
aiobearychat Documentation

0.2.0

mozillazg

8 27, 2017

Contents

1	Features	3
2	Contents	5
2.1	aiobearychat	5
2.2	Installation	6
2.3	API (OpenAPI)	6
2.4	RTM HTTP API	7
2.5	Contributing	7
2.6	Credits	9
2.7	Changelog	10
2.8	API Reference	10
3	Indices and tables	35
	Python	37

BearyChat Python SDK

- Free software: MIT license
- Documentation: <https://aiobearychat.readthedocs.org>
- GitHub: <https://github.com/mozillazg/aiobearychat>
- PyPI: <https://pypi.python.org/pypi/aiobearychat>
- Python version: 3.5, 3.6

CHAPTER 1

Features

- OpenAPI
- RTM HTTP API
- HTTP aiohttp, tornado, ...)

aiobearychat

BearyChat Python SDK

- Free software: MIT license
- Documentation: <https://aiobearychat.readthedocs.org>
- GitHub: <https://github.com/mozillazg/aiobearychat>
- PyPI: <https://pypi.python.org/pypi/aiobearychat>
- Python version: 3.5, 3.6

Features

- OpenAPI
- RTM HTTP API
- HTTP aiohttp, tornado, ...)

Installation

At the command line:

```
$ pip install aiobearychat[aiohttp]
```

Usage

OpenAPI

```
import aiohttp

from aiobearychat.openapi.aiohttp import OpenAPI

async def main(token):
    async with aiohttp.ClientSession() as session:
        api = OpenAPI(session, token=token)
        response = await api.user.list()
        print(response.json())
```

RTM HTTP API

```
import aiohttp

from aiobearychat.rtm.aiohttp import RtmAPI

async def main(token):
    async with aiohttp.ClientSession() as session:
        api = RtmAPI(session, token=token)
        response = await api.start()
        pprint(response.json())
```

Credits

This package was created with [Cookiecutter](#) and the [mozillazg/cookiecutter-pypackage](#) project template.

Installation

At the command line:

```
$ pip install aiobearychat
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv aiobearychat
$ pip install aiobearychat
```

API (OpenAPI)

```
import aiohttp

from aiobearychat.openapi.aiohttp import OpenAPI
```

```

async def main(token):
    async with aiohttp.ClientSession() as session:
        api = OpenAPI(session, token=token)
        response = await api.user.list()
        print(response.json())

```

aiohttp http
api aiobearychat.openapi.OpenAPI

aiohttp http :

- *aiobearychat.openapi.aiohttp.OpenAPI*

RTM HTTP API

```

import aiohttp

from aiobearychat.rtm.aiohttp import RtmAPI

async def main(token):
    async with aiohttp.ClientSession() as session:
        api = RtmAPI(session, token=token)
        response = await api.start()
        pprint(response.json())

```

aiohttp http
api aiobearychat.rtm.RtmAPI

aiohttp http :

- *aiobearychat.rtm.aiohttp.RtmAPI*

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at <https://github.com/mozillazg/aiobearychat/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

aiobearychat could always use more documentation, whether as part of the official aiobearychat docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mozillazg/aiobearychat/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here’s how to set up *aiobearychat* for local development.

1. Fork the *aiobearychat* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/aiobearychat.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv aiobearychat
$ cd aiobearychat/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 aiobearychat tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5 and 3.6. Check https://travis-ci.org/mozillazg/aiobearychat/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test tests/test_aiobearychat.py
```

Credits

Development Lead

- mozillazg <mozillazg101@gmail.com>

Contributors

None yet. Why not be the first?

Changelog

0.2.0 (2017-08-27)

- Python 3.5
- RTM HTTP API.
- readthedocs
-

0.1.0 (2017-08-20)

- First release on PyPI.
- OpenAPI API.

API Reference

`aiobeyrchat.abc`

```
class aiobeyrchat.abc.API (requester: aiobeyrchat.abc.Requester, token: str, *, base_url: str,
                        **kwargs)
```

```
class aiobeyrchat.abc.Requester
```

```
    request (method: str, url: str, body: bytes = b'', headers: typing.Union[typing.Dict, NoneType] =
            None, **requester_params: typing.Dict) → aiobeyrchat.sansio.Response
```

```
    sleep (seconds: float) → None
```

`aiobeyrchat.requesters.aiohttp`

```
class aiobeyrchat.requesters.aiohttp.Requester (session: aiohttp.client.ClientSession,
                                                *args: typing.Any, **kwargs: typing.Any)
        → None
```

```
    request (method: str, url: str, body: bytes = b'', headers: typing.Union[typing.Dict, NoneType] =
            None, **requester_params: typing.Any) → aiobeyrchat.sansio.Response
```

```
    sleep (seconds: float) → None
```

aiobearychat.sansio

class aiobearychat.sansio.**Response** (*status, headers, body*) **body** = Attribute

headers = Attribute(name='headers', default=NOTHING, validator=None, repr=True, cmp=True, hash=None, init=True, c

json () → typing.Any
body

status = Attribute(name='status', default=NOTHING, validator=None, repr=True, cmp=True, hash=None, init=True, c
200

aiobearychat.sansio.**clean_nothing_keys** (*origin_dict: typing.Dict*) → typing.Dict
value NOTHING key

aiobearychat.sansio.**format_url** (*url: str, url_params: typing.Union[typing.Dict, NoneType] = None*) → str

aiobearychat.openapi

class aiobearychat.openapi.**OpenAPI** (*requester: aiobearychat.abc.Requester, token: str = NOTHING, *, base_url: str = ''*)
API (OpenAPI).

- **requester** (*aiobearychat.abc.Requester*) – requester
- **token** (*str*) – RTM token, hubot
- **base_url** (*str*) – API

Members

Undoc-members

base_url

API <https://api.bearychat.com/v1>

meta

meta API (*aiobearychat.openapi.meta_api.MetaAPI*).

team

API (*aiobearychat.openapi.team_api.TeamAPI*).

user

API (*aiobearychat.openapi.user_api.UserAPI*).

vchannel

API (*aiobearychat.openapi.vchannel_api.VchannelAPI*).

channel

API (*aiobearychat.openapi.channel_api.ChannelAPI*).

session_channel

API (*aiobearychat.openapi.session_channel_api.SessionChannelAPI*).

p2p

P2P API (*aiobearychat.openapi.p2p_api.P2pAPI*).

message

API (*aiobearychat.openapi.message_api.MessageAPI*).

emoji

emoji API(*aiobearychat.openapi.emoji_api.EmojiAPI*).

sticker

sticker API(*aiobearychat.openapi.sticker_api.StickerAPI*).

rtm

RTM API(*aiobearychat.openapi.rtm_api.RtmAPI*).

aiobearychat.openapi.aiohttp

class aiobearychat.openapi.aiohttp.**OpenAPI** (*session: aiohttp.client.ClientSession, token: str = NOTHING, *, base_url: str = ''*)

aiohttp OpenAPI API HTTP

:

```

async def main(token):
    async with aiohttp.ClientSession() as session:
        api = OpenAPI(session, token=token)
        response = await api.user.list()
        print(response.json())
    
```

aiobearychat.openapi.channel_api

API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.channel_api.**ChannelAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

API.

archive (*channel_id: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.archive.md>

channel_id(str) – id (: =bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```

response.status 200 :
{
  "private": false,
  "general": true,
  "latest_ts": 1486367046281,
  "uid": "=bw520",
  "name": "",
  "is_member": false,
  "is_active": false,
  "type": "normal",
  "topic": null,
  "member_uids": [
    "=bw520"
  ],
}
    
```



```

    "vchannel_id": "=bw520",
    "id": "=bw520",
    "team_id": "=bw520"
  }

```

create (*name: str, *, topic: str = NOTHING, private: bool = NOTHING, **requester_params: typing.Any*) → aiobearychat.sansio.Response

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.create.md>

param name

2 20 1 10

(_)(-) (:):type name: str

param topic (:)

type topic str

param private

type private bool

return response. *aiobearychat.sansio.Response*

rtype *aiobearychat.sansio.Response*

response.body:

```

response.status 201 :
  {
    "private": false,
    "general": true,
    "latest_ts": 1486367046281,
    "uid": "=bw520",
    "name": "",
    "is_member": false,
    "is_active": true,
    "type": "normal",
    "topic": null,
    "member_uids": [
      "=bw520"
    ],
    "vchannel_id": "=bw520",
    "id": "=bw520",
    "team_id": "=bw520"
  }

```

info (*channel_id: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.info.md>

channel_id (str) – id (=bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 200 :
{
  "private": false,
  "general": true,
  "latest_ts": 1486367046281,
  "uid": "=bw520",
  "name": "",
  "is_member": false,
  "is_active": true,
  "type": "normal",
  "topic": null,
  "member_uids": [
    "=bw520"
  ],
  "vchannel_id": "=bw520",
  "id": "=bw520",
  "team_id": "=bw520"
}
```

invite (*channel_id*: str, *invite_uid*: str, ***requester_params*: typing.Any) → aioberychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.invite.md>

- **channel_id**(str)– id (: =bw520)
- **invite_uid**(str)– id (: =bw520)

response. *aioberychat.sansio.Response*
aioberychat.sansio.Response

response.body :

```
response.status 204 :
```

join (*channel_id*: str, ***requester_params*: typing.Any) → aioberychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.join.md>

channel_id(str)– id (: =bw520)
response. *aioberychat.sansio.Response*
aioberychat.sansio.Response

response.body :

```
response.status 200 :
{
  "private": false,
  "general": false,
  "latest_ts": 1486367046281,
  "uid": "=bw520",
  "name": "",
  "is_member": false,
  "is_active": true,
  "type": "normal",
```

```

    "topic": null,
    "member_uids": [
      "=bw520"
    ],
    "vchannel_id": "=bw520",
    "id": "=bw520",
    "team_id": "=bw520"
  }

```

kick (*channel_id: str, kick_uid: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.kick.md>

- **channel_id**(str) – id (: =bw520)
- **kick_uid**(str) – id (: =bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 204 :
```

kickout (*channel_id: str, kick_uid: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.kickout.md>

- **channel_id**(str) – id (: =bw520)
- **kick_uid**(str) – id (: =bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 204 :
```

leave (*channel_id: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.leave.md>

channel_id(str) – id (: =bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 204 :
```

list (***requester_params: typing.Any*) → aiobearychat.sansio.Response
channel.info.

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.list.md>

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

```
response.body:
```

```
response.status 200 :
[
  {
    "private": false,
    "general": true,
    "latest_ts": 1486367046281,
    "uid": "=bw520",
    "name": "",
    "is_member": false,
    "is_active": true,
    "type": "normal",
    "topic": null,
    "member_uids": [
      "=bw520"
    ],
    "vchannel_id": "=bw520",
    "id": "=bw520",
    "team_id": "=bw520"
  }
]
```

unarchive (*channel_id: str, **requester_params: typing.Any*) → *aiobearychat.sansio.Response*
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/channel.unarchive.md>

channel_id(str) – id (=bw520)

response. aiobearychat.sansio.Response

aiobearychat.sansio.Response

```
response.body:
```

```
response.status 200 :
{
  "private": false,
  "general": true,
  "latest_ts": 1486367046281,
  "uid": "=bw520",
  "name": "",
  "is_member": false,
  "is_active": true,
  "type": "normal",
  "topic": null,
  "member_uids": [
    "=bw520"
  ],
  "vchannel_id": "=bw520",
  "id": "=bw520",
  "team_id": "=bw520"
}
```

aiobearychat.openapi.emoji_api

emoji API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.emoji_api.**EmojiAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

emoji API.

list (***requester_params: typing.Any*) → aiobearychat.sansio.Response
emoji .

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/emoji.list.md>

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 200 :

[
  {
    "id": "=bw520",
    "uid": "=bw520",
    "team_id": "=bw520",
    "name": "hello",
    "type": "emoji",
    "created": "2017-03-13T13:54:16.000+0000",
    "updated": "2017-03-13T13:54:16.000+0000",
    "url": "http://example.com/1.jpg"
  }
]
```

aiobearychat.openapi.message_api

API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.message_api.**MessageAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

API.

create (*vchannel_id: str, text: str, attachments: list, **requester_params: typing.Any*) →
aiobearychat.sansio.Response

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/message.create.md>

- **vchannel_id** (str)– id (=bw520)
- **text** (str)– (:)
- **attachments** (list)–

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 201 :

{
```

```

    "key": "1485236262366.0193",
    "updated": "2017-01-24T13:37:42.000+0000",
    "is_channel": false,
    "uid": "=bw520",
    "fallback": null,
    "attachments": [],
    "created": "2017-01-24T13:37:42.000+0000",
    "vchannel_id": "=bw520",
    "refer_key": null,
    "robot_id": null,
    "created_ts": 1485236262366,
    "team_id": "=bw520",
    "subtype": "normal",
    "text": "hello"
  }

```

delete (*vchannel_id*: str, *message_key*: str, ***requester_params*: typing.Any) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/message.delete.md>

- **vchannel_id** (str) – id (: =bw520)
- **message_key** (str) – key (: 1487667236785.0077)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 204 :
```

info (*vchannel_id*: str, *message_key*: str, ***requester_params*: typing.Any) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/message.info.md>

- **vchannel_id** (str) – id (: =bw520)
- **message_key** (str) – key (: 1487667236785.0077)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```

response.status 200 :

{
  "key": "1485236262366.0193",
  "updated": "2017-01-24T13:37:42.000+0000",
  "is_channel": false,
  "uid": "=bw520",
  "fallback": null,
  "attachments": [],
  "created": "2017-01-24T13:37:42.000+0000",
  "vchannel_id": "=bw520",
  "refer_key": null,

```

```

    "robot_id": null,
    "created_ts": 1485236262366,
    "team_id": "=bw520",
    "subtype": "normal",
    "text": "hello"
  }

```

query (*vchannel_id*: str, *query*: str, ***requester_params*: typing.Any) → aiobearychat.sansio.Response
vchannel

vchannel

- limit: 100, 20

- key: key, ts

- ts: ts, ts

- forward:

- backward:

:

- 1.key key

- 2.ts ts

- 3.forward/backward

- 4.forward/backward 100,

- 5.forward/backward forward=100

- from_key/to_key: key

- from_ts/to_ts: ts

- forward: from to

- backward: to from

:

- 1.{from,to}_key {from,to}_ts

- 2.{from,to}_key key

- 3.{from,to}_ts ts

- 4.forward backward

- 5.forward/backward forward=100

6. <https://github.com/bearyinnovative/OpenAPI/blob/master/api/message.query.md>

param vchannel_id vchannel_id (: =bw520)

type vchannel_id str

param query payload

type query str

return response. *aiobearychat.sansio.Response*

rtype *aiobearychat.sansio.Response*

response.body:

```
response.status 200 :
{
  "messages": [
    {
      "key": "1485236262366.0193",
      "updated": "2017-01-24T13:37:42.000+0000",
      "is_channel": false,
      "uid": "=bw520",
      "fallback": null,
      "attachments": [],
      "created": "2017-01-24T13:37:42.000+0000",
      "vchannel_id": "=bw520",
      "refer_key": null,
      "robot_id": null,
      "created_ts": 1485236262366,
      "team_id": "=bw520",
      "subtype": "normal",
      "text": "hello"
    }
  ]
}
```

update_text (*vchannel_id*: str, *message_key*: str, *text*: str, ***requester_params*: typing.Any) →

aiobearychat.sansio.Response

https://github.com/bearyinnovative/OpenAPI/blob/master/api/message.update_text.md

- **vchannel_id** (str)– id (: =bw520)
- **message_key** (str)– key (: 1487667236785.0077)
- **text** (str)– (:)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body:

```
response.status 200 :
{
  "key": "1485236262366.0193",
  "updated": "2017-01-24T13:37:42.000+0000",
  "is_channel": false,
  "uid": "=bw520",
  "fallback": null,
  "attachments": [],
  "created": "2017-01-24T13:37:42.000+0000",
  "vchannel_id": "=bw520",
  "refer_key": null,
  "robot_id": null,
  "created_ts": 1485236262366,
  "team_id": "=bw520",
  "subtype": "normal",
```



```

    "text": "hello"
  }

```

aiobearychat.openapi.meta_api

meta API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.meta_api.**MetaAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

meta API.

get (***requester_params: typing.Any*) → aiobearychat.sansio.Response
BearyChat API

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/meta.md>

response. aiobearychat.sansio.Response

aiobearychat.sansio.Response

response.body:

```

response.status 200 :
{
  "version": "1"
}

```

aiobearychat.openapi.p2p_api

P2P API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.p2p_api.**P2pAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

P2P API.

create (*user_id: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response
P2P

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/p2p.create.md>

user_id(str) – P2P id (=bw520)

response. aiobearychat.sansio.Response

aiobearychat.sansio.Response

response.body:

```

response.status 201 :
{
  "id": "=bw520",
  "team_id": "=bw520",
  "vchannel_id": "=bw520",
  "type": "p2p",
}

```

```

    "is_active": true,
    "is_member": true,
    "member_uids": [
      "=bw520",
      "=bw52P"
    ],
    "latest_ts": 1485238998284
  }

```

info (*p2p_channel_id*: str, ***requester_params*: typing.Any) → aiobearychat.sansio.Response
P2P

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/p2p.info.md>

p2p_channel_id(str) – P2P id (=bw520)
response. *aiobearychat.sansio.Response*
aiobearychat.sansio.Response

response.body:

```

response.status 200 :
{
  "id": "=bw520",
  "team_id": "=bw520",
  "vchannel_id": "=bw520",
  "type": "p2p",
  "is_active": true,
  "is_member": true,
  "member_uids": [
    "=bw520",
    "=bw52P"
  ],
  "latest_ts": 1485238998284
}

```

list (***requester_params*: typing.Any) → aiobearychat.sansio.Response
P2P P2P p2p.info.

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/p2p.list.md>

response. *aiobearychat.sansio.Response*
aiobearychat.sansio.Response

response.body:

```

response.status 200 :
[
  {
    "id": "=bw520",
    "team_id": "=bw520",
    "vchannel_id": "=bw520",
    "type": "p2p",
    "is_active": true,
    "is_member": true,
    "member_uids": [
      "=bw520",

```

```

    "=bw52P"
  ],
  "latest_ts": 1485238998284
}
]

```

aioberychat.openapi.rtm_api

RTM API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aioberychat.openapi.rtm_api.**RtmAPI** (*requester: aioberychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

RTM API.

start (***requester_params: typing.Any*) → aioberychat.sansio.Response
RTM .

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/rtm.start.md>

response. *aioberychat.sansio.Response*

aioberychat.sansio.Response

response.body:

```

response.status 200 :

{
  "ws_host": "wss://rtm.bearychat.com/nimbus/ws:xxx",
  "user": {
    "inactive": false,
    "role": "normal",
    "email": "support@bearyinnovative.com",
    "name": "BearyBot",
    "type": "assistant",
    "created": "2017-01-11T12:28:31.000+0000",
    "id": "=bwMkR",
    "avatars": {
      "small": null,
      "medium": null,
      "large": null
    },
  },
  "team_id": "=bw520",
  "full_name": "",
  "mobile": null,
  "profile": {
    "bio": null,
    "position": null,
    "skype": null
  }
}
}

```

aiobearychat.openapi.session_channel_api

API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

```
class aiobearychat.openapi.session_channel_api.SessionChannelAPI (requester:
                                aiobearychat.abc.Requester,
                                token: str, *,
                                base_url: str,
                                **kwargs)
```

API.

```
archive (session_channel_id: str, **requester_params: typing.Any) → aiobearychat.sansio.Response
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session\_channel.archive.md
```

```
session_channel_id (str) – id (: =bw520)
```

```
response. aiobearychat.sansio.Response
```

```
aiobearychat.sansio.Response
```

```
response.body :
```

```
response.status 200 :
{
  "latest_ts": "1489242467694",
  "name": "",
  "is_member": true,
  "is_active": false,
  "type": "session",
  "member_uids": [
    "bw520"
  ],
  "vchannel_id": "bw520",
  "id": "bw520",
  "team_id": "bw520"
}
```

```
convert_to_channel (session_channel_id: str, name: str, *, private: bool = True, **re-
                    requester_params: typing.Any) → aiobearychat.sansio.Response
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session\_channel.convert\_to\_
channel.md
```

```
param session_channel_id id (: =bw520)
```

```
type session_channel_id str
```

```
param name
```

```
2 20 1 10
```

```
(_) (-) (: ) :type name: str
```

```
param private : True
```

```
type private bool
```

```
return response. aiobearychat.sansio.Response
```

rtype *aiobearychat.sansio.Response*

response.body :

```
response.status 201 :

{
  "latest_ts": "1489242467694",
  "name": "",
  "is_member": true,
  "is_active": false,
  "type": "session",
  "member_uids": [
    "bw520"
  ],
  "vchannel_id": "bw520",
  "id": "bw520",
  "team_id": "bw520"
}
```

create (*member_uids*: <function *SessionChannelAPI.list* at 0x7f4f011f5b70>, *, *name*: str = NOTHING, ****requester_params**: typing.Any) → aiobearychat.sansio.Response
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.create.md

param name

2 20 1 10

(_)(-)(:):type name: str

param member_uids id (: ["bw520", "bw52P"])

type member_uids list

return response. *aiobearychat.sansio.Response*

rtype *aiobearychat.sansio.Response*

response.body :

```
response.status 201 :

{
  "latest_ts": "1489242467694",
  "name": "",
  "is_member": true,
  "is_active": true,
  "type": "session",
  "member_uids": [
    "bw520"
  ],
  "vchannel_id": "bw520",
  "id": "bw520",
  "team_id": "bw520"
}
```

info (*session_channel_id*: str, ****requester_params**: typing.Any) → aiobearychat.sansio.Response
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.info.md

```

session_channel_id(str) – id(=bw520)
response. aiobearychat.sansio.Response
aiobearychat.sansio.Response

```

response.body:

```

response.status 200 :
{
  "latest_ts": "1489242467694",
  "name": "",
  "is_member": true,
  "is_active": true,
  "type": "session",
  "member_uids": [
    "bw520"
  ],
  "vchannel_id": "bw520",
  "id": "bw520",
  "team_id": "bw520"
}

```

invite (*session_channel_id*: str, *invite_uid*: str, ***requester_params*: typing.Any) → *aiobearychat.sansio.Response*
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.invite.md

- **session_channel_id**(str) – id(=bw520)
- **invite_uid**(str) – id(=bw520)

```

response. aiobearychat.sansio.Response
aiobearychat.sansio.Response

```

response.body:

```

response.status 204 :

```

kick (*session_channel_id*: str, *kick_uid*: str, ***requester_params*: typing.Any) → *aiobearychat.sansio.Response*
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.kick.md

- **session_channel_id**(str) – id(=bw520)
- **kick_uid**(str) – id(=bw520)

```

response. aiobearychat.sansio.Response
aiobearychat.sansio.Response

```

response.body:

```

response.status 204 :

```

kickout (*session_channel_id*: str, *kick_uid*: str, ***requester_params*: typing.Any) → *aiobearychat.sansio.Response*
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.kickout.md

- **session_channel_id**(str) – id(=bw520)
- **kick_uid**(str) – id(=bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body:

```
response.status 204 :
```

leave (*session_channel_id: str, **requester_params: typing.Any*) → *aiobearychat.sansio.Response*
https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.leave.md

session_channel_id(str) – id(=bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body:

```
response.status 204 :
```

list (***requester_params: typing.Any*) → *aiobearychat.sansio.Response*

session_channel.info.

https://github.com/bearyinnovative/OpenAPI/blob/master/api/session_channel.list.md

return response. *aiobearychat.sansio.Response*

rtype *aiobearychat.sansio.Response*

response.body:

```
response.status 200 :

[
  {
    "latest_ts": "1489242467694",
    "name": "",
    "is_member": true,
    "is_active": true,
    "type": "session",
    "member_uids": [
      "bw520"
    ],
    "vchannel_id": "bw520",
    "id": "bw520",
    "team_id": "bw520"
  }
]
```

aiobearychat.openapi.sticker_api

sticker API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.sticker_api.**StickerAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

sticker API.

list (***requester_params: typing.Any*) → aiobearychat.sansio.Response
sticker .

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/sticker.list.md>

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 200 :  
  
[  
  {  
    "pack": "",  
    "stickers": [  
      {  
        "url": "https://dn-bearychat.qbox.me/sticker-gif-20.gif",  
        "name": "",  
        "width": 240,  
        "height": 240  
      }  
    ]  
  }  
]
```

aiobearychat.openapi.team_api

API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.team_api.**TeamAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

API.

info (***requester_params: typing.Any*) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/team.info.md>

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 200 :  
  
{  
  "id": "bw520",  
  "subdomain": "openapi",  
}
```



```

    "name": "BearyChat OpenAPI",
    "email_domain": null,
    "logo_url": null,
    "description": "",
    "plan": "basic",
    "created": "2017-01-11T12:28:31.000+0000"
  }

```

aiobearychat.openapi.user_api

API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.user_api.**UserAPI** (*requester: aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs*)

API.

info (*user_id: str, **requester_params: typing.Any*) → aiobearychat.sansio.Response

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/user.info.md>

user_id (str) – id (:=bw520)

response. aiobearychat.sansio.Response

aiobearychat.sansio.Response

response.body:

```

response.status 200 :

{
  "inactive": false,
  "role": "normal",
  "email": "support@bearyinnovative.com",
  "name": "BearyBot",
  "type": "assistant",
  "created": "2017-01-11T12:28:31.000+0000",
  "id": "=bwMkR",
  "avatars": {
    "small": null,
    "medium": null,
    "large": null
  },
  "team_id": "=bw520",
  "full_name": "",
  "mobile": null,
  "profile": {
    "bio": null,
    "position": null,
    "skype": null
  }
}

```

list (***requester_params: typing.Any*) → aiobearychat.sansio.Response

user.info.

<https://github.com/bearyinnovative/OpenAPI/blob/master/api/user.list.md>

```
response. aiobearychat.sansio.Response
```

```
aiobearychat.sansio.Response
```

```
response.body:
```

```
response.status 200 :

[
  {
    "inactive": false,
    "role": "normal",
    "email": "support@bearyinnovative.com",
    "name": "BearyBot",
    "type": "assistant",
    "created": "2017-01-11T12:28:31.000+0000",
    "id": "=bwMkR",
    "avatars": {
      "small": null,
      "medium": null,
      "large": null
    },
    "team_id": "=bw520",
    "full_name": "",
    "mobile": null,
    "profile": {
      "bio": null,
      "position": null,
      "skype": null
    }
  }
]
```

```
me (**requester_params: typing.Any) → aiobearychat.sansio.Response
```

```
https://github.com/bearyinnovative/OpenAPI/blob/master/api/user.me.md
```

```
response. aiobearychat.sansio.Response
```

```
aiobearychat.sansio.Response
```

```
response.body:
```

```
response.status 200 :

{
  "inactive": false,
  "role": "normal",
  "email": "support@bearyinnovative.com",
  "name": "BearyBot",
  "type": "assistant",
  "created": "2017-01-11T12:28:31.000+0000",
  "id": "=bwMkR",
  "avatars": {
    "small": null,
    "medium": null,
    "large": null
  },
  "team_id": "=bw520",
  "full_name": "",
  "mobile": null,
```

```

    "profile": {
      "bio": null,
      "position": null,
      "skype": null
    }
  }
}

```

update_me (*, name: str = NOTHING, full_name: str = NOTHING, **requester_params: typing.Any)
 → aiobearychat.sansio.Response
https://github.com/bearyinnovative/OpenAPI/blob/master/api/user.update_me.md

- **name** (str)– (@name) (: beary_bot)
- **full_name** (str)– (: BearyBot)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body:

```

response.status 200 :

{
  "inactive": false,
  "role": "normal",
  "email": "support@bearyinnovative.com",
  "name": "BearyBot",
  "type": "assistant",
  "created": "2017-01-11T12:28:31.000+0000",
  "id": "=bwMkR",
  "avatars": {
    "small": null,
    "medium": null,
    "large": null
  },
  "team_id": "=bw520",
  "full_name": "",
  "mobile": null,
  "profile": {
    "bio": null,
    "position": null,
    "skype": null
  }
}

```

aiobearychat.openapi.vchannel_api

API.

auto generated by script at 2017-08-24 21:50:13.459796. don't edit it.

class aiobearychat.openapi.vchannel_api.VchannelAPI (requester:
aiobearychat.abc.Requester, token: str, *, base_url: str, **kwargs)

API.

info (*vchannel_id*: str, ***requester_params*: typing.Any) → aiobearychat.sansio.Response
<https://github.com/bearyinnovative/OpenAPI/blob/master/api/vchannel.info.md>

vchannel_id(str) – id (=bw520)

response. *aiobearychat.sansio.Response*

aiobearychat.sansio.Response

response.body :

```
response.status 200 :

// Channel
{
  "private": false,
  "general": true,
  "latest_ts": 1486367046281,
  "uid": "=bw520",
  "name": "",
  "is_member": false,
  "is_active": true,
  "type": "normal",
  "topic": null,
  "member_uids": [
    "=bw520"
  ],
  "vchannel_id": "=bw520",
  "id": "=bw520",
  "team_id": "=bw520"
}

// SessionChannel
{
  "latest_ts": "1489242467694",
  "name": "",
  "is_member": true,
  "is_active": true,
  "type": "session",
  "member_uids": [
    "=bw520"
  ],
  "vchannel_id": "=bw520",
  "id": "=bw520",
  "team_id": "=bw520"
}

// P2PChannel
{
  "id": "=bw520",
  "team_id": "=bw520",
  "vchannel_id": "=bw520",
  "type": "p2p",
  "is_active": true,
  "is_member": true,
  "member_uids": [
    "=bw520",
    "=bw52P"
  ],
  "latest_ts": 1485238998284
```

```
}

```

aioberychat.rtm

class aioberychat.rtm.**RtmAPI** (*requester: aioberychat.abc.Requester, token: str = NOTHING, *, base_url: str = ''*)

RTM HTTP API.

- **requester** (*aioberychat.abc.Requester*) – requester
- **token** (*str*) – RTM token, hubot
- **base_url** (*str*) – RTM HTTP API

base_url = 'https://rtm.bearychat.com'

RTM HTTP API

message (*vchannel: str, text: str, *, markdown: bool = True, attachments: typing.Union[list, NoneType] = None, **requester_params: typing.Any*) → aioberychat.sansio.Response

https://github.com/bearyinnovative/OpenAPI/blob/master/rtm/api_message.md

- **vchannel** (*str*) – id=bw520
- **text** (*str*) –
- **markdown** (*bool*) – markdown True
- **attachments** (*list*) – attachment:

```
[
  {
    "title": "test_title",      # attachment title / text
    "text": "test_text",      # attachment title / text
    "images": [{              # attachment
      "url": "http://example.com/1.jpg"
    }],
    "color": "#cb3f20"        # attachment
  }
]
```

response. *aioberychat.sansio.Response*

aioberychat.sansio.Response

response.body:

```
response.status 200 :

{
  "code": 0,
  "result": null
}
```

start (***requester_params: typing.Any*) → aioberychat.sansio.Response

RTM

https://github.com/bearyinnovative/OpenAPI/blob/master/rtm/api_start.md

```
response. aiobearychat.sansio.Response
aiobearychat.sansio.Response
response.body:
```

```
response.status 200 :
{
  "code": 0,
  "result": {
    // RTM token
    "user": {},
    // RTM 5
    "ws_host": "wss://rtm.bearychat.com/nimbus/ws:xxx"
  }
}
```

aiobearychat.rtm.aiohttp

class aiobearychat.rtm.aiohttp.**RtmAPI** (*session: aiohttp.client.ClientSession, token: str = NOTHING, *, base_url: str = ''*)

aiohttp RtmAPI API HTTP

:

```
async def main(token):
    async with aiohttp.ClientSession() as session:
        api = RtmAPI(session, token=token)
        response = await api.start()
        print(response.json())
```

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

a

aiobearychat.abc, 10
aiobearychat.openapi, 11
aiobearychat.openapi.aihttp, 12
aiobearychat.openapi.channel_api, 12
aiobearychat.openapi.emoji_api, 16
aiobearychat.openapi.message_api, 17
aiobearychat.openapi.meta_api, 21
aiobearychat.openapi.p2p_api, 21
aiobearychat.openapi.rtm_api, 23
aiobearychat.openapi.session_channel_api,
24
aiobearychat.openapi.sticker_api, 28
aiobearychat.openapi.team_api, 28
aiobearychat.openapi.user_api, 29
aiobearychat.openapi.vchannel_api, 31
aiobearychat.requesters.aihttp, 10
aiobearychat.rtm.aihttp, 34
aiobearychat.sansio, 11

A

aiobearychat.abc (), 10
aiobearychat.openapi (), 11
aiobearychat.openapi.aiohhttp (), 12
aiobearychat.openapi.channel_api (), 12
aiobearychat.openapi.emoji_api (), 16
aiobearychat.openapi.message_api (), 17
aiobearychat.openapi.meta_api (), 21
aiobearychat.openapi.p2p_api (), 21
aiobearychat.openapi.rtm_api (), 23
aiobearychat.openapi.session_channel_api (), 24
aiobearychat.openapi.sticker_api (), 28
aiobearychat.openapi.team_api (), 28
aiobearychat.openapi.user_api (), 29
aiobearychat.openapi.vchannel_api (), 31
aiobearychat.requesters.aiohhttp (), 10
aiobearychat.rtm.aiohhttp (), 34
aiobearychat.sansio (), 11
archive() (aiobearychat.openapi.channel_api.ChannelAPI), 12
archive() (aiobearychat.openapi.session_channel_api.SessionChannelAPI), 24

B

base_url (aiobearychat.openapi.OpenAPI), 11
base_url (aiobearychat.rtm.RtmAPI), 33
body (aiobearychat.sansio.Response), 11

C

channel (aiobearychat.openapi.OpenAPI), 11
convert_to_channel() (aiobearychat.openapi.session_channel_api.SessionChannelAPI), 24
create() (aiobearychat.openapi.channel_api.ChannelAPI), 13
create() (aiobearychat.openapi.message_api.MessageAPI), 17
create() (aiobearychat.openapi.p2p_api.P2pAPI), 21
create() (aiobearychat.openapi.session_channel_api.SessionChannelAPI), 25

D

delete() (aiobearychat.openapi.message_api.MessageAPI), 18

E

emoji (aiobearychat.openapi.OpenAPI), 11

G

get() (aiobearychat.openapi.meta_api.MetaAPI), 21

H

headers (aiobearychat.sansio.Response), 11

I

info() (aiobearychat.openapi.channel_api.ChannelAPI), 13
info() (aiobearychat.openapi.message_api.MessageAPI), 18
info() (aiobearychat.openapi.p2p_api.P2pAPI), 22
info() (aiobearychat.openapi.session_channel_api.SessionChannelAPI), 25
info() (aiobearychat.openapi.team_api.TeamAPI), 28
info() (aiobearychat.openapi.user_api.UserAPI), 29
info() (aiobearychat.openapi.vchannel_api.VchannelAPI), 31
invite() (aiobearychat.openapi.channel_api.ChannelAPI), 14
invite() (aiobearychat.openapi.session_channel_api.SessionChannelAPI), 26

J

join() (aiobearychat.openapi.channel_api.ChannelAPI), 14
json() (aiobearychat.sansio.Response), 11

K

kick() (aiobearychat.openapi.channel_api.ChannelAPI), 15
kick() (aiobearychat.openapi.session_channel_api.SessionChannelAPI), 26

kickout() (aiobeyrchat.openapi.channel_api.ChannelAPI update_me() (aiobeyrchat.openapi.user_api.UserAPI),
, 15 31
kickout() (aiobeyrchat.openapi.session_channel_api.SessionChannelAPI (aiobeyrchat.openapi.message_api.MessageAPI
, 26), 20

L

leave() (aiobeyrchat.openapi.channel_api.ChannelAPI), 15
leave() (aiobeyrchat.openapi.session_channel_api.SessionChannelAPI), 27
list() (aiobeyrchat.openapi.channel_api.ChannelAPI), 15
list() (aiobeyrchat.openapi.emoji_api.EmojiAPI), 17
list() (aiobeyrchat.openapi.p2p_api.P2pAPI), 22
list() (aiobeyrchat.openapi.session_channel_api.SessionChannelAPI), 27
list() (aiobeyrchat.openapi.sticker_api.StickerAPI), 28
list() (aiobeyrchat.openapi.user_api.UserAPI), 29

V

vchannel (aiobeyrchat.openapi.OpenAPI), 11

M

me() (aiobeyrchat.openapi.user_api.UserAPI), 30
message (aiobeyrchat.openapi.OpenAPI), 11
message() (aiobeyrchat.rtm.RtmAPI), 33
meta (aiobeyrchat.openapi.OpenAPI), 11

P

p2p (aiobeyrchat.openapi.OpenAPI), 11

Q

query() (aiobeyrchat.openapi.message_api.MessageAPI), 19

R

request() (aiobeyrchat.abc.Requester), 10
request() (aiobeyrchat.requesters.aiohhttp.Requester), 10
rtm (aiobeyrchat.openapi.OpenAPI), 12

S

session_channel (aiobeyrchat.openapi.OpenAPI), 11
sleep() (aiobeyrchat.abc.Requester), 10
sleep() (aiobeyrchat.requesters.aiohhttp.Requester), 10
start() (aiobeyrchat.openapi.rtm_api.RtmAPI), 23
start() (aiobeyrchat.rtm.RtmAPI), 33
status (aiobeyrchat.sansio.Response), 11
sticker (aiobeyrchat.openapi.OpenAPI), 12

T

team (aiobeyrchat.openapi.OpenAPI), 11

U

unarchive() (aiobeyrchat.openapi.channel_api.ChannelAPI), 16