
Typhoon Discord Bot

Release 2.0.0

Fisch37

Apr 13, 2023

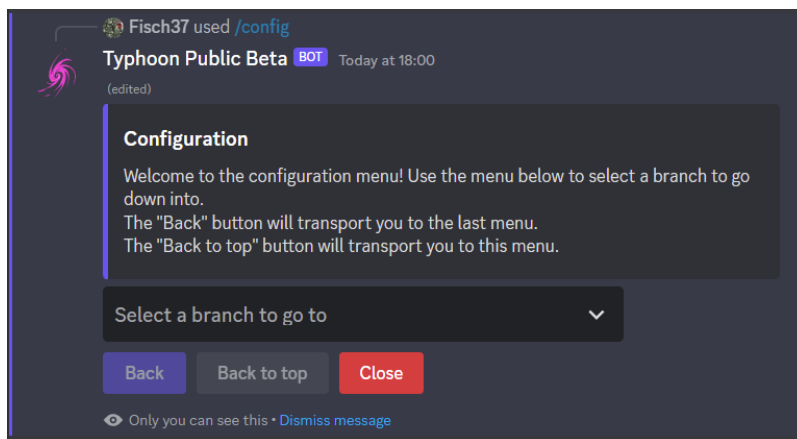
CONTENTS:

1	Getting Started with Typhoon	1
1.1	Configuring the Bot	1
1.2	Fun	3
1.3	Leveling	6
1.4	Moderation	10
1.5	Utility	13
2	Setting up your own version of Typhoon	17
3	Indices and tables	19

GETTING STARTED WITH TYPHOON

To get started using Typhoon, [click here to invite the bot](#). That's all you need to do for Typhoon to work! You might want an introduction to the features though. For this, read further into what interests you.

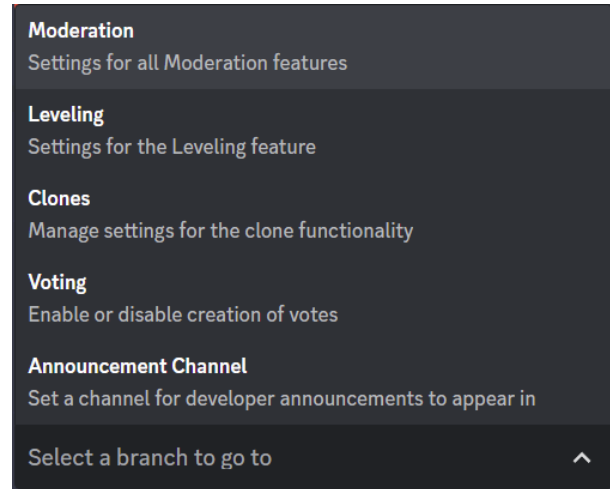
1.1 Configuring the Bot



To allow easy use of the bot Typhoon comes with the command `/config`. Using the UI provided by the command you can edit almost all of the bot's settings. Some settings such as those provided by `/lock` are not available since they reference specific channels. The config menu uses a few terms such as *branch*. A branch is just a submenu within the config system.

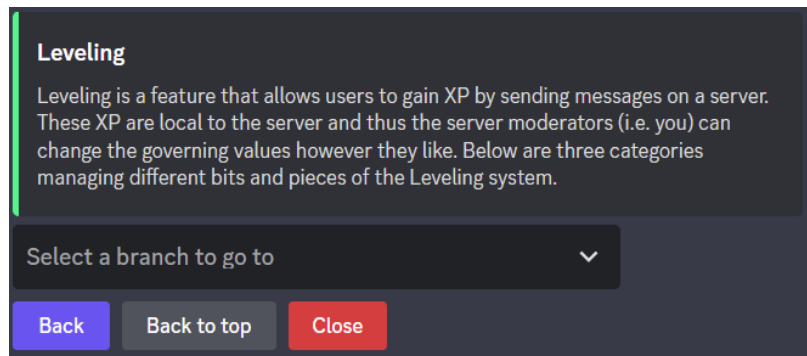
Note: This chapter does not show all possible configuration options. These options will be explained within their respective categories.

1.1.1 Traversing the config menu



To select a new branch, use the select menu and click on the branch you want. Each branch comes with a title and a *brief*. A brief is a short description of the branch's use.

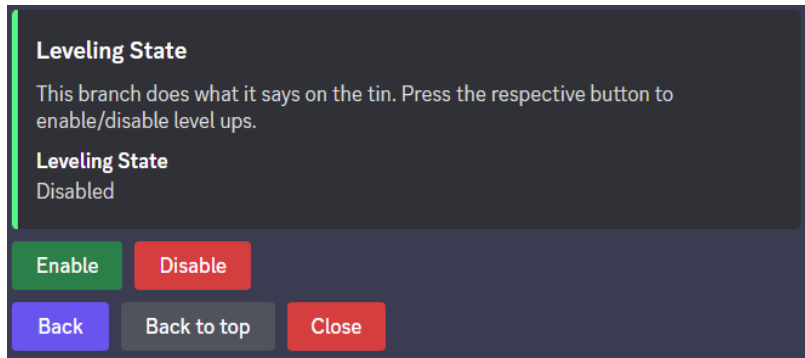
Note: Branches are not always equivalent to the categories of `/help`.



You may have noticed the buttons “Back” and “Back to top” not being activated in the main menu. These buttons allow you to travel up the configuration tree, allowing you to go back to a previous branch. “Back” sends you back to the previous higher branch, while “Back to top” sends you to the main menu.

1.1.2 Appearances of Branches and Elements

Every branch comes with a description, which is different from its brief. A description is usually more detailed and (if required) explain what the subbranches do.



Some branches do not have subbranches. Internally, these are called *elements*. Elements manage specific settings of the bot. Similarly to branches, elements have a description that instructs you on how to use them.

1.2 Fun

This category includes a mixed assortment of more or less useless commands to entertain yourself and others.

1.2.1 Cloning



This feature allows you to create a near-perfect copy of yourself. The clone will send the same messages you send across all channels where cloning is enabled. However, this feature is disabled by default, which means a server moderator will have to enable it first.

Managing the Cloning feature

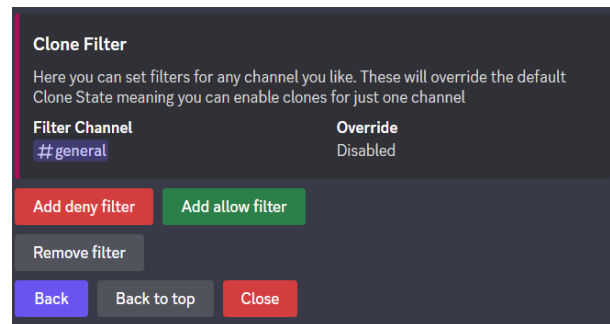
The cloning feature can be easily managed using `/config`. For help regarding this command, look [here](#).

Note: Changing the permissions of `/clone` will not work to configure the cloning system. This would only disable use of the command, however, a user's cloning state is changed per server, meaning members could go to a channel where the command is enabled and then use the feature in all channels, regardless of the command setting.

Global enabling and disabling

If you already know how the command works, go to `Clones>Clone State`. Here you will find two buttons “Enable” and “Disable”. These set the global cloning feature. If cloning is enabled, users can use the feature anywhere, so long as no filter overrides this setting.

Setting filters



Often times moderators want features like cloning to only be enabled in certain channels. This is implemented using so-called filters. These filters override the global setting, meaning you can allow use of clones in #spam, while it is disabled everywhere else.

To create such a filter use `/config` and navigate to `Clones>Clone Filter`. There you will find three buttons. The button “Add deny filter” allows you to disable cloning in a certain channel, “Add allow filter” allows you to enable it. Using “Remove filter” you can remove an existing filter.

Disabling cloning in one channel also renders `/clone` useless. Clones enabled outside the channel will not work within a channel that has disabled cloning.

Using cloning

To use the cloning feature, use `/clone` in a channel where cloning is enabled (either by the global state or filters) with `state: True`. Cloning is now enabled for you specifically on the current server. To disable the feature use `/clone` with `state: False`.

1.2.2 Tableshrug

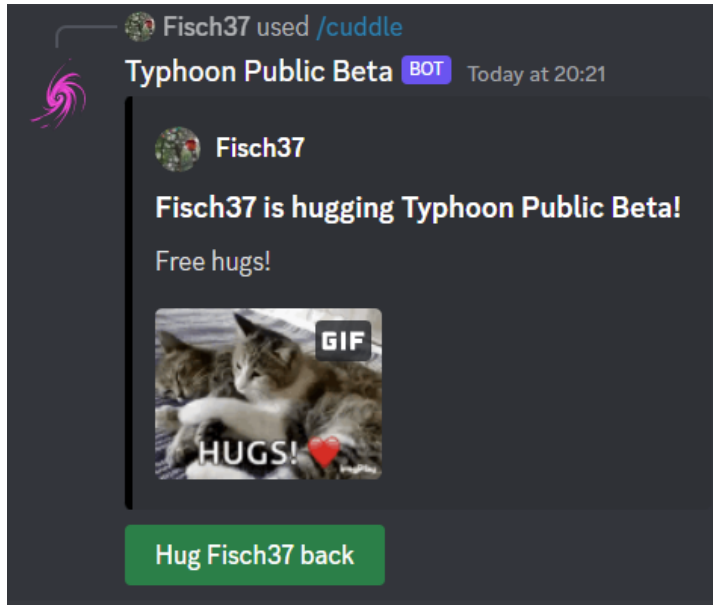
A tableshrug (¯_()_/) is a combination of /shrug and /tableflip. The /tableshrug command uses webhooks to allow you to send a message with a tableshrug attached at the end. Use it for when you are really mad but pretend to be perfectly calm... or just as a joke.

1.2.3 Passing the Burrito

Do you know that feeling when you just want to be nice to people and offer a burrito? No? That's too bad. But for everyone else: Use /passtheburrito <amount> to pass the burrito down.

1.2.4 Patting & Cuddling





There are some things in life that all people should enjoy. I think cuddles are one of them, which is why I made these two commands. Use `/patpat <target> [<comment>]` and `/cuddle <target> [<comment>]` to hand out pats and cuddles respectively.

You have to specify a server member as a target and can optionally add a comment that will be sent within the message.

The targeted member may use the button below the bot's post to respond with another cuddle/pat. This response cannot be directly responded to.

1.3 Leveling

The leveling system is a per-server activity-based reward system for members of your server. It needs to be enabled by a moderator, before being used, a guide for which you will find [here](#).

For *every** message sent, the member gains a randomised, configurable amount of experience points. To learn how to configure this amount, see [Managing XP Gains](#)

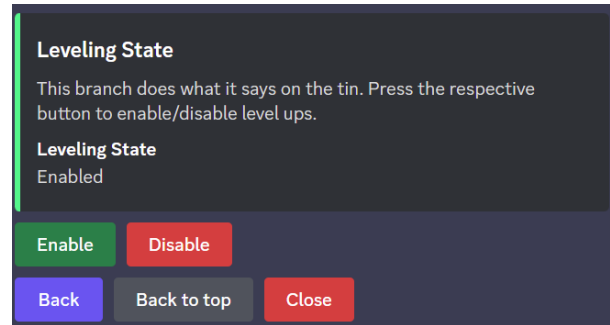
As soon as a member has a certain amount of XP, they will achieve a new *level*. Levels don't do anything, besides cause the bot to send congratulations and triggering *Reward Roles*.

The message sent by the bot upon a levelup may be configured to fit your needs. For that, see [Customised Level Up Messages](#).

A Reward Role is, as previously stated, a role a member will be granted automatically upon achieving a certain level. To configure a Reward Role, see [Setting up Reward Roles](#).

1.3.1 Enabling Leveling

Enabling leveling can be done by use of the `/config` command. If you are unfamiliar with the menu provided by it, see [this](#).



If you are familiar with it, head to *Leveling > Leveling State*. There you will find the two buttons “Enable” and “Disable”. These do exactly what they sound like. All you need to do now is click on “Enable” and confirm that leveling is enabled.

1.3.2 Viewing your and other’s XP

There are two commands that allow you to see xp statistics. The first is `/level [<user>]`, which allows you to see a server member’s experience points and level. This command can be executed without setting the user argument, in which case your own statistics will be shown. If the argument is set to a server member, that member’s statistics will be shown.

You may also want to see the statistics of everyone on a server at once. For this, there is `/leaderboard`. This command will give you a list of all members that have gained xp ordered by the xp they have earned.

1.3.3 Managing XP Gains

Managing xp gains is achieved by use of `/config`. If you are unfamiliar with this command, read [Configuring the Bot](#), otherwise continue reading.

Configuring Randomised Gains

The bot decides the amount of xp gained for a certain message by a random generator within two values you can configure yourself. These are the lower bound and upper bound. As you may think, the lower bound is the smallest amount of xp a user will get for a message and the upper bound is the largest possible amount a user will get for a single message.

The screenshot shows a dark-themed menu titled "XP Gain". Below the title, there is explanatory text: "The XP gain for a message is randomised according to two values. 1. is the Lower Bound. It is the minimal amount of XP a user can get per message. 2. Is the Upper Bound. It is the maximum amount of XP a user can get per message." Below this text, there are two columns of values: "Lower gain" with the value "15" and "Upper gain" with the value "25". At the bottom of the menu, there are six buttons arranged in two rows and three columns: "Set Lower Gain" (blue), "Reset Lower Gain" (grey), "Set Upper Gain" (blue), "Reset Upper Gain" (grey), "Back" (blue), and "Close" (red). A "Back to top" button is also present between the "Back" and "Close" buttons.

You may configure these in the config menu under *Leveling* > *XP Gain*. There you will find four buttons arranged in pairs of two. With the buttons in the first column you can set the relevant bound to a value of your choosing, while the second column buttons allow you to reset the bounds to their default values.

Note:

You may not choose a configuration in which the following conditions are violated:

1. The lower bound may not exceed the upper bound.
2. Neither of the bounds may be less than 0.
3. Both bounds must be whole numbers.
4. Neither of the bounds may exceed 255.

Any configuration violating one or more of these rules, will not be accepted.

Configuring Message Timeouts

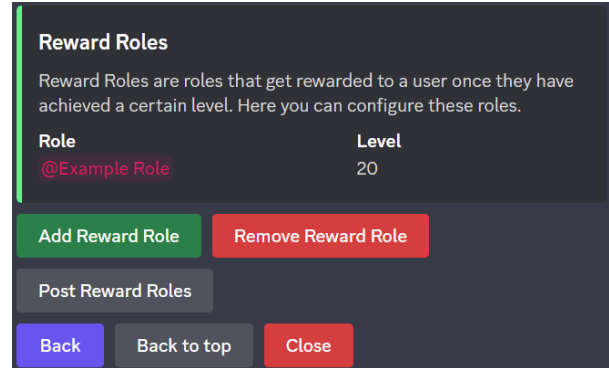
The message timeout is the distance in time between two messages of the same user that will cause an xp gain. If you are on a server with few people that do not usually spam, you may find it reasonable to decrease this value. Servers with many people might want to increase this value to aid against spamming.

The screenshot shows a dark-themed menu titled "Message Timeout". Below the title, there is explanatory text: "To prevent spamming, the Leveling feature only allows a user to gain XP a certain amount of time after the last gain. This value can be set to anything you like. Setting it to 0 will allow users to gain XP without any timeout. Do consider though, that messages deleted by the Automod will still count towards the user's XP." Below this text, there is a label "XP Timeout" followed by the value "0H 1M 0S". At the bottom of the menu, there are four buttons arranged in two rows and two columns: "Set Timeout" (blue), "Reset Timeout" (grey), "Back" (blue), and "Close" (red). A "Back to top" button is also present between the "Back" and "Close" buttons.

To configure this option go to *Leveling* > *Message Timeout* of the `/config` command. There you will find two buttons, one allowing you to set a timeout, the other one resetting it to the default value.

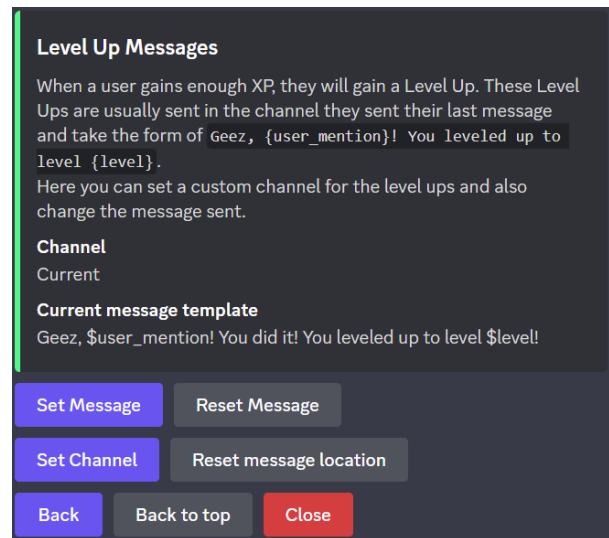
A timeout may be specified using the following format: `<hours>H<minutes>M<seconds>S` This means a timeout of 1 hour 23 minutes and 4 seconds would be sent as the following: `1H23M4S` You may choose not to pass a certain time unit, meaning `5M` is completely valid despite it missing hour and second declarations. Furthermore, minutes and seconds are not limited by their usual value range, allowing `600S` instead of `10M`.

1.3.4 Setting up Reward Roles



Setting up a Reward Role works using `/config` by heading to *Leveling > Reward Roles*. The buttons “Add Reward Role” and “Remove Reward Role” allow you to add or remove a reward role. The third option “Post Reward Roles” will post a message describing **all** reward roles publicly in the current channel.

1.3.5 Customised Level Up Messages



Using `/config` you can customise the message sent when a user receives a levelup (i.e. increasing in level). You can also customise the channel where the level up message is sent. This is set to “Current” by default, which means that the message will be sent in the channel of the message that caused the levelup.

The branch required is *Leveling > Level Up Messages* and features two rows with two buttons.

The first row

The second row allows you to set the channel. If you wish to set it to “Current”, use “Reset message location”.

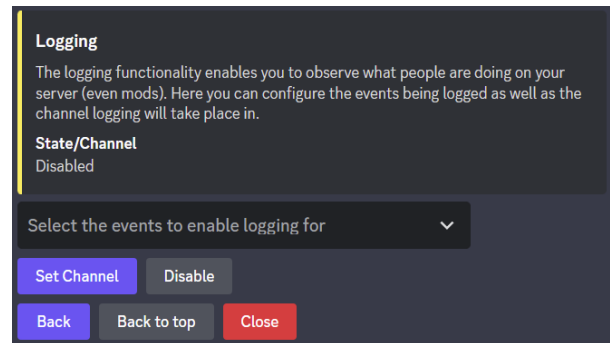
The first row allows you to customise the level up message. This is done using a *template*. This is a generalised expression of the message that will be applied for every message. A template may feature variables, which are replaced by their respective values for messages being sent. A variable must be preceded by a \$. You can still use \$ without issues, just be careful not to write variables. The following variables are accepted:

Variable Name	Description
user_name	The username. This does not change with the nickname and does not mention the user.
user_nick	The nickname of a user. This may change with the server and does not mention the user.
user_mention	This is the mention of the user.
server_name	The name of the current server.
server_members	The amount of members on the server.
level	The new level of the user.
xp	The xp of the user.
rank	The rank of the user on the leaderboard.
channel	The channel the user has sent the message in that caused the levelup. This is not necessarily the channel the message appears in.
send_channel	The channel the levelup message will be sent in

1.4 Moderation

1.4.1 Logging

The logging system is an automatised event watcher, that keeps track of events occuring on your server. Events are to be understood not as the Discord Events available on community servers, but instead as changes to the server or message history potentially relevant for moderation.



Setting up logging can be done using `/config`. If you are unfamiliar with this command, see [Configuring the Bot](#). If you are familiar with it, head to `Moderation>Logging`. First you will need to set a channel for any events to be logged in. Use the “Set Channel” button to achieve this. Then use the select menu above the buttons to select the events you wish to log. The following event categories are supported:

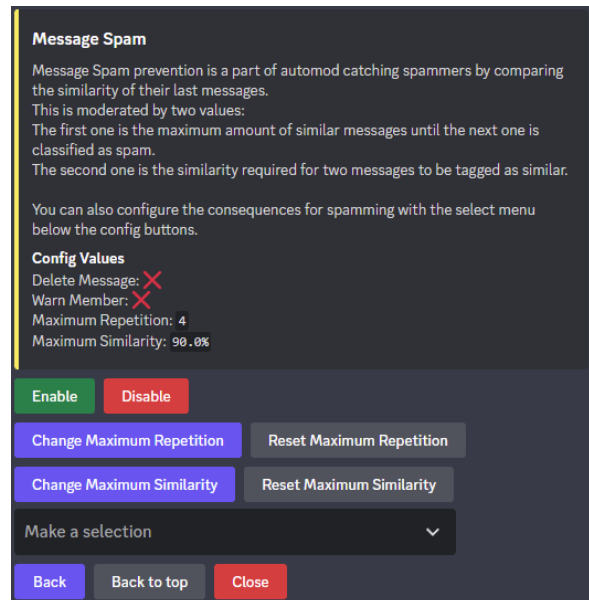
- Moderation: Logging of Automod or member kicks/bans
- Channels: Log changes to channels (creation, deletion, etc.)
- Server Updates: Log changes regarding the server (e.g. name, security, emojis/stickers)
- Invites: Log creation/deletion of invites
- Member Changes: Log changes to the members (joins, leaves, profile changes)
- Messages: Log message changes (deletion/edits)
- Reaction Moderation: Log moderation actions related to reactions
- Roles: Log changes to the roles of this server
- Thread Moderation: Log changes to threads

If you want to disable logging, simply click the “Disable” button next to “Set Channel”

1.4.2 Auto Moderation

Automatic Moderation (short: Automod) is a feature allowing Typhoon to moderate common issues taking some load off the moderators. This feature includes moderation of caps spam (e.g. HELLO! READ MY MESSAGES!), regular spam (i.e. repeatedly sending a similar message), and emoji spam (e.g.). This feature can be configured using `/config`. If you are unfamiliar with the workings of this command, see [Configuring the Bot](#).

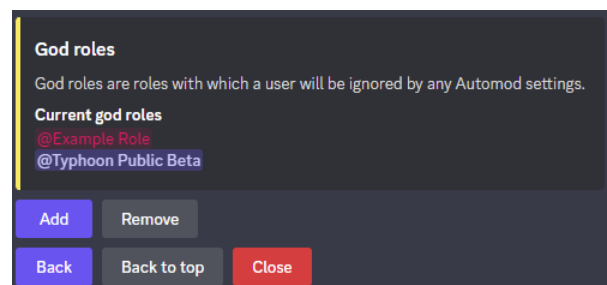
Setting it up



Head to `/config`'s *Moderation>Automod* and select the option you want to configure. (If you want to configure multiple, repeat this with the relevant feature). Click the button “Enable” and select “Delete Message” in the select menu.

Boom! Now the feature of your choice is enabled with the default configuration. If you want to change when a message is detected, read the instructions on the relevant config branch.

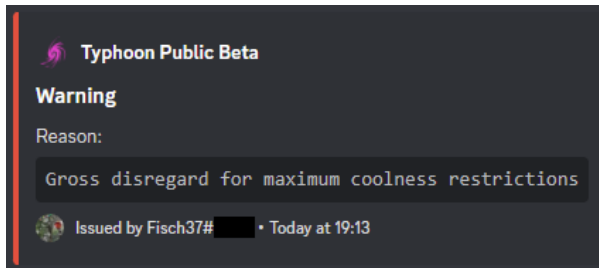
Immunity roles



Sometimes you want certain people (like moderators) to not be affected by Automod. For this, Typhoon offers “God Roles”. Any member with at least one God Role will not be considered for Automod. To add or remove God Roles head to *Moderation>God roles* of `/config`. To add a new role, use the “Add” button, to remove a role, use “Remove”.

1.4.3 Warning System

Warnings are special events intended to serve as a notice to the moderators. There are two things that may trigger a warning: Automod features with the “Warn Member” consequence enabled and the command `/warn`.



A warning will be logged in the logging channel given that the “Automod” event category is enabled. For more information on this, see [Logging](#). It features the targeted member, an optional reason, an author notice, and a date.

You may hand out warnings manually using `/warn <target> [<reason>]` as was mentioned previously. With this command a warning is created authored by you addressed at the target with an optional reason. It will be timestamped to the current time.

1.4.4 Locking Channels

Is the situation really escalating in a certain channel? Do you feel like your moderators aren’t being listened to? In this case, you might want to temporarily remove everyone’s write permissions in that channel.

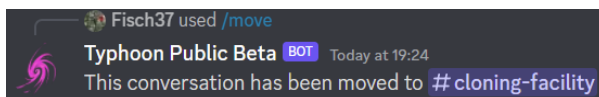
This is what `/lock` achieves with its subcommands `/lock set` and `/lock remove`. The bot will also send an announcement in that channel to notify of the lock.

Note: This feature is a shorthand for removing the Send Messages permission from `@everyone`. It will not work when `@everyone` did not have this permission to begin with.

1.4.5 Moving Messages

This feature allows you to move the last `n` messages of the current channel to another channel. It has one command `/move <channel> <messages>`. You may move up to 100 messages at a time. Moved messages will be grouped together into as few messages as possible.

Warning: There is no support for images, embeds or videos in this feature. If you attempt to move a message that has this, the relevant attachment **will** be deleted. (This does not apply to links, these will still work as expected.)





1.4.6 Mass-deleting Messages

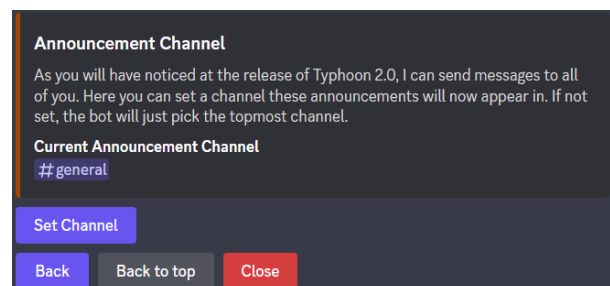
If you find yourself overwhelmed by spam, you might wish to delete a lot of messages at once. `/purge <messages>` allows you to delete the last `n` messages permanently. The maximum messages that will be deleted at once is 100.

1.5 Utility

1.5.1 Bot Announcements

With this version of the bot, you may be notified of important changes via a message. Announcements will be sent in the *Announcement Channel* or the first channel in the order.

Note: The Announcement Channel is unrelated to the Discord channel type “Announcement”.



If you wish to set a custom channel for announcements to appear in, you can use the `/config` command to do that. If you are unfamiliar with the `/config` command, read *Configuring the Bot*, otherwise continue reading.

If you are familiar with the command, use the *Announcement Channel* branch and use the `Set Channel` button to change the Announcement Channel.

1.5.2 Inviting the Bot

Inviting the bot is quite simple. Just use `/invite` and click the link in the reply. On new versions of Discord applications, this will open in the application. Older versions will open the link in a browser.

1.5.3 Rolling dice

Rolling dice is done via the command `/roll <die> [<sort>]`.

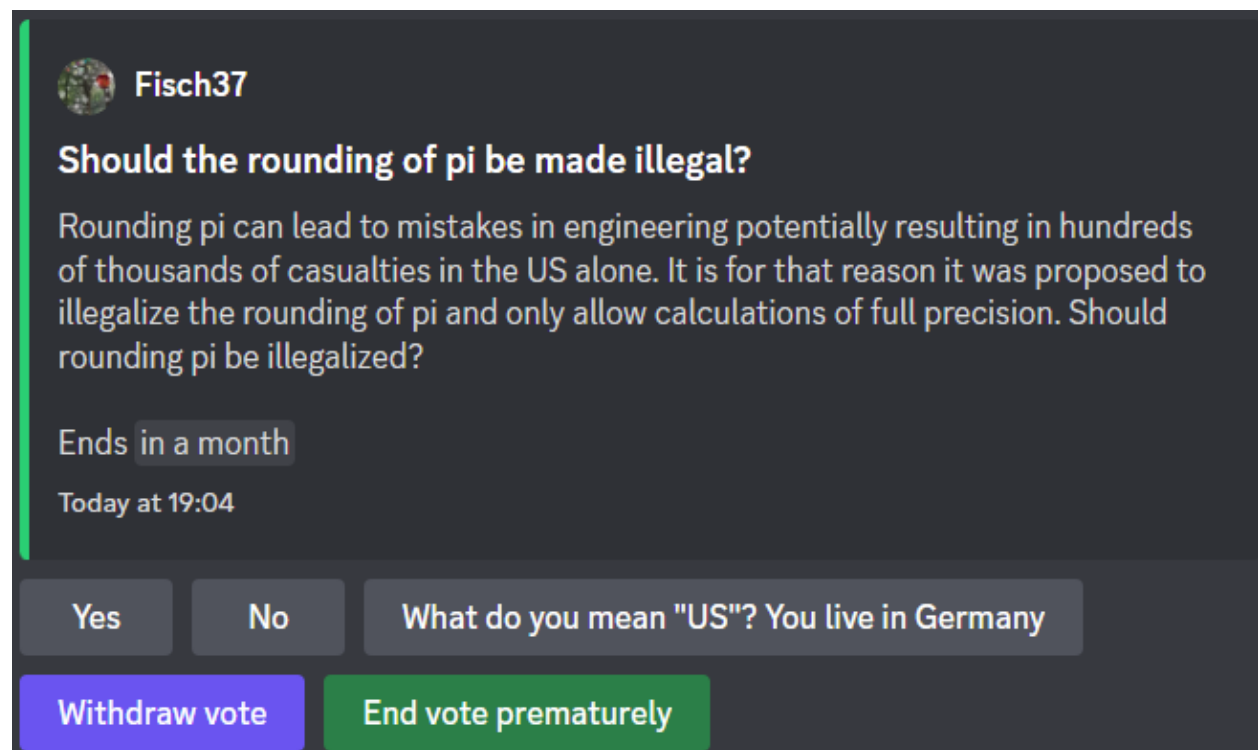
The *die* argument is expressed in the format of `<n>d<f>`, where *n* is the amount of dice and *f* is the highest possible value. The dice is implied to have 1 as a lowest possible value. You may also have multiple dice of this format, which will need to be separated by `+`. The *n*-value is capped collectively at 100, *f* is capped at 500. The maximum dice types you can chain is 10.

The sort argument is an optional argument and will sort the results in decreasing order.

This is an example result of `/roll 10d8+5d6 sort: true`:

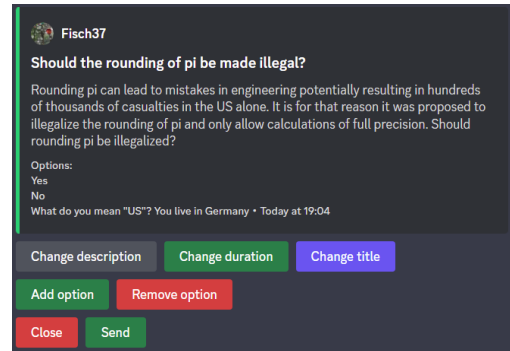
```
10d8: 7, 7, 5, 5, 3, 2, 2, 1, 1, 1 Sum: 34
5d6: 6, 5, 5, 5, 3 Sum: 24
```

1.5.4 Voting



The screenshot shows a Discord poll interface. At the top, the user's profile picture and name "Fisch37" are displayed. The poll title is "Should the rounding of pi be made illegal?". The poll description reads: "Rounding pi can lead to mistakes in engineering potentially resulting in hundreds of thousands of casualties in the US alone. It is for that reason it was proposed to illegalize the rounding of pi and only allow calculations of full precision. Should rounding pi be illegalized?". Below the description, it says "Ends in a month" and "Today at 19:04". At the bottom, there are four buttons: "Yes" (light blue), "No" (light blue), "What do you mean 'US'? You live in Germany" (light blue), and "Withdraw vote" (orange). There is also a green button labeled "End vote prematurely".

The voting feature allows you to perform anonymous polls of your community.



To create a vote use `/vote create` and follow the instructions. After passing the initial *Modal*, you will see an editor, where you can change the options further. Here you will also need to specify the options.

The author of a vote also has the option to end it before the duration. While this button is visible to everyone, it won't work for anyone but the author.

After the vote is completed, the results will appear in a footer of the embed.

1.5.5 Scheduled Messages

This feature allows you to schedule messages to be sent at a set point in the future. The feature works by the command `/schedule_msg <time> <content>`.

The `time` argument should to be of the format `YYYY.MM.DD;HH:MM:SS`. Instead of `.` you may also use one of the following separators: `"-"` `"."` `"/"` `":"` and `" "`.

Note: Due to an API limitation all times need to be given in Universal Coordinated Time (UTC) or Greenwich Mean Time (GMT). Not obeying this will result in your message schedules being offset by up to 12 hours depending on your timezone.

1.5.6 Anonymous Messages

You might want your users to be able to send messages without their name being visible. For this, `/send_anon <message>` will work great. The command works as simple as you might think. Simply write the message you want into the message argument and see the message pop up.

If you want to restrict or remove access to this feature, head into the integrations settings of your server and remove access to the command.

SETTING UP YOUR OWN VERSION OF TYPHOON

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`