Slack Bot

**What is slack bot?**

Bot users have many of the same qualities as their human counterparts: they have profile photos, names, and bios, they exist in the team directory, they can be direct messaged or mentioned, they can post messages and upload files.

The biggest difference between bot users and regular users is that instead of interacting with a team via one of Slack's mobile or desktop apps, bot users are controlled programmatically via a bot user token. Bot users can't "log-in", they don't have a password.

**HTTP Stay Alive vs Web Sockets**

A Keep Alive HTTP header since HTTP 1.0, which is used to indicate a HTTP client would like to maintain a persistent connection with HTTP server. The main objects is to eliminate the needs for opening TCP connection for each HTTP request. However, while there is a persistent connection open, the protocol for communication between client and server is still following the basic HTTP request/response pattern. In other word, server side can't push data to client.

WebSocket is completely different mechanism, which is used to setup a persistent, full-duplex connection. With this full-duplex connection, server side can push data to client and client should be expected to process data from server side at any time.

WebSocket is used in this slack bot designing.

**Real Time Messaging**

The Real Time Messaging API is a WebSocket-based API that allows you to receive [events](https://api.slack.com/events) from Slack in real time and send messages as users. To begin a RTM session make an [authenticated](https://api.slack.com/docs/oauth) call to the [rtm.connect](https://api.slack.com/methods/rtm.connect). The Websocket URLs provided by rtm.connect are single-use and are only valid for 30 seconds, so make sure to connect quickly.

You can send a message to Slack by sending JSON over the websocket connection. Every event should have a unique (for that connection) positive integer ID. All replies to that message will include this ID allowing the client to correlate responses with the messages sent; replies may be "out of order" due to the asynchronous nature of the message servers. The message server will disconnect any client that sends a message longer than 16 kilobytes.This includes all parts of the message, including JSON syntax, not just the message text.