integrations set up on this team.

For more information see https://api.slack.com/methods/start

`__call__`(*mpim_aware=None*, *no_latest=None*, *no_unreads=None*, *simple_latest=None*)

Starts a Real Time Messaging session.

Parameters

- **`mpim_aware`** – Optional. Returns MPIMs to the client in the API response. e.g. true
- **`no_latest`** – Optional, default=0. Exclude latest timestamps for channels, groups, mpims, and ims. Automatically sets no_unreads to 1 e.g. 1
- **`no_unreads`** – Optional. Skip unread counts for each channel (improves performance). e.g. true
- **`simple_latest`** – Optional. Return timestamp only for latest message object of each channel (improves performance). e.g. true

class `slackly.api.endpoints.search.`**`SearchAll`**(*bind=None*)

This method allows users and applications to search both messages and files in a single call.

The response returns matches broken down by their type of content, similar to the facebook/gmail auto-completed search widgets.

```
{
    "ok": true,
    "query": "Best Pickles",
    "messages": {...},
    "files": {...}
}
```

Within each content group, data is returned in the following format:

```
{
    "matches": [],
    "paging": {
        "count": 100,  - number of records per page
        "total": 15,   - total records matching query
        "page": 1,     - page of records returned
        "pages": 1     - total pages matching query
    }
}
```

This block gives the (estimated) total number of matches of this type, then has an array containing the specified page of the top matches. The format of matches depends on the match type, as described in the documentation for search.messages and search.files. These methods can be used to fetch further pages of messages or files.

For more information see https://api.slack.com/methods/all

`__call__`(*query*, *count=None*, *highlight=None*, *page=None*, *sort=None*, *sort_dir=None*)
Searches for messages and files matching a query.

Parameters

- **`query`** – Required. Search query. May contains booleans, etc. e.g. pickleface
- **`count`** – Optional, default=20. Number of items to return per page. e.g. 20
- **`highlight`** – Optional. Pass a value of true to enable query highlight markers (see below). e.g. true
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2
- **`sort`** – Optional, default=score. Return matches sorted by either score or timestamp. e.g. timestamp
- **`sort_dir`** – Optional, default=desc. Change sort direction to ascending (asc) or descending (desc). e.g. asc

class `slackly.api.endpoints.search.`**`SearchFiles`**(*bind=None*)
This method returns files matching a search query.

The response envelope contains paging and result information:

```
{
    "ok": true,
    "query": "test",
    "files": {
        "total": 829,
        "paging": {
            "count": 20,
            "total": 829,
            "page": 1,
            "pages": 42
        },
        "matches": [
            {...},
            {...},
            {...}
        ]
    }
}
```

Matches contains a list of file objects. All search methods support the highlight parameter. If specified, the matching query terms will be marked up in the results so that clients may replace them with appropriate highlighting markers (e.g. <span class=”highlight”></span>). The UTF-8 markers we use are: start: “î”; # U+E000 (private-use) end : “î”; # U+E001 (private-use)

Please note that the max count value is 1000 and the max page value is 100.

For more information see https://api.slack.com/methods/files

`__call__`(*query*, *count=None*, *highlight=None*, *page=None*, *sort=None*, *sort_dir=None*)
Searches for files matching a query.

Parameters

- **`query`** – Required. Search query. May contain booleans, etc. e.g. pickleface
- **`count`** – Optional, default=20. Number of items to return per page. e.g. 20
- **`highlight`** – Optional. Pass a value of true to enable query highlight markers (see below). e.g. true

- **page** – Optional, default=1. Page number of results to return. e.g. 2
- **sort** – Optional, default=score. Return matches sorted by either score or timestamp. e.g. timestamp
- **sort_dir** – Optional, default=desc. Change sort direction to ascending (asc) or descending (desc). e.g. asc

class slackly.api.endpoints.search.**SearchMessages**(*bind=None*)

This method returns messages matching a search query.

The response envelope contains paging and result information:

```
{
    "ok": true,
    "query": "test",
    "messages": {
        "total": 829,
        "paging": {
            "count": 20,
            "total": 829,
            "page": 1,
            "pages": 42
        },
        "matches": [
            {...},
            {...},
            {...}
        ]
    }
}
```

The actual matches are returned as hashes containing contextual messages:

```
{
    "type": "message",
    "channel": {
        "id": "C2147483753",
        "name": "foo"
    },
    "user": "U2147483709",
    "username": "johnnytest",
    "ts": "1359414002.000003",
    "text": "mention test: johnnyrodgers".
    "permalink": "https:\/\/example.slack.com\/channels\/foo\/p1359414002000003",
    "previous_2": {
        "user": "U2147483709",
        "username": "johnnytest",
        "text": "This was said before before",
        "ts": "1359413987.000000",
        "type": "message"
    },
    "previous": {
        "user": "U2147483709",
        "username": "johnnytest",
        "text": "This was said before",
        "ts": "1359414001.000000",
        "type": "message"
    },
    "next": {
        "user": "U2147483709",
        "username": "johnnytest",
        "text": "This was said after",
        "ts": "1359414020.000000",
        "type": "message"
    },
    "next_2": {
        "user": "U2147483709",
        "username": "johnnytest",
        "text": "This was said after after",
        "ts": "1359414021.000000",
        "type": "message"
    }
}
```

Messages are searched primarily inside the message text themselves, with a lower priority on the messages immediately before and after. If more than one search term is provided, user and channel are also matched at a lower priority. To specifically search within a channel, group, or DM, add in:channel_name, in:group_name, or in:username. To search for messages from a specific speaker, add from:username or from:botname. For IM results, the type is set to “im” and the channel.name property contains the user ID of the target user. For private group results, type is set to “group”. All search methods support the highlight parameter. If specified, the matching query terms will be marked up in the results so that clients may replace them with appropriate highlighting markers (e.g. <span class=”highlight”></span>). The UTF-8 markers we use are: start: “î”; # U+E000 (private-use) end : “î”; # U+E001 (private-use)

Please note that the max count value is 1000 and the max page value is 100.

For more information see https://api.slack.com/methods/messages

`__call__`(*query*, *count=None*, *highlight=None*, *page=None*, *sort=None*, *sort_dir=None*)

Searches for messages matching a query.

Parameters

- **`query`** – Required. Search query. May contains booleans, etc. e.g. pickleface
- **`count`** – Optional, default=20. Number of items to return per page. e.g. 20
- **`highlight`** – Optional. Pass a value of true to enable query highlight markers (see below). e.g. true
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2
- **`sort`** – Optional, default=score. Return matches sorted by either score or timestamp. e.g. timestamp
- **`sort_dir`** – Optional, default=desc. Change sort direction to ascending (asc) or descending (desc). e.g. asc

class `slackly.api.endpoints.stars.`**`StarsAdd`**(*bind=None*)

This method adds a star to an item (message, file, file comment, channel, private group, or DM) on behalf of the authenticated user. One of file, file_comment, channel, or the combination of channel and timestamp must be specified.

```
{
    "ok": true
}
```

After making this call, the item will be starred and a star_added event is broadcast through the RTM API for the calling user.

For more information see https://api.slack.com/methods/add

__call__(*channel=None*, *file=None*, *file_comment=None*, *timestamp=None*)

Adds a star to an item.

Parameters

- **channel** – Optional. Channel to add star to, or channel where the message to add star to was posted (used with timestamp). e.g. C1234567890
- **file** – Optional. File to add star to. e.g. F1234567890
- **file_comment** – Optional. File comment to add star to. e.g. Fc1234567890
- **timestamp** – Optional. Timestamp of the message to add star to. e.g. 1234567890.123456

class slackly.api.endpoints.stars.**StarsList**(*bind=None*)

This method lists the items starred by the authed user.

The response contains a list of starred items followed by pagination information.

```
{
    "ok": true,
    "items": [
        {
            "type": "message",
            "channel": "C2147483705",
            "message": {...}
        },
        {
            "type": "file",
            "file": { ... }
        }
        {
            "type": "file_comment",
            "file": { ... },
            "comment": { ... }
        }
        {
            "type": "channel",
            "channel": "C2147483705"
        },

    ],
    "paging": {
        "count": 100,
        "total": 4,
        "page": 1,
        "pages": 1
    }
}
```

Different item types can be starred. Every item in the list has a type property, the other property depend on the type of item. The possible types are:

message: the item will have a message property containing a message object file: this item will have a file property containing a file object. file_comment: the item will have a file property containing the file object and a comment property containing the file comment. channel: the item will have a channel property containing the channel ID. im: the item will have a channel property containing the channel ID for this direct message. group: the item will have a group property containing the channel ID for the private group.

The paging information contains the count of files returned, the total number of items starred, the page of results returned in this response and the total number of pages available. Please note that the max count value is 1000 and the max page value is 100.

For more information see https://api.slack.com/methods/list

`__call__`(*count=None*, *page=None*)
Lists stars for a user.

Parameters

- **`count`** – Optional, default=100. Number of items to return per page. e.g. 20
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2

class `slackly.api.endpoints.stars.`**`StarsRemove`**(*bind=None*)
This method removes a star from an item (message, file, file comment, channel, private group, or DM) on behalf of the authenticated user. One of file, file_comment, channel, or the combination of channel and timestamp must be specified.

```
{
    "ok": true
}
```

After making this call, the item will be unstarred and a star_removed event is broadcast through the RTM API for the calling user.

For more information see https://api.slack.com/methods/remove

`__call__`(*channel=None*, *file=None*, *file_comment=None*, *timestamp=None*)
Removes a star from an item.

Parameters

- **`channel`** – Optional. Channel to remove star from, or channel where the message to remove star from was posted (used with timestamp). e.g. C1234567890
- **`file`** – Optional. File to remove star from. e.g. F1234567890
- **`file_comment`** – Optional. File comment to remove star from. e.g. Fc1234567890
- **`timestamp`** – Optional. Timestamp of the message to remove star from. e.g. 1234567890.123456

class `slackly.api.endpoints.team.`**`ProfileGet`**(*bind=None*)
This method is used to get the profile field definitions for this team.

The response contains a profile item with an array of key value pairs. Right now only the fields key is supported, and it contains a list of field definitions for this team. Note that returned field definitions always have an id.

```
{
    "ok": true,
    "profile": {
        "fields": [
            {
                "id": "Xf06054AAA",
                "ordering": 0,
                "label": "Phone extension",
                "hint": "Enter the extension to reach your desk",
                "type": "text",
                "possible_values": null,
                "options": null,
                "is_hidden": 1
            },
            {
                "id": "Xf06054BBB",
                "ordering": 1,
                "label": "Date of birth",
                "hint": "When you were born",
                "type": "date",
                "possible_values": null,
                "options": null
            },
            {
                "id": "Xf06054CCC",
                "ordering": 2,
                "label": "Facebook",
                "hint": "Enter a link to your Facebook profile",
                "type": "link",
                "possible_values": null,
                "options": null
            },
            {
                "id": "Xf06054DDD",
                "ordering": 3,
                "label": "House",
                "hint": "Hogwarts, obviously",
                "type": "options_list",
                "possible_values": [
                    "Gryffindor",
                    "Hufflepuff",
                    "Ravenclaw",
                    "Slytherin",
                ],
                "options": null
            },
            {
                "id": "Xf06054EEE",
                "ordering": 4,
                "label": "Location",
                "hint": "Office location (LDAP)",
                "type": "text",
                "possible_values": null,
                "options": {
                    "is_protected": 1
                }
            },
            {
               "id": "Xf06054FFF",
                "ordering": 5,
                "label": "Manager",
                "hint": "The boss",
                "type": "user",
                "possible_values": null,
                "options": null
            }
        ]
    }
}
```

For more information see https://api.slack.com/methods/get

__call__(*visibility=None*)

Retrieve a team's profile.

Parameters **visibility** – Optional. Filter by visibility. e.g. all

class slackly.api.endpoints.team.**TeamAccessLogs**(*bind=None*)

This method is used to get the access logs for users on a team.

The response contains a list of logins followed by pagination information.

```
{
    "ok": true,
    "logins": [
        {
            "user_id": "U12345",
            "username": "bob",
            "date_first": 1422922864,
            "date_last": 1422922864,
            "count": 1,
            "ip": "127.0.0.1",
            "user_agent": "SlackWeb Mozilla\/5.0 (Macintosh; Intel Mac OS X 10_10_
↪2) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/41.0.2272.35 Safari\/537.36",
            "isp": "BigCo ISP",
            "country": "US",
            "region": "CA"
        },
        {
            "user_id": "U45678",
            "username": "alice",
            "date_first": 1422922493,
            "date_last": 1422922493,
            "count": 1,
            "ip": "127.0.0.1",
            "user_agent": "SlackWeb Mozilla\/5.0 (iPhone; CPU iPhone OS 8_1_3␣
↪like Mac OS X) AppleWebKit\/600.1.4 (KHTML, like Gecko) Version\/8.0 Mobile\/
↪12B466 Safari\/600.1.4",
            "isp": "BigCo ISP",
            "country": "US",
            "region": "CA"
        },
    ],
    "paging": {
        "count": 100,
        "total": 2,
        "page": 1,
        "pages": 1
    }
}
```

Each login contains the user id and username that logged in. date_first is a unix timestamp of the first login for this user/ip/user_agent combination. date_last is the most recent for that combination. count is the total number of logins for that combination. ip is the ip address of the device used to login. user_agent is the reported user agent string from the browser or client application. isp is our best guess at the internet service provider who owns the ip address. country and region are similarly where we think that login came from, based on the ip address. The paging information contains the count of items returned, the total number of items reacted to, the page of results returned in this response and the total number of pages available. Please note that the max count value is 1000 and the max page value is 100.

For more information see https://api.slack.com/methods/accessLogs

`__call__`(*before=None*, *count=None*, *page=None*)

Gets the access logs for the current team.

Parameters

- **`before`** – Optional, default=now. End of time range of logs to include in results (inclusive). e.g. 1457989166
- **`count`** – Optional, default=100. Number of items to return per page. e.g. 20
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2

class `slackly.api.endpoints.team.`**`TeamBillableInfo`**(*bind=None*)

This method lists billable information for each user on the team. Currently this consists solely of whether the user is subject to billing per Slack's Fair Billing policy.

The response contains a list of activity logs followed by pagination information.

```
{
    "ok": true,
    "billable_info": {
        "U0632EWRW": {
            "billing_active": false
        },
        "U02UCPE1R": {
            "billing_active": true
        },
        "U02UEBSD2": {
            "billing_active": true
        }
    }
}
```

A billing_active status of true indicates that the user is eligible per billing per Slack's Fair Billing policy. The billing_active status is computed periodically and the values returned by this API reflect the most recently computed status and should not be interpreted as a real-time view of each user's billing_active status.

For more information see https://api.slack.com/methods/billableInfo

`__call__`(*user=None*)

Gets billable users information for the current team.

Parameters **`user`** – Optional. A user to retrieve the billable information for. Defaults to all users. e.g. U1234567890

class `slackly.api.endpoints.team.`**`TeamInfo`**(*bind=None*)

This method provides information about your team.

```
{
    "ok": true,
    "team": {
        "id": "T12345",
        "name": "My Team",
        "domain": "example",
        "email_domain": "example.com",
        "icon": {
            "image_34": "https:\/\/...",
            "image_44": "https:\/\/...",
            "image_68": "https:\/\/...",
            "image_88": "https:\/\/...",
            "image_102": "https:\/\/...",
            "image_132": "https:\/\/...",
            "image_default": true
        },
       "enterprise_id": "E1234A12AB",
       "enterprise_name": "Umbrella Corporation"
    }
}
```

Team Icon If a team has not yet set a custom icon, the value of team.icon.image_default will be true. If the team belongs to an Enterprise Grid, the enterprise_id and enterprise_name fields will indicate the owning enterprise organization.

For more information see https://api.slack.com/methods/info

`__call__``()`
Gets information about the current team.

class `slackly.api.endpoints.team.`**`TeamIntegrationLogs`**`(`*bind=None*`)`
This method lists the integration activity logs for a team, including when integrations are added, modified and removed. This method can only be called by Admins.

The response contains a list of activity logs followed by pagination information.

```
{
    "ok": true,
    "logs": [
        {
            "service_id": "1234567890",
            "service_type": "Google Calendar",
            "user_id": "U1234ABCD",
            "user_name": "Johnny",
            "channel": "C1234567890",
            "date": "1392163200",
            "change_type": "enabled",
            "scope": "incoming-webhook"
        },
        {
            "app_id": "2345678901",
            "app_type": "Johnny App"
            "user_id": "U2345BCDE",
            "user_name": "Billy",
            "date": "1392163201",
            "change_type": "added"
            "scope": "chat:write:user,channels:read"
        },
        {
            "service_id": "3456789012",
            "service_type": "Airbrake",
            "user_id": "U3456CDEF",
            "user_name": "Joey",
            "channel": "C1234567890",
            "date": "1392163202",
            "change_type": "disabled",
            "reason": "user",
            "scope": "incoming-webhook"
        }
    ],
    "paging": {
            "count": 3,
            "total": 3,
            "page": 1,
            "pages": 1
    }
}
```

Logs can contain data for either a service or API application. If it's a service, service_id and service_type will be returned, and if it's an API application, app_id and app_type will be returned. Logs can also contain different change_types. The possible types are: added, removed, enabled, disabled, and updated. If the log entry is an RSS feed, the log will also contain rss_feed (with a value of true), rss_feed_change_type, rss_feed_title and rss_feed_url. When a disabled event is logged, its log entry will also contain a reason for why the event occurred. The list of possible reasons are:

user - Manually disabled by user rate_limits - Rate limits exceeded slack - Disabled by Slack errors - Too many errors system - A system change (i.e. channel archived) admin - Admin (i.e. user deleted) api_decline - User declied the API TOS deauth - Service deauthorized

The paging information contains the count of logs returned, the total number of logs, the page of results returned in this response and the total number of pages available.

For more information see https://api.slack.com/methods/integrationLogs

`__call__`(*app_id=None*, *change_type=None*, *count=None*, *page=None*, *service_id=None*, *user=None*)

Gets the integration logs for the current team.

Parameters

- **`app_id`** – Optional. Filter logs to this Slack app. Defaults to all logs. e.g.
- **`change_type`** – Optional. Filter logs with this change type. Defaults to all logs. e.g. added
- **`count`** – Optional, default=100. Number of items to return per page. e.g. 20
- **`page`** – Optional, default=1. Page number of results to return. e.g. 2
- **`service_id`** – Optional. Filter logs to this service. Defaults to all logs. e.g.
- **`user`** – Optional. Filter logs generated by this user's actions. Defaults to all logs. e.g. U1234567890

class `slackly.api.endpoints.usergroups.`**`UsergroupsCreate`**(*bind=None*)

This method is used to create a User Group.

If successful, the command returns a usergroup object, including preferences:

```
{
    "ok": true,
    "usergroup": {
        "id": "S0615G0KT",
        "team_id": "T060RNRCH",
        "is_usergroup": true,
        "name": "Marketing Team",
        "description": "Marketing gurus, PR experts and product advocates.",
        "handle": "marketing-team",
        "is_external": false,
        "date_create": 1446746793,
        "date_update": 1446746793,
        "date_delete": 0,
        "auto_type": null,
        "created_by": "U060RNRCZ",
        "updated_by": "U060RNRCZ",
        "deleted_by": null,
        "prefs": {
            "channels": [

            ],
            "groups": [

            ]
        },
        "user_count": "0"
    }
}
```

For more information see https://api.slack.com/methods/create

__call__(*name*, *channels=None*, *description=None*, *handle=None*, *include_count=None*)
Create a User Group

Parameters

- **name** – Required. A name for the User Group. Must be unique among User Groups. e.g. My Test Team
- **channels** – Optional. A comma separated string of encoded channel IDs for which the User Group uses as a default. e.g. C1234567890,C2345678901,C3456789012
- **description** – Optional. A short description of the User Group. e.g.
- **handle** – Optional. A mention handle. Must be unique among channels, users and User Groups. e.g. marketing
- **include_count** – Optional. Include the number of users in each User Group. e.g. true

class slackly.api.endpoints.usergroups.**UsergroupsDisable**(*bind=None*)
This method disables an existing User Group.

```
{
    "ok": true,
    "usergroup": {
        "id": "S0615G0KT",
        "team_id": "T060RNRCH",
        "is_usergroup": true,
        "name": "Marketing Team",
        "description": "Marketing gurus, PR experts and product advocates.",
        "handle": "marketing-team",
        "is_external": false,
        "date_create": 1446746793,
        "date_update": 1446747568,
        "date_delete": 1446747568,
        "auto_type": null,
        "created_by": "U060RNRCZ",
        "updated_by": "U060RNRCZ",
        "deleted_by": "U060RNRCZ",
        "prefs": {
            "channels": [

            ],
            "groups": [
            ]
        },
        "user_count": "0"
    }
}
```

When a User Group has been disabled its date_delete parameter will be non-zero.

For more information see https://api.slack.com/methods/disable

__call__(*usergroup*, *include_count=None*)

Disable an existing User Group

Parameters

- **usergroup** – Required. The encoded ID of the User Group to disable. e.g. S0604QSJC
- **include_count** – Optional. Include the number of users in the User Group. e.g. true

class slackly.api.endpoints.usergroups.**UsergroupsEnable**(*bind=None*)

This method enables a User Group which was previously disabled.

```
{
    "ok": true,
    "usergroup": {
        "id": "S0615G0KT",
        "team_id": "T060RNRCH",
        "is_usergroup": true,
        "name": "Marketing Team",
        "description": "Marketing gurus, PR experts and product advocates.",
        "handle": "marketing-team",
        "is_external": false,
        "date_create": 1446746793,
        "date_update": 1446747767,
        "date_delete": 0,
        "auto_type": null,
        "created_by": "U060RNRCZ",
        "updated_by": "U060RNRCZ",
        "deleted_by": null,
        "prefs": {
            "channels": [

            ],
            "groups": [

            ]
        },
        "user_count": "0"
    }
}
```

When a User Group is enabled, it's date_delete parameter will be 0 (zero).

For more information see https://api.slack.com/methods/enable

__call__(*usergroup*, *include_count=None*)

Enable a User Group

Parameters

- **usergroup** – Required. The encoded ID of the User Group to enable. e.g. S0604QSJC

- **include_count** – Optional. Include the number of users in the User Group. e.g. true

class slackly.api.endpoints.usergroups.**UsergroupsList**(*bind=None*)

This method returns a list of all User Groups in the team. This can optionally include disabled User Groups.

Returns a list of usergroup objects, in no particular order:

```
{
    "ok": true,
    "usergroups": [
        {
            "id": "S0614TZR7",
            "team_id": "T060RNRCH",
            "is_usergroup": true,
            "name": "Team Admins",
            "description": "A group of all Administrators on your team.",
            "handle": "admins",
            "is_external": false,
            "date_create": 1446598059,
            "date_update": 1446670362,
            "date_delete": 0,
            "auto_type": "admin",
            "created_by": "USLACKBOT",
            "updated_by": "U060RNRCZ",
            "deleted_by": null,
            "prefs": {
                "channels": [

                ],
                "groups": [

                ]
            },
            "user_count": "2"
        },
        {
            "id": "S06158AV7",
            "team_id": "T060RNRCH",
            "is_usergroup": true,
            "name": "Team Owners",
            "description": "A group of all Owners on your team.",
            "handle": "owners",
            "is_external": false,
            "date_create": 1446678371,
            "date_update": 1446678371,
            "date_delete": 0,
            "auto_type": "owner",
            "created_by": "USLACKBOT",
            "updated_by": "USLACKBOT",
            "deleted_by": null,
            "prefs": {
                "channels": [

                ],
                "groups": [

                ]
            },
            "user_count": "1"
        },
        {
            "id": "S0615G0KT",
            "team_id": "T060RNRCH",
            "is_usergroup": true,
            "name": "Marketing Team",
            "description": "Marketing gurus, PR experts and product advocates.",
            "handle": "marketing-team",
            "is_external": false,
            "date_create": 1446746793,
            "date_update": 1446747767,
            "date_delete": 1446748865,
            "auto_type": null,
            "created_by": "U060RNRCZ",
            "updated_by": "U060RNRCZ",
            "deleted_by": null,
            "prefs": {
                "channels": [

                ],
                "groups": [

                ]
            },
            "user_count": "0"
        }
    ]
}
```

For more information see https://api.slack.com/methods/list

`__call__`(*include_count=None*, *include_disabled=None*, *include_users=None*)

List all User Groups for a team

Parameters

- **`include_count`** – Optional. Include the number of users in each User Group. e.g. true
- **`include_disabled`** – Optional. Include disabled User Groups. e.g. true
- **`include_users`** – Optional. Include the list of users for each User Group. e.g. true

class `slackly.api.endpoints.usergroups.`**`UsergroupsUpdate`**(*bind=None*)

This method updates the properties of an existing User Group.

```
{
    "ok": true,
    "usergroup": {
        "id": "S0615G0KT",
        "team_id": "T060RNRCH",
        "is_usergroup": true,
        "name": "Marketing Gurus",
        "description": "Marketing gurus, PR experts and product advocates.",
        "handle": "marketing-team",
        "is_external": false,
        "date_create": 1446746793,
        "date_update": 1446748574,
        "date_delete": 0,
        "auto_type": null,
        "created_by": "U060RNRCZ",
        "updated_by": "U060RNRCZ",
        "deleted_by": null,
        "prefs": {
            "channels": [

            ],
            "groups": [

            ]
        },
        "user_count": "0"
    }
}
```

For more information see https://api.slack.com/methods/update

__call__(*usergroup*, *channels=None*, *description=None*, *handle=None*, *include_count=None*, *name=None*)

Update an existing User Group

Parameters

- **usergroup** – Required. The encoded ID of the User Group to update. e.g. S0604QSJC
- **channels** – Optional. A comma separated string of encoded channel IDs for which the User Group uses as a default. e.g. C1234567890,C2345678901,C3456789012
- **description** – Optional. A short description of the User Group. e.g.
- **handle** – Optional. A mention handle. Must be unique among channels, users and User Groups. e.g. marketing
- **include_count** – Optional. Include the number of users in the User Group. e.g. true
- **name** – Optional. A name for the User Group. Must be unique among User Groups. e.g. My Test Team

class `slackly.api.endpoints.usergroups.`**UsersList**(*bind=None*)

This method returns a list of all users within a User Group.

```
{
    "ok": true,
    "users": [
        "U060R4BJ4"
    ]
}
```

For more information see https://api.slack.com/methods/list

__call__(*usergroup*, *include_disabled=None*)

List all users in a User Group

Parameters

- **usergroup** – Required. The encoded ID of the User Group to update. e.g. S0604QSJC
- **include_disabled** – Optional. Allow results that involve disabled User Groups. e.g. true

class `slackly.api.endpoints.usergroups.`**UsersUpdate**(*bind=None*)

This method updates the list of users that belong to a User Group. This method replaces all users in a User Group with the list of users provided in the users parameter.

```
{
    "ok": true,
    "usergroup": {
        "id": "S0616NG6M",
        "team_id": "T060R4BHN",
        "is_usergroup": true,
        "name": "Marketing Team",
        "description": "Marketing gurus, PR experts and product advocates.",
        "handle": "marketing-team",
        "is_external": false,
        "date_create": 1447096577,
        "date_update": 1447102109,
        "date_delete": 0,
        "auto_type": null,
        "created_by": "U060R4BJ4",
        "updated_by": "U060R4BJ4",
        "deleted_by": null,
        "prefs": {
            "channels": [

            ],
            "groups": [

            ]
        },
        "users": [
            "U060R4BJ4",
            "U060RNRCZ"
        ],
        "user_count": 1
    }
}
```

For more information see https://api.slack.com/methods/update

__call__(*usergroup*, *users*, *include_count=None*)

Update the list of users for a User Group

Parameters

- **usergroup** – Required. The encoded ID of the User Group to update. e.g. S0604QSJC
- **users** – Required. A comma separated string of encoded user IDs that represent the entire list of users for the User Group. e.g. U060R4BJ4,U060RNRCZ
- **include_count** – Optional. Include the number of users in the User Group. e.g. true

class slackly.api.endpoints.users.**ProfileGet**(*bind=None*)

Use this method to retrieve a user's profile information.

The response contains a profile item with an array of key:value pairs. We hope you find the first_name, last_name, email attributes self-explanatory. The user's custom-set "current status" can be found in the status_text and status_emoji attributes. See custom status for more. The image_ keys hold links to the different sizes we support for the user's profile image from 24x24 to 1024x1024 pixels. A link to the image in its original size is stored in image_original. Bot users may contain an always_active profile field, indicating whether the bot user is active in a way that overrides traditional presence rules. The presence docs tell the whole story. For a description of the fields key, see the users.profile.set method.

```
{
    "ok": true,
    "profile": {
        "status_text": "riding a train",
        "status_emoji": ":mountain_railway:",
        "first_name": "John",
        "last_name": "Smith",
        "email": "john@smith.com",
        "skype": "johnsmith",
        "image_24": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_24.jpg",
        "image_32": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_32.jpg",
        "image_48": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_48.jpg",
        "image_72": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_72.jpg",
        "image_192": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_192.jpg",
        "image_512": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_512.jpg",
        "image_1024": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_1024.jpg",
        "image_original": "https://s3.amazonaws.com/slack-files/avatars/2015-11-
↪16/123456_original.jpg",
        "fields": {
            "Xf06054AAA": {
                "value": "San Francisco",
                "alt": "Giants, yo!",
                "label": "Favorite Baseball Team"
            },
            "Xf06054BBB": {
                "value": "Barista",
                "alt": "I make the coffee & the tea!",
                "label": "Position"
            }
        }
    }
}
```

For more information see https://api.slack.com/methods/get

__call__(*include_labels=None*, *user=None*)

Retrieves a user's profile information.

Parameters

- **include_labels** – Optional, default=false. Include labels for each ID in custom profile fields e.g. true
- **user** – Optional. User to retrieve profile info for e.g. U1234567890

class slackly.api.endpoints.users.**ProfileSet**(*bind=None*)

Use this method to set a user's profile information, including name, email, current status, and other attributes.

After making this call, the complete user's profile will be returned in the same format as above.

```
{
    "ok": true,
    "profile": {
        "avatar_hash": "ge3b51ca72ze",
        "status_text": "riding a train",
        "status_emoji": ":mountain_railway:",
        "first_name": "John",
        "last_name": "Smith",
        "email": "john@smith.com",
        "skype": "johnsmith",
        "image_24": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_24.jpg",
        "image_32": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_32.jpg",
        "image_48": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_48.jpg",
        "image_72": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_72.jpg",
        "image_192": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_192.jpg",
        "image_512": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_512.jpg",
        "image_1024": "https://s3.amazonaws.com/slack-files/avatars/2015-11-16/
↪123456_1024.jpg",
        "image_original": "https://s3.amazonaws.com/slack-files/avatars/2015-11-
↪16/123456_original.jpg",
        "fields": {
            "Xf06054AAA": {
                "value": "San Francisco",
                "alt": "Giants, yo!",
                "label": "Favorite Baseball Team"
            },
            "Xf06054BBB": {
                "value": "Barista",
                "alt": "I make the coffee & the tea!",
                "label": "Position"
            }
        }
    }
}
```

This method will generate a user_change event on success, containing the complete user.

For more information see https://api.slack.com/methods/set

__call__(*name=None*, *profile=None*, *user=None*, *value=None*)
Set the profile information for a user.

Parameters

- **name** – Optional. Name of a single key to set. Usable only if profile is not passed. e.g. first_name
- **profile** – Optional. Collection of key:value pairs presented as a URL-encoded JSON hash. e.g. { first_name: “John”, ... }
- **user** – Optional. ID of user to change. This argument may only be specified by team admins on paid teams. e.g. U1234567890
- **value** – Optional. Value to set a single key to. Usable only if profile is not passed. e.g. John

class slackly.api.endpoints.users.**UsersDeletePhoto**(*bind=None*)
This method allows the user to delete their profile image. It will clear whatever image is currently set.

To upload a new profile image, use the companion method users.setPhoto.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/deletePhoto

__call__()
Delete the user profile photo

class slackly.api.endpoints.users.**UsersGetPresence**(*bind=None*)
This method lets you find out information about a user's presence. Consult the presence documentation for more details.

When requesting information for a different user, this method just returns the current presence (either active or away):

```
{
    "ok": true,
    "presence": "active"
}
```

If you are requesting presence information for the authed user, this method returns the current presence, along with details on how it was calculated:

```
{
    "ok": true,
    "presence": "active",
    "online": true,
    "auto_away": false,
    "manual_away": false,
    "connection_count": 1,
    "last_activity": 1419027078
}
```

presence indicates the user's overall presence, it will either be active or away. online will be true if they have a client currently connected to Slack. auto_away will be true if our servers haven't detected any activity from the user in the last 30 minutes. manual_away will be true if the user has manually set their presence to away. connection_count gives a count of total connections. last_activity indicates the last activity seen by our servers. If a user has no connected clients then this property will be absent.

For more information see https://api.slack.com/methods/getPresence

__call__(*user*)
Gets user presence information.

Parameters **user** – Required. User to get presence info on. Defaults to the authed user. e.g. U1234567890

class slackly.api.endpoints.users.**UsersIdentity**(*bind=None*)
After your Slack app is awarded an identity token through Sign in with Slack, use this method to retrieve a user's identity.

The returned fields depend on any additional authorization scopes you've requested.

This method may only be used by tokens with the identity.basic scope, as provided in the Sign in with Slack process.

You will receive at a minimum the following information:

```
{
    "ok": true,
    "user": {
        "name": "Sonny Whether",
        "id": "U0G9QF9C6"
    },
    "team": {
      "id": "T0G9PQBBK"
    }
}
```

Notice that user IDs are not guaranteed to be globally unique across all Slack users. The combination of user ID and team ID, on the other hand, is guaranteed to be globally unique. See the Sign in with Slack docs for even more information on these responses.

Authorization scopes In addition, you can request access to additional profile fields by adding the following authorization scopes to your OAuth request: identity.email provides the team member's email address, if available:

```
{
    "ok": true,
    "user": {
        "name": "Sonny Whether",
        "id": "U0G9QF9C6",
        "email": "bobby@example.com"
    },
    "team": {
      "id": "T0G9PQBBK"
    }
}
```

identity.avatar yield the team member's avatar images. Available sizes may vary in the future.

```
{
    "ok": true,
    "user": {
        "name": "Sonny Whether",
        "id": "U0G9QF9C6",
        "image_24": "https://cdn.example.com/sonny_24.jpg",
        "image_32": "https://cdn.example.com/sonny_32.jpg",
        "image_48": "https://cdn.example.com/sonny_48.jpg",
        "image_72": "https://cdn.example.com/sonny_72.jpg",
        "image_192": "https://cdn.example.com/sonny_192.jpg"
    },
    "team": {
        "id": "T0G9PQBBK"
    }
}
```

Use identity.team to retrieve the user's team name:

```
{
    "ok": true,
    "user": {
        "name": "Sonny Whether",
        "id": "U0G9QF9C6"
    },
    "team": {
        "name": "Captain Fabian's Naval Supply",
        "id": "T0G9PQBBK"
    }
}
```

For more information see https://api.slack.com/methods/identity

__call__()

Get a user's identity.

class slackly.api.endpoints.users.**UsersInfo**(*bind=None*)

This method returns information about a team member.

Returns a user object:

```
{
    "ok": true,
    "user": {
        "id": "U023BECGF",
        "name": "bobby",
        "deleted": false,
        "color": "9f69e7",
        "profile": {
            "avatar_hash": "ge3b51ca72de",
            "current_status": ":mountain_railway: riding a train",
            "first_name": "Bobby",
            "last_name": "Tables",
            "real_name": "Bobby Tables",
            "email": "bobby@slack.com",
            "skype": "my-skype-name",
            "phone": "+1 (123) 456 7890",
            "image_24": "https:\/\/...",
            "image_32": "https:\/\/...",
            "image_48": "https:\/\/...",
            "image_72": "https:\/\/...",
            "image_192": "https:\/\/..."
        },
        "is_admin": true,
        "is_owner": true,
        "updated": 1490054400,
        "has_2fa": true
    }
}
```

Profile The profile hash contains as much information as the user has supplied in the default profile fields: first_name, last_name, real_name, email, skype, and the image_* fields. Only the image_* fields are guaranteed to be included. Data that has not been supplied may not be present at all, may be null or may contain the empty string (“”). A user's custom profile fields may be discovered using users.profile.get.

Email addresses Apps created after January 4th, 2017 must request both the users:read and users:read.email OAuth permission scopes when using the OAuth app installation flow to enable access to the email field of user objects returned by this method.

For more information see https://api.slack.com/methods/info

__call__(*user*)

Gets information about a user.

Parameters `user` – Required. User to get info on e.g. U1234567890

class `slackly.api.endpoints.users.`**`UsersList`**(*bind=None*)

This method returns a list of all users in the team. This includes deleted/deactivated users.

Returns a list of user objects, in no particular order:

```
{
    "ok": true,
    "members": [
        {
            "id": "U023BECGF",
            "team_id": "T021F9ZE2",
            "name": "bobby",
            "deleted": false,
            "color": "9f69e7",
            "real_name": "Bobby Tables",
            "tz": "America\/Los_Angeles",
            "tz_label": "Pacific Daylight Time",
            "tz_offset": -25200,
            "profile": {
                "avatar_hash": "ge3b51ca72de",
                "current_status": ":mountain_railway: riding a train",
                "first_name": "Bobby",
                "last_name": "Tables",
                "real_name": "Bobby Tables",
                "email": "bobby@slack.com",
                "skype": "my-skype-name",
                "phone": "+1 (123) 456 7890",
                "image_24": "https:\/\/...",
                "image_32": "https:\/\/...",
                "image_48": "https:\/\/...",
                "image_72": "https:\/\/...",
                "image_192": "https:\/\/..."
            },
            "is_admin": true,
            "is_owner": true,
            "updated": 1490054400,
            "has_2fa": false
        },
        ...
    ]
}
```

Profile The profile hash contains as much information as the user has supplied in the default profile fields: first_name, last_name, real_name, email, skype, and the image_* fields. Only the image_* fields are guaranteed to be included. Data that has not been supplied may not be present at all, may be null or may contain the empty string (“”). A user’s custom profile fields may be discovered using users.profile.get.

Email addresses Apps created after January 4th, 2017 must request both the users:read and users:read.email OAuth permission scopes when using the OAuth app installation flow to enable access to the email field of user objects returned by this method.

For more information see https://api.slack.com/methods/list

`__call__`(*presence=None*)

Lists all users in a Slack team.

Parameters **`presence`** – Optional. Whether to include presence data in the output e.g. true

class `slackly.api.endpoints.users.`**`UsersSetActive`**(*bind=None*)

This method lets the slack messaging server know that the authenticated user is currently active. Consult the presence documentation for more details.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/setActive

`__call__`()
Marks a user as active.

class `slackly.api.endpoints.users.`**`UsersSetPhoto`**(*bind=None*)
This method allows the user to set their profile image. The caller can pass image data via image.

Providing a “crop box” with crop_x, crop_y, and crop_w is optional. Otherwise, the whole image will be used. If cropping instructions are not specified and the source image is not square, the image will be letterboxed, just like your favorite old laserdiscs.

Please limit your images to a maximum size of 1024 by 1024 pixels. 512x512 pixels is the minimum.

To remove a profile image, use the companion method users.deletePhoto.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/setPhoto

`__call__`(*image*, *crop_w=None*, *crop_x=None*, *crop_y=None*)
Set the user profile photo

Parameters

- **`image`** – Required. File contents via multipart/form-data. e.g. ...
- **`crop_w`** – Optional. Width/height of crop box (always square) e.g. 100
- **`crop_x`** – Optional. X coordinate of top-left corner of crop box e.g. 10
- **`crop_y`** – Optional. Y coordinate of top-left corner of crop box e.g. 15

class `slackly.api.endpoints.users.`**`UsersSetPresence`**(*bind=None*)
This method lets you set the calling user’s manual presence. Consult the presence documentation for more details.

```
{
    "ok": true
}
```

For more information see https://api.slack.com/methods/setPresence

`__call__`(*presence*)
Manually sets user presence.

Parameters **`presence`** – Required. Either auto or away e.g. away

Slack Types

This package defines every type that Slackly defines. They are defined in a variety of sub modules for ease of discoverability and collaboration, but all types should be exposed in this top level package.

Base Classes and Utilities

class `slackly.schema.types._base.`**`SlackType`**(*kwargs*)

The base SlackType. It handles the default creation of attributes from keyword arguments

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types._base.`**`HasIDMixin`**

This mixin adds the behavior of allowing the object to be instantiated from an ID. Also updates the __repr__ to include the id in the repr

Types

class `slackly.schema.types._base.`**`HasIDMixin`**

This mixin adds the behavior of allowing the object to be instantiated from an ID. Also updates the __repr__ to include the id in the repr

classmethod **`Id`**(*id*)

Class method to create the class from an ID

Parameters **`id`** – A slack ID, e.g. T1NASNDSP

Returns A class instance of whatever class this is mixed in to

`__weakref__`

list of weak references to the object (if defined)

class `slackly.schema.types._base.`**`SlackType`**(*kwargs*)

The base SlackType. It handles the default creation of attributes from keyword arguments

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

`__init__`(*kwargs*)

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

`__weakref__`

list of weak references to the object (if defined)

`as_dict`

Returns a deepcopy of the object as a dictionary

Returns A `dict`

`as_json`

Returns a utf-8 string of the json dump of the dictionary

Returns A `str`

`slackly.schema.types.basic.`**`Bool`**(*value*)

A function to convert a value into a Boolean

Parameters **`value`** – any value that can be passed to bool

Returns a `bool` or None

`slackly.schema.types.basic.`**`Epoch`**(*value*)

A Function to take an epoch and return a datetime.datetime object. Also defaults that 0 is None

Parameters **value** – An epoch

Returns a `datetime.datetime` or None

`slackly.schema.types.basic.`**`Integer`**(*value*)

A function to convert a value into a string

Parameters **value** – any intable value

Returns a `int` or None

class `slackly.schema.types.basic.`**`List`**(*slack_type*)

A collection. It gets instantiated with a type factory, and when it's called, it iterates through its argument and returns a `list` of whatever type it was instantiated with.

Parameters **slack_type** – A `SlackType`

`__init__`(*slack_type*)

Parameters **slack_type** – A `SlackType`

class `slackly.schema.types.basic.`**`SlackTimestamp`**(*timestamp*)

A type to wrap slack timestamps. These represent actual times, but also can be considered uuids in the context of a team.

Parameters **timestamp** – A `str` representing a timestamp

`__init__`(*timestamp*)

Parameters **timestamp** – A `str` representing a timestamp

`dt`

Returns the `SlackTimeStamp` as a `datetime.datetime`

Returns A `datetime.datetime`

`uuid`

Returns A `str` representing a a timestampe

`slackly.schema.types.basic.`**`String`**(*value*)

A function to convert a value into a string

Parameters **value** – any stringifyable value

Returns a `str` or None

class `slackly.schema.types.basic.`**`Timestamp`**(*kwargs*)

An actual timestamp (not necessarily a uuid

Parameters **kwargs** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.message.`**`Message`**(*kwargs*)

A Message

Parameters **kwargs** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.message.`**`PostedMeMessage`**(*kwargs*)

A posted meMessage (message posted as a user)

Parameters **kwargs** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.message.`**`PostedMessage`**(*kwargs*)
A Posted Message

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.message.`**`UpdatedMessage`**(*kwargs*)
An updated message

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`Channel`**(*kwargs*)
A Channel

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`ChannelPurpose`**(*kwargs*)
A Channel Purpose

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`ChannelTopic`**(*kwargs*)
A Channel Topic

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`ChannelType`**(*kwargs*)
A Channel Type

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`CreatedChannel`**(*kwargs*)
A Created Channel

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`IM`**(*kwargs*)
An Single Person Direct Message Channel

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

class `slackly.schema.types.channel.`**`MPIM`**(*kwargs*)
An Multi-Person Direct Message Channel

Parameters **`kwargs`** – An arbitrary dictionary of keyword arguments

Returns None

Slack Events

This package contains all events

Base Classes and Utilities

class `slackly.events._base.`**`BaseEvent`**

The base event

classmethod **`parse`**(*event*)

Given an event dictionary, gets the event schema and applies it to the dictionary, otherwise just returns the *`BaseEvent`* from the dict.

Parameters **`event`** – A `dict` representing an event

Returns A *`BaseEvent`* subclass

`slackly.events._base.`**`register_event`**(*event_type*)

This registered an event in the global event registry

Parameters **`event_type`** – A `str` representing an event type

Returns A callable that will register an event class as the given event type

Events

class `slackly.events.accounts_changed.`**`AccountsChanged`**

Listing 5.1: Example json response

```
{
    "type": "accounts_changed"
}
```

For more information see https://api.slack.com/events/accounts_changed

class `slackly.events.bot_added.`**`BotAdded`**

Listing 5.2: Example json response

```
{
    "type": "bot_added",
    "bot": {
        "id": "B024BE7LH",
        "app_id": "A4H1JB4AZ",
        "name": "hugbot",
        "icons": {
            "image_48": "https:\/\/slack.com\/path\/to\/hugbot_48.png"
        }
    }
}
```

For more information see https://api.slack.com/events/bot_added

class `slackly.events.accounts_changed.`**`AccountsChanged`**

Listing 5.3: Example json response

```
{
    "type": "accounts_changed"
}
```

For more information see https://api.slack.com/events/accounts_changed

class `slackly.events.bot_added.`**`BotAdded`**

Listing 5.4: Example json response

```
{
    "type": "bot_added",
    "bot": {
        "id": "B024BE7LH",
        "app_id": "A4H1JB4AZ",
        "name": "hugbot",
        "icons": {
            "image_48": "https:\/\/slack.com\/path\/to\/hugbot_48.png"
        }
    }
}
```

For more information see https://api.slack.com/events/bot_added

class `slackly.events.bot_changed.`**`BotChanged`**

Listing 5.5: Example json response

```
{
    "type": "bot_changed",
    "bot": {
        "id": "B024BE7LH",
        "app_id": "A4H1JB4AZ",
        "name": "hugbot",
        "icons": {
            "image_48": "https:\/\/slack.com\/path\/to\/hugbot_48.png"
        }
    }
}
```

For more information see https://api.slack.com/events/bot_changed

class `slackly.events.channel_archive.`**`ChannelArchive`**

Listing 5.6: Example json response

```
{
    "type": "channel_archive",
    "channel": "C024BE91L",
    "user": "U024BE7LH"
}
```

For more information see https://api.slack.com/events/channel_archive

class `slackly.events.channel_created.`**`ChannelCreated`**

Listing 5.7: Example json response

```
{
    "type": "channel_created",
    "channel": {
        "id": "C024BE91L",
        "name": "fun",
        "created": 1360782804,
        "creator": "U024BE7LH"
    }
}
```

For more information see https://api.slack.com/events/channel_created

class `slackly.events.channel_deleted.`**`ChannelDeleted`**

Listing 5.8: Example json response

```
{
    "type": "channel_deleted",
    "channel": "C024BE91L"
}
```

For more information see https://api.slack.com/events/channel_deleted

class `slackly.events.channel_history_changed.`**`ChannelHistoryChanged`**

Listing 5.9: Example json response

```
{
    "type": "channel_history_changed",
    "latest": "1358877455.000010",
    "ts": "1361482916.000003",
    "event_ts": "1361482916.000004"
}
```

For more information see https://api.slack.com/events/channel_history_changed

class `slackly.events.channel_joined.`**`ChannelJoined`**

Listing 5.10: Example json response

```
{
    "type": "channel_joined",
    "channel": {
         ...
    }
}
```

For more information see https://api.slack.com/events/channel_joined

class `slackly.events.channel_left.`**`ChannelLeft`**

Listing 5.11: Example json response

```
{
    "type": "channel_left",
    "channel": "C024BE91L"
}
```

For more information see https://api.slack.com/events/channel_left

class slackly.events.channel_marked.**ChannelMarked**

Listing 5.12: Example json response

```
{
    "type": "channel_marked",
    "channel": "C024BE91L",
    "ts": "1401383885.000061"
}
```

For more information see https://api.slack.com/events/channel_marked

class slackly.events.channel_rename.**ChannelRename**

Listing 5.13: Example json response

```
{
    "type": "channel_rename",
    "channel": {
        "id":"C02ELGNBH",
        "name":"new_name",
        "created":1360782804
    }
}
```

For more information see https://api.slack.com/events/channel_rename

class slackly.events.channel_unarchive.**ChannelUnarchive**

Listing 5.14: Example json response

```
{
    "type": "channel_unarchive",
    "channel": "C024BE91L",
    "user": "U024BE7LH"
}
```

For more information see https://api.slack.com/events/channel_unarchive

class slackly.events.commands_changed.**CommandsChanged**

Listing 5.15: Example json response

```
{
    "type": "commands_changed",
    "event_ts" : "1361482916.000004"
}
```

For more information see https://api.slack.com/events/commands_changed

class slackly.events.desktop_notification.**DesktopNotification**

Listing 5.16: Example json response

```
{
    "type": "desktop_notification",
    "title": "FoolishBotDev",
    "subtitle": "#new_channel",
    "msg": "1489584201.102512",
    "content": "hunter: hi @basebot",
    "channel": "C1QCTSZAM",
    "launchUri": "slack://channel?id=C1QCTSZAM&message=1489584201102512&
↪team=T1NASNDSP",
    "avatarImage": "https://secure.gravatar.com/avatar/
↪13affcd29fcf0d3e861f5ca2c8eedc44.jpg?s=192&d=https%3A%2F%2Fa.slack-edge.
↪com%2F7fa9%2Fimg%2Favatars%2Fava_0019-192.png",
    "ssbFilename": "knock_brush.mp3",
    "imageUri": null,
    "is_shared": false,
    "event_ts": "1489584201.895898"
}
```

This event tye is undocumented

class `slackly.events.dnd_updated.`**`DndUpdated`**

Listing 5.17: Example json response

```
{
    "type": "dnd_updated",
    "user": "U1234",
    "dnd_status": {
        "dnd_enabled": true,
        "next_dnd_start_ts": 1450387800,
        "next_dnd_end_ts": 1450423800,
        "snooze_enabled": true,
        "snooze_endtime": 1450373897
    }
}
```

For more information see https://api.slack.com/events/dnd_updated

class `slackly.events.dnd_updated_user.`**`DndUpdatedUser`**

Listing 5.18: Example json response

```
{
    "type": "dnd_updated_user",
    "user": "U1234",
    "dnd_status": {
        "dnd_enabled": true,
        "next_dnd_start_ts": 1450387800,
        "next_dnd_end_ts": 1450423800
    }
}
```

For more information see https://api.slack.com/events/dnd_updated_user

class `slackly.events.email_domain_changed.`**`EmailDomainChanged`**

Listing 5.19: Example json response

```
{
    "type": "email_domain_changed",
    "email_domain":"example.com",
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/email_domain_changed

class slackly.events.emoji_changed.**EmojiChanged**

Listing 5.20: Example json response

```
{
    "type": "emoji_changed",
    "subtype": "remove",
    "names": ["picard_facepalm"],
    "event_ts" : "1361482916.000004"
}
```

For more information see https://api.slack.com/events/emoji_changed

class slackly.events.file_change.**FileChange**

Listing 5.21: Example json response

```
{
    "type": "file_change",
    "file": {  ...  }
}
```

For more information see https://api.slack.com/events/file_change

class slackly.events.file_comment_added.**FileCommentAdded**

Listing 5.22: Example json response

```
{
    "type": "file_comment_added",
    "file": {  ...  },
    "comment": {  ...  }
}
```

For more information see https://api.slack.com/events/file_comment_added

class slackly.events.file_comment_deleted.**FileCommentDeleted**

Listing 5.23: Example json response

```
{
    "type": "file_comment_deleted",
    "file": {  ...  },
    "comment": "Fc67890"
}
```

For more information see https://api.slack.com/events/file_comment_deleted

class slackly.events.file_comment_edited.**FileCommentEdited**

Listing 5.24: Example json response

```
{
    "type": "file_comment_edited",
    "file": {  ...  },
    "comment": {  ...  }
}
```

For more information see https://api.slack.com/events/file_comment_edited

class `slackly.events.file_created.`**`FileCreated`**

Listing 5.25: Example json response

```
{
    "type": "file_created",
    "file": {  ...  }
}
```

For more information see https://api.slack.com/events/file_created

class `slackly.events.file_deleted.`**`FileDeleted`**

Listing 5.26: Example json response

```
{
    "type": "file_deleted",
    "file_id": "F2147483862",
    "event_ts": "1361482916.000004"
}
```

For more information see https://api.slack.com/events/file_deleted

class `slackly.events.file_public.`**`FilePublic`**

Listing 5.27: Example json response

```
{
    "type": "file_public",
    "file_id": "F4LDY26EL",
    "user_id": "USLACKBOT",
    "file": {  ...  }
    "event_ts": "1490062255.798583",
}
```

For more information see https://api.slack.com/events/file_public

class `slackly.events.file_shared.`**`FileShared`**

Listing 5.28: Example json response

```
{
    "type": "file_shared",
    "file": {  ...  }
}
```

For more information see https://api.slack.com/events/file_shared

class `slackly.events.file_unshared.`**`FileUnshared`**

Listing 5.29: Example json response

```
{
    "type": "file_unshared",
    "file": {  ...  }
}
```

For more information see https://api.slack.com/events/file_unshared

class `slackly.events.goodbye.`**`Goodbye`**

Listing 5.30: Example json response

```
{
    "type": "goodbye"
}
```

For more information see https://api.slack.com/events/goodbye

class `slackly.events.group_archive.`**`GroupArchive`**

Listing 5.31: Example json response

```
{
    "type": "group_archive",
    "channel": "G024BE91L"
}
```

For more information see https://api.slack.com/events/group_archive

class `slackly.events.group_close.`**`GroupClose`**

Listing 5.32: Example json response

```
{
    "type": "group_close",
    "user": "U024BE7LH",
    "channel": "G024BE91L"
}
```

For more information see https://api.slack.com/events/group_close

class `slackly.events.group_history_changed.`**`GroupHistoryChanged`**

Listing 5.33: Example json response

```
{
    "type": "group_history_changed",
    "latest": "1358877455.000010",
    "ts": "1361482916.000003",
    "event_ts": "1361482916.000004"
}
```

For more information see https://api.slack.com/events/group_history_changed

class `slackly.events.group_joined.`**`GroupJoined`**

Listing 5.34: Example json response

```
{
    "type": "group_joined",
    "channel": {
        ...
    }
}
```

For more information see https://api.slack.com/events/group_joined

class slackly.events.group_left.**GroupLeft**

Listing 5.35: Example json response

```
{
    "type": "group_left",
    "channel": "G02ELGNBH"
}
```

For more information see https://api.slack.com/events/group_left

class slackly.events.group_marked.**GroupMarked**

Listing 5.36: Example json response

```
{
    "type": "group_marked",
    "channel": "G024BE91L",
    "ts": "1401383885.000061"
}
```

For more information see https://api.slack.com/events/group_marked

class slackly.events.group_open.**GroupOpen**

Listing 5.37: Example json response

```
{
    "type": "group_open",
    "user": "U024BE7LH",
    "channel": "G024BE91L"
}
```

For more information see https://api.slack.com/events/group_open

class slackly.events.group_rename.**GroupRename**

Listing 5.38: Example json response

```
{
    "type": "group_rename",
    "channel": {
        "id":"G02ELGNBH",
        "name":"new_name",
        "created":1360782804
    }
}
```

For more information see https://api.slack.com/events/group_rename

class slackly.events.group_unarchive.**GroupUnarchive**

Listing 5.39: Example json response

```
{
    "type": "group_unarchive",
    "channel": "G024BE91L"
}
```

For more information see https://api.slack.com/events/group_unarchive

class slackly.events.hello.**Hello**

Listing 5.40: Example json response

```
{
    "type": "hello"
}
```

For more information see https://api.slack.com/events/hello

class slackly.events.im_close.**ImClose**

Listing 5.41: Example json response

```
{
    "type": "im_close",
    "user": "U024BE7LH",
    "channel": "D024BE91L"
}
```

For more information see https://api.slack.com/events/im_close

class slackly.events.im_created.**ImCreated**

Listing 5.42: Example json response

```
{
    "type": "im_created",
    "user": "U024BE7LH",
    "channel": {...}
}
```

For more information see https://api.slack.com/events/im_created

class slackly.events.im_history_changed.**ImHistoryChanged**

Listing 5.43: Example json response

```
{
    "type": "im_history_changed",
    "latest": "1358877455.000010",
    "ts": "1361482916.000003",
    "event_ts": "1361482916.000004"
}
```

For more information see https://api.slack.com/events/im_history_changed

class slackly.events.im_marked.**ImMarked**

Listing 5.44: Example json response

```
{
    "type": "im_marked",
    "channel": "D024BE91L",
    "ts": "1401383885.000061"
}
```

For more information see https://api.slack.com/events/im_marked

class slackly.events.im_open.**ImOpen**

Listing 5.45: Example json response

```
{
    "type": "im_open",
    "user": "U024BE7LH",
    "channel": "D024BE91L"
}
```

For more information see https://api.slack.com/events/im_open

class slackly.events.manual_presence_change.**ManualPresenceChange**

Listing 5.46: Example json response

```
{
    "type": "manual_presence_change",
    "presence": "away"
}
```

For more information see https://api.slack.com/events/manual_presence_change

class slackly.events.member_joined_channel.**MemberJoinedChannel**

Listing 5.47: Example json response

```
{
    "type": "member_joined_channel",
    "user": "W06GH7XHN",
    "channel": "C0698JE0H",
    "channel_type": "C",
    "inviter": "U123456789"
}
```

For more information see https://api.slack.com/events/member_joined_channel

class slackly.events.message.**Message**

Listing 5.48: Example json response

```
{
    "type": "message",
    "channel": "C2147483705",
    "user": "U2147483697",
    "text": "Hello world",
    "ts": "1355517523.000005"
}
```

For more information see https://api.slack.com/events/message

class slackly.events.pin_added.**PinAdded**

Listing 5.49: Example json response

```
{
    "type": "pin_added",
    "user": "U024BE7LH",
    "channel_id": "C02ELGNBH",
    "item": {
         ...
    },
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/pin_added

class slackly.events.pin_removed.**PinRemoved**

Listing 5.50: Example json response

```
{
    "type": "pin_removed",
    "user": "U024BE7LH",
    "channel_id": "C02ELGNBH",
    "item": {
         ...
    },
    "has_pins": false,
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/pin_removed

class slackly.events.pong.**Pong**

Listing 5.51: Example json response

```
{
    'type': 'pong'
}
```

This event type is returned after doing the special ping message

class slackly.events.pref_change.**PrefChange**

Listing 5.52: Example json response

```
{
    "type": "pref_change",
    "name": "messages_theme",
    "value": "dense"
}
```

For more information see https://api.slack.com/events/pref_change

class slackly.events.presence_change.**PresenceChange**

Listing 5.53: Example json response

```
{
    "type": "presence_change",
    "user": "U024BE7LH",
    "presence": "away"
}
```

For more information see https://api.slack.com/events/presence_change

class `slackly.events.reaction_added.`**`ReactionAdded`**

Listing 5.54: Example json response

```
{
    "type": "reaction_added",
    "user": "U024BE7LH",
    "reaction": "thumbsup",
    "item_user": "U0G9QF9C6",
    "item": {
        ...
    },
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/reaction_added

class `slackly.events.reaction_removed.`**`ReactionRemoved`**

Listing 5.55: Example json response

```
{
    "type": "reaction_removed",
    "user": "U024BE7LH",
    "reaction": "thumbsup",
    "item_user":"U0G9QF9C6",
    "item": {
        ...
    },
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/reaction_removed

class `slackly.events.reconnect_url.`**`ReconnectUrl`**

Listing 5.56: Example json response

```
{
    "type": "reconnect_url",
    "url": "wss://mpmulti-0m6f.slack-msgs.com/websocket/...."
}
```

For more information see https://api.slack.com/events/reconnect_url

class `slackly.events.star_added.`**`StarAdded`**

Listing 5.57: Example json response

```
{
    "type": "star_added",
    "user": "U024BE7LH",
    "item": {
        ...
    },
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/star_added

class `slackly.events.star_removed.`**`StarRemoved`**

Listing 5.58: Example json response

```
{
    "type": "star_removed",
    "user": "U024BE7LH",
    "item": {
        ...
    },
    "event_ts": "1360782804.083113"
}
```

For more information see https://api.slack.com/events/star_removed

class `slackly.events.subteam_created.`**`SubteamCreated`**

Listing 5.59: Example json response

```
{
    "type": "subteam_created",
    "subteam": {
        "id": "S0615G0KT",
        "team_id": "T060RNRCH",
        "is_usergroup": true,
        "name": "Marketing Team",
        "description": "Marketing gurus, PR experts and product advocates.",
        "handle": "marketing-team",
        "is_external": false,
        "date_create": 1446746793,
        "date_update": 1446746793,
        "date_delete": 0,
        "auto_type": null,
        "created_by": "U060RNRCZ",
        "updated_by": "U060RNRCZ",
        "deleted_by": null,
        "prefs": {
            "channels": [

            ],
            "groups": [

            ]
        },
        "user_count": "0"
        }
}
```

For more information see https://api.slack.com/events/subteam_created

class slackly.events.subteam_self_added.**SubteamSelfAdded**

Listing 5.60: Example json response

```
{
    "type": "subteam_self_added",
    "subteam_id": "S0615G0KT"
}
```

For more information see https://api.slack.com/events/subteam_self_added

class slackly.events.subteam_self_removed.**SubteamSelfRemoved**

Listing 5.61: Example json response

```
{
    "type": "subteam_self_removed",
    "subteam_id": "S0615G0KT"
}
```

For more information see https://api.slack.com/events/subteam_self_removed

class slackly.events.subteam_updated.**SubteamUpdated**

Listing 5.62: Example json response

```
{
    "type": "subteam_updated",
    "subteam": {
       "id": "S0614TZR7",
       "team_id": "T060RNRCH",
       "is_usergroup": true,
       "name": "Team Admins",
       "description": "A group of all Administrators on your team.",
       "handle": "admins",
       "is_external": false,
       "date_create": 1446598059,
       "date_update": 1446670362,
       "date_delete": 0,
       "auto_type": "admin",
       "created_by": "USLACKBOT",
       "updated_by": "U060RNRCZ",
       "deleted_by": null,
       "prefs": {
           "channels": [

           ],
           "groups": [

           ]
       },
       "users": [
           "U060RNRCZ",
           "U060ULRC0",
           "U06129G2V",
           "U061309JM"
       ],
       "user_count": "4"
    }
}
```

For more information see https://api.slack.com/events/subteam_updated

class `slackly.events.team_domain_change.`**`TeamDomainChange`**

Listing 5.63: Example json response

```
{
    "type": "team_domain_change",
    "url": "https://my.slack.com",
    "domain": "my"
}
```

For more information see https://api.slack.com/events/team_domain_change

class `slackly.events.team_join.`**`TeamJoin`**

Listing 5.64: Example json response

```
{
    "type": "team_join",
    "user": {
         ...
    }
}
```

For more information see https://api.slack.com/events/team_join

class `slackly.events.team_migration_started.`**`TeamMigrationStarted`**

Listing 5.65: Example json response

```
{
    "type": "team_migration_started",
}
```

For more information see https://api.slack.com/events/team_migration_started

class `slackly.events.team_plan_change.`**`TeamPlanChange`**

Listing 5.66: Example json response

```
{
    "type": "team_plan_change",
    "plan": "std"
}
```

For more information see https://api.slack.com/events/team_plan_change

class `slackly.events.team_pref_change.`**`TeamPrefChange`**

Listing 5.67: Example json response

```
{
    "type": "team_pref_change",
    "name": "slackbot_responses_only_admins",
    "value": true
}
```

For more information see https://api.slack.com/events/team_pref_change

class slackly.events.team_profile_change.**TeamProfileChange**

Listing 5.68: Example json response

```
{
    "type": "team_profile_change",
    "profile": {
        "fields": [
            {
                "id": "Xf06054AAA",
                ...
            },
            ...
        ]
    }
}
```

For more information see https://api.slack.com/events/team_profile_change

class slackly.events.team_profile_delete.**TeamProfileDelete**

Listing 5.69: Example json response

```
{
    "type": "team_profile_delete",
    "profile": {
        "fields": [
            "Xf06054AAA",
            ...
        ]
    }
}
```

For more information see https://api.slack.com/events/team_profile_delete

class slackly.events.team_profile_reorder.**TeamProfileReorder**

Listing 5.70: Example json response

```
{
    "type": "team_profile_reorder",
    "profile": {
        "fields": [
            {
                "id": "Xf06054AAA",
                "ordering": 0,
            },
            ...
        ]
    }
}
```

For more information see https://api.slack.com/events/team_profile_reorder

class slackly.events.team_rename.**TeamRename**

Listing 5.71: Example json response

```
{
    "type": "team_rename",
    "name": "New Team Name Inc."
}
```

For more information see https://api.slack.com/events/team_rename

class slackly.events.thread_marked.**ThreadMarked**
This event type is under development

class slackly.events.thread_subscribed.**ThreadSubscribed**
This event type is under development

class slackly.events.thread_unsubscribed.**ThreadUnsubscribed**
This event type is under development

class slackly.events.update_thread_state.**UpdateThreadState**
This event type is under development

class slackly.events.url_verification.**UrlVerification**

Listing 5.72: Example json response

```
{
    "token": "Jhj5dZrVaK7ZwHHjRyZWjbDl",
    "challenge": "3eZbrw1aBm2rZgRNFdxV2595E9CY3gmdALWMmHkvFXO7tYXAYM8P",
    "type": "url_verification"
}
```

For more information see https://api.slack.com/events/url_verification

class slackly.events.user_change.**UserChange**

Listing 5.73: Example json response

```
{
    "type": "user_change",
    "user": {
        ...
    }
}
```

For more information see https://api.slack.com/events/user_change

class slackly.events.user_typing.**UserTyping**

Listing 5.74: Example json response

```
{
    "type": "user_typing",
    "channel": "C02ELGNBH",
    "user": "U024BE7LH"
}
```

For more information see https://api.slack.com/events/user_typing

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index

Symbols