

# botrac3r help guide

Your complete guide to all things botrac3r.

## DON'T PANIC!

Just relax and command the bot.

## NEED HELP?

Simply DM me on Discord.

@cloudrac3r#5547

## About botrac3r

botrac3r is a Discord bot written in node.js (<http://nodejs.org/>) and intended for use on the Epicord server. Its function is mainly to run commands that Epibot does not have, or to improve the ones that it does.

## How to use this guide

Instructions are written in this black font. Information that you should type is in this monospace font. *I seperate various other things with italics.*

If you haven't used botrac3r before, read the *General command* syntax to find out how to use commands, then read the *Command reference* to see what is available. For details on the syntax of each command, and usage examples, scroll down until you see *List of commands and usage*.

## General command syntax

All botrac3r commands start with `..` botrac3r will only process information that starts with two dots.

After the `..` comes the command name. This tells botrac3r what you want to do.

After the command name comes a semicolon `;` This tells botrac3r where the end of the command name is.

After the semicolon `;` you may need to type extra information to give to the command to either control the way that it works or to give it data. For example, the temp command needs the temperature to convert (such as `..temp;40`), and the vote command needs to be told whether to open a new poll or vote on an existing one (such as `..vote;add`). These are known as *arguments*. To separate several arguments, use another semicolon (such as `..roll;1;6;Dice`).

If a command is given too many arguments, it will simply ignore the extra ones. However if it is not given enough arguments, it will either not function at all, function incorrectly or show an error message with the correct usage.

## Quick help

If you're not sure how to use a command, and you just need a quick refresher of the available choices for an argument instead of a full tutorial from this handbook, replace the argument that you're not sure about with `help`. Usually, botrac3r will give you a list of the available options.

For example, if you need help with voting on a poll, and you can't remember what to type after `..vote;add;` then you should enter `..vote;add;help`. Simple!

If you don't get anything help-related, DM me (cloudrac3r#5547) on Discord and tell me which command you tried to run, and that you want the quick help to be created. I'll do it soon enough. In the meanwhile, check this manual.

## List of commands and usage

### **`..roll;low;high;message`**

The roll command is used to choose a random number. The lowest number that can be picked can be set with *low* and the highest number will be *high*. You may also include a *message* which describes the purpose of the number that you will get.

### Examples

```
SAMPLE INPUT  ..roll;1;6;Dice roll
SAMPLE OUTPUT @you Dice roll: 6
SAMPLE INPUT  ..roll;120;180;BPM for new song
SAMPLE OUTPUT @you BPM for new song: 141
```

### **`..yn;message`**

The *yn* command (short for yes/no) is used to choose either *yes* or *no*. Many Epicord users may enjoy using this to decide if they should go to sleep or not. You may like to include the *message* argument to show the purpose of running the command.

If you currently use EpiBot's `~8ball` command for this purpose, you'll be pleased to know that *yn* always produces an answer, whereas EpiBot likes to choose answers such as "I don't know m8" and, my personal favourite, ">///<". Of course

you could use `~8ball strict` instead, but that didn't exist when this command was created. Besides, `..yn;` is faster to type.

## Example

```
SAMPLE INPUT  ..yn;Sleep?
```

```
SAMPLE OUTPUT @you asked `Sleep?`, I decided: Hmm, probably not...
```

## **`..temp;temperature`**

This command lets you easily convert temperatures between degrees Celsius and Fahrenheit. You should specify the temperature that you want to convert as *temperature*.

All the numbers that are output by botrac3r are rounded to 1 decimal place – however, the calculations are done with unrounded numbers.

Note that you do not need to say whether your input is in Celsius or Fahrenheit – botrac3r will perform both calculations for you. Just make sure that you look at the correct part of the output!

## Examples

The first example might be used to convert a weather forecast in °F to °C. The second example might be used to convert a cooking recipe in °C into °F.

```
SAMPLE INPUT  ..temp;53
```

```
SAMPLE OUTPUT @you 53°F = 11.7°C, and 53°C = 127.4°F!
```

```
SAMPLE INPUT  ..temp;180
```

```
SAMPLE OUTPUT @you 180°F = 82.2°C, and 180°C = 356°F!
```

## **`..proxy;channelID;message`**

Warning: If you're planning on using this command, it's not very easy. If you have any ideas for improvements, DM me (cloudrac3r#5547) with your feedback. Note that I, as the author of botrac3r, can check its logs to see who sent the command, even in DMs to botrac3r, so I can still report any inappropriate usage.

This command will let you make botrac3r send messages on your behalf. If you've ever wanted to post a message on a server, but not let anyone know who it came from, then this is the command for you. To hide the origin of the message completely you must send this command in a DM.

The *channelID* is the numerical ID of the channel that you want botrac3r to send the message to. You can find this ID by enabling *Developer tools* in the Discord settings, then right-clicking a channel name. Obviously, botrac3r must be on that server, and must have permission to send messages in the channel.

*message* is the message that you want to send in the channel.

## Example

```
SAMPLE INPUT  ..proxy;123456789;I believe in ghosts!
SAMPLE OUTPUT Message sent via proxy: ``I believe in ghosts!``
```

### **..vote;**

The vote command lets you create and vote on polls, which is useful for getting an organised collection of people's opinions without needing to check back through many messages and keep a mental tally.

After typing `..vote;` you must then tell botrac3r whether you want to create a new poll, show details of existing polls, or vote on an existing poll.

### **..vote;start;pollName;choice1;choice2[;choice3[;...]]**

This command creates a new poll.

Choose the name of the poll, *pollName*. This should be the question that is being asked. For example, *What should we do now?*

Think about the choices that you want. There must be at least 2 (obviously!). Then, drop them into the places marked by *choice1*, *choice2*, etc. If you have more than 3 choices, just continue adding more choices separated by semicolons.

### **..vote;check**

This command shows a list of all the polls that have been started.

You will see a list of the names of the polls. Before each name, you will see the ID of the poll in bold. You will need this if you want to vote or to check votes. After the name of the poll, you will see a username in brackets. This is the person who created the poll.

## **..*vote;check;pollID***

This command shows details about one poll which has been started.

You can find the *pollID* when you show the list of polls – it is the number before the name of the poll.

You will see a list of the available choices. Before each choice, there will be a number in bold. This is the ID of the choice. After each choice, you will see a number in brackets. This shows the number of people who have voted for that choice.

## **..*vote;add;pollID;choiceID***

This command lets you vote on one choice of an existing poll.

You will need the ID of the poll and the ID of the choice that you want to vote for. These can be found by using the 2 commands above. After you successfully vote on a poll, you cannot vote again, take back your vote or change your vote. If you chose the wrong option... well, hopefully you look more carefully and choose the correct one next time.

Note: Everyone on the server is allowed to vote, including the creator. Every channel on every server uses the same polls database.

## **This is too complicated!**

DM me (cloudrac3r#5547) and tell me what you would like to change about the command.

## **Examples**

These examples behave as if all the commands were typed by the same person in the order shown. If you were to enter these commands in this order, you would get the same output.

```
SAMPLE INPUT  ..vote;start;What movie should I watch?;Blackadder;Harry Potter;Matrix
SAMPLE OUTPUT Poll `What movie should I watch` started with the options
                `Blackadder`, `Harry Potter`, `Matrix`!
SAMPLE INPUT  ..vote;check
SAMPLE OUTPUT Ongoing polls:
                1: When should I go to bed? (@someone)
                2: What movie should I watch? (@you)
SAMPLE INPUT  ..vote;check;2
SAMPLE OUTPUT Poll: What movie should I watch? (@you)
                1: Blackadder (0)
                2: Harry Potter (0)
                3: Matrix (0)
                To vote, type ..vote;add;2;choiceNumber
SAMPLE INPUT  ..vote;add;2;3
SAMPLE OUTPUT Your vote was cast - type ..vote;check;2 to see it!
SAMPLE INPUT  ..vote;check;2
SAMPLE OUTPUT Poll: What movie should I watch? (@you)
                1: Blackadder (0)
                2: Harry Potter (0)
                3: Matrix (1)
                To vote, type ..vote;add;2;choiceNumber
SAMPLE INPUT  ..vote;add;2;1
SAMPLE OUTPUT You've already voted on this poll.
```

## **..wwg;**

This controls One Night Ultimate Werewolf (often abbreviated ONUW or WWG), which is a real life card game but can also be played on Discord. This command was requested by Master Wonder Mage. DM me (cloudrac3r#5547) if you want to suggest your own command.

Before you begin, you should thoroughly read the rules of the game, and make sure you understand how it is played before you commit to joining and possibly messing up a game. The rules for the basic game can be found at:

<http://fully-faltoo.com/uploads/werewolf.pdf>

## **..wwg;explain**

Shows a quick explanation of the available commands and a link to the rules of the game.

## **..wwg;join**

Join the game. When the game is started, you will take part.

## **..wwg;leave**

Leave the game. When the game is started, you will not take part.

## **..wwg;check**

Show details of the game which will be started, such as players that have joined, which cards will be used, and whether the game has the correct conditions to start it.

## **..wwg;start**

Start the game. The subcommands *join*, *leave* and *start* will be unavailable. The subcommand *check* will change slightly in behaviour. botrac3r will assign roles to all the players in the game, DM each player their roles, deal the centre cards, and then act out the role of the announcer.

As the announcer, it will prompt players with possible actions, wait for their response and perform the correct actions in the correct order.

## **..wwg;end**

The game time must be measured by one of the players. After the agreed time has elapsed, one of the players must enter this command to end the game. After that happens, everyone will be prompted to vote either via DM or in the main channel on which player to kill. After everyone has voted, the results and the deaths will be displayed, along with everyone's starting and ending roles so that you can work out what actually happened, without needing to sort out what was a bluff and what wasn't.

## **..wwg;interact**

This command is used during the night phase to perform the night actions. botrac3r will automatically prompt you via DM when the night begins of the correct command syntax, because it varies between roles. For example, it is used by the seer to tell botrac3r which cards should be viewed, or by the robber to select a player to steal from.



## **..lenny;**

This one is a bit different from the other commands because it can appear anywhere in the message. If botrac3r finds `..lenny;` in a message, it will store your message contents, replace every occurrence of `..lenny;` with `(^◡^)`, prepend your username (not nickname), then send it as a new message. The message that you sent will be deleted if botrac3r has the *Manage messages* permission on that server.

```
SAMPLE INPUT  ..lenny;
```

```
SAMPLE OUTPUT you: (^◡^)
```

```
SAMPLE INPUT  Wait, you never watched Star Wars? ..lenny;
```

```
SAMPLE OUTPUT you: Wait, you never watched Star Wars? (^◡^)
```

## **..shrug;**

Identical to `..lenny;` except that `..shrug;` is replaced with `¯_(ツ)_/¯`.

## **..flag;emoji1;emoji2;emoji3**

Use the three emojis supplied (can be either built-in or custom) to create a small model of the US flag. You should supply the emojis in the order *red*, *white*, *blue*. You must manually remove whitespace between the `;` of the command separator and the `:` of the emoji. However, whitespace around custom emojis will be removed automatically.

## **..time;**

The `..time;` command is used to see a person's local time, if they have set it up. To see someone's local time, use `..time; @mention`. To set your own time so that others can check it, use `..time;set;timezone` where *timezone* is a TZ-formatted location (a list can be found at <http://bit.do/tzlist>).

```
SAMPLE INPUT  ..time;set;Pacific/Auckland
```

```
SAMPLE OUTPUT @you Your local time was set to 18:41 (24hr)
```

```
SAMPLE INPUT  ..time; @cloudrac3r
```

```
SAMPLE OUTPUT @you cloudrac3r's local time is 18:41 (24hr)
```

## **..choose;choice1;choice2[;choice3...]**

Randomly choose an item from the semicolon-seperated list that follows ..choose;

```
SAMPLE INPUT  ..choose;apples;bananas;grapes
```

```
SAMPLE OUTPUT @you I choose: apples
```

## **..8hippo;[message]**

Collects a list of all the custom emoji on the server which contain *hippo* in their names, then sends the supplied message (or an amusing default if no message was given) to the same channel, followed by a randomly chosen emoji from the list.