
heatherrd Documentation

Release 1.0.0

Praekelt Foundation

March 22, 2016

1	Connecting a Slack account	3
2	Disconnecting a Slack account	5
3	Send a custom RTM message	7

A simple Python microservice that turns RTM into a synchronous HTTP interface. It's built for [Heatherr](#) which provides some commands & bot functionality for Praekelt's Slack accounts.

```
$ pip install heatherrd
$ heatherrd --listen=tcp:8001 --url=http://example.com/callback-url/
```

There's an example application in `echo.py` to illustrate how it works. Run the following commands to give it a go.

Start heatherrd:

```
$ heatherrd --url=http://localhost:8000
```

Start the echo app on <http://localhost:8000>

```
$ python echo.py
```

Have heatherrd setup the RTM session:

```
$ curl -X POST --user 'bot-user-id:bot-access-token' \
  http://localhost:8001/connect
```

Say something in Slack to your bot and it will echo it back to you via RTM.

```
$ curl --user <bot-user-id>:<bot-access-token> http://localhost:8001/connect
```

It'll maintain the RTM connections to Slack for the bot and relay incoming RTM messages as JSON to the URL provided.

The URL can echo an array of RTM responses as JSON which will be sent back over the RTM connection as a reply.

Connecting a Slack account

```
$ curl -XPOST http://localhost:8001/connect --user 'bot-user-id:bot-access-token'
```

This returns the session data as returned by the [rtm.start](#) API call.

Disconnecting a Slack account

```
$ curl -XPOST http://localhost:8001/disconnect --user 'bot-user-id:bot-access-token'
```

Send a custom RTM message

```
$ curl -XPOST http://localhost:8001/rtm \  
  -d '{"id": 1, "type": "message", "channel": "C024BE91L", "text": "Hello world"}' \  
  --user 'bot-user-id:bot-access-token'
```

The easiest way to generate replies is to have applications reply with JSON inline in the response to the HTTP callbacks. Have a look at `echo.py` for an example.