
zbot Documentation

Release 3.6.6

ZRunner

Jul 04, 2020

List of available sections:

1	Permissions	3
1.1	General Permissions	3
1.2	Text Permissions	4
1.3	Voice Permissions	6
2	Moderation	7
2.1	Warn	7
2.2	Mute	7
2.3	Slowmode	8
2.4	Clear	8
2.5	Kick	9
2.6	Softban	9
2.7	Ban/Unban	9
2.8	Banlist	10
2.9	Handling cases	10
2.10	Anti-raid	11
2.11	Anti-bot verification	11
2.12	Miscellaneous	12
3	Server configuration	15
3.1	Config options	15
3.2	XP System	18
3.3	Partners system	19
3.4	Server backup	20
4	User configuration	23
4.1	XP system	23
4.2	Your info	24
5	Information	27
5.1	About	27
5.2	Bot invite	27
5.3	Help	27
5.4	Info	28
5.5	Membercount	28
5.6	Permissions	28
5.7	Ping	29

5.8	Prefix	29
5.9	Usernames history	29
5.10	Welcome message	29
6	Minecraft	31
6.1	MC	31
6.2	Mojang	32
7	Rss	33
7.1	See the last post	33
7.2	Follow a feed	33
7.3	See every feed	34
7.4	Mention a role	34
7.5	Change the text	34
7.6	Move a feed	35
7.7	Setup a feed embed	35
7.8	Delete a followed feed	35
7.9	Reload every feed	35
8	Roles reactions	37
8.1	Add and remove a reaction	37
8.2	Get or leave a role	37
8.3	List every roles-reactions	38
8.4	Update your embed	38
9	Fun	39
9.1	List of every command	39
9.2	Bot events	45
10	Miscellaneous	47
10.1	Bitly urls	47
10.2	Book	47
10.3	Changelogs	48
10.4	Embed	48
10.5	Hour & Weather	48
10.6	Markdown	48
10.7	Reminders	49
10.8	Say	49
10.9	Tic-tac-toe	50
10.10	Vote	50
11	Indices and tables	51

Welcome to the **ZBot** bot documentation, a Discord bot coded in Python 3 by a small French developer, *Z_runner*. You will find in the documentation all the explanations on each of the usable commands, as well as the list of the last additions and some code examples. The documentation is currently in English but the bot is available in several languages, including its main language, French.

The *Moderation* section displays every command related to the moderation of a server: clear, slowmode, kick, ban... The *Minecraft* section allows you to discover all Minecraft related commands: item search or server check. The *Info* section is about commands that allow you to get information about members, roles, channels, server etc. The *Configuration* sections are a guide to allow you to correctly configure the bot on your server, or for yourself as user. The *Permissions* page allows you to put exactly the right permissions in the bot, nothing unnecessary or dangerous! *RSS* section is there to follow websites, YouTube/Twitter/Twitch channels and other feeds, without worrying about it. And other sections are continually being added!

<p>Warning: The bot and documentation are still in progress. This means that not all the commands in the bot are necessarily present in the doc, and vice versa!</p>

If you want to invite this bot to a server, [just click here!](#) And to join our official support server, [click on this link!](#)

The permissions given to members is an important part in the configuration of a server. The same is also true for bots. This page is intended to show you each of the permissions necessary for the bot, as well as explain why they are necessary. All this in order to avoid putting unnecessary permissions on the bot, which it turn will keep your server clean and safe.

Warning: Never *never* **never NEVER** never (*yes, 5 times never*) put a bot with administration permissions. It has already happened once that the bot's security key is stolen, which allows the thief to take full control of the bot, such as deleting your channels or banning your members. Even though safety has been completely redesigned since this incident, zero risk is not possible. Even bots like Mee6 are not immune from carelessness (as a MEE6 staff, I know what I'm saying).

1.1 General Permissions

1.1.1 View Audit Log

Allows the bot to read server logs (adding roles, changing names, editing channels...). Not necessary for the moment

1.1.2 Manage Server

Allows the bot to change the name, image and region of the server, or get the list of all invites. Used for: [invite](#)

1.1.3 Manage Roles

Allows the bot to create and delete roles, or edit the permissions of roles lower than his own, and to give them to other members. Used for: [mute](#)

1.1.4 Manage Channels

Allows the bot to create, delete and modify channels (create invitations for example). Used for: [membercounter](#) option

1.1.5 Kick Members

Allows the bot to eject a member from the server. Used for: [kick](#)

1.1.6 Ban Members

Allows the bot to ban or unban a member from the server, as well as to consult the list of banned members. Used for: [ban](#) , [unban](#), [banlist](#), [softban](#)

1.1.7 Create Instant Invite

Allows the bot to create invitations to any visible room, without being able to modify or delete them. Used for:

1.1.8 Change Nickname

Allows the bot to change your own nickname. Not used at this moment.

1.1.9 Manage Nickname

Allows the bot to change the nickname of any member hierarchically equal or inferior to you. Not used.

1.1.10 Manage Emojis

Allows the bot to add, rename or delete emojis from the server. Used for: [emoji](#)

1.1.11 Manage Webhooks

Allows the bot to read, add, modify or delete [webhooks](#) . Used for: [infos](#)

1.1.12 Read Text Channels & See Voice Channels

Allows the bot to see chats and voice channels. This permission does not allow you to write in these chats or connect to the voice channels. Required for the bot.

1.2 Text Permissions

1.2.1 Read Messages

Allows the bot to read messages from a chat, but not the history. In other words, the bot will react to your messages but will not be able to read them again. Remove this permission in a channel to prevent the bot from being there.

1.2.2 Send Messages

Allows the bot you to write messages in written rooms. Required for almost all functionalities, but not necessarily for all channels.

1.2.3 Send TTS Messages

Allows the bot to send a TTS message, i.e. a message that will be read aloud by your application. No need for the bot.

1.2.4 Manage Messages

Allows the bot to pin or delete any message. Used for: `mute` , `freeze` , `clear` , `purge` , `fun commands`

1.2.5 Embed Links

Allows the bot the bot to send an embed. Used for: `membercount` , `mojang`, `XP system` . Required for: `infos` , `mc` , `config see`, a few `rss commands`, `embeds generator`

1.2.6 Attach Files

Allows the bot to send files (such as images) in a channel. Used for: `fun commands`, `XP cards`

1.2.7 Read Message History

Allows the bot to read the history of all messages in a channel. Used for: `clear` , `purge` , `fun commands`

1.2.8 Mention Everyone

Allows the bot to mention the `@everyone` role (which results in sending a notification to all members with access to the channel) or `@here` (sends a notification to all connected members with access to the channel). It is recommended to disable it.

1.2.9 Use External Emojis

Allows the bot to use emojis from any other server. The bot uses them in many situations to diversify emotions, so it is strongly recommended to keep it activated.

1.2.10 Add Reactions

Allows the bot you to add reactions to a message, whether they are Discord or server emotions. Used for: `react`, `vote command` and `poll channels`

1.3 Voice Permissions

1.3.1 Connect

Allows the bot to connect in this voice channel. It is also required to edit this channel. Used for: `membercounter` option

1.3.2 Speak

Allows the bot to speak in a voice chat room. No use for the moment.

CHAPTER 2

Moderation

Like many Discord bots, ZBot allows you to moderate your server in different ways. You will find the classic commands to delete messages, mute, kick out or ban a member, as well as being able to slow down or freeze a chat completely.

Among the features in preparation you will find the members' records as well as the possibility of sending warnings, or a section on automatic moderation.

Note: Like most of the features of this bot, the Moderation section is constantly being developed. Feel free to help us by offering suggestions, voting for the best ideas or reporting bugs at our [Discord server!](#)

Warning: Most of these commands are reserved for certain roles only. To allow roles to use a command, see the [config](#) command

2.1 Warn

Syntax: warn <user> <message>:

This command allows you to warn a member, without really sanctioning him. This member will receive this warning by personal message (if they have not disabled them), and the warning will be stored in his logs.

2.2 Mute

Syntax: mute <user> [duration] [reason]

This command mutes a member, preventing them from typing.

The principle is to assign the *muted* role to the member, in order to distinguish him from the others. Simply configure the server permissions to have the “send messages” option disabled. And even if you don’t, the bot will delete messages from recalcitrant mute members!

The duration of the tempmute is quite flexible: use `XXd` for days, `XXh` for hours and `XXm` for minutes (replacing **XX** by the corresponding number, of course!)

Warning: The muted role must be placed below the bot role, and the bot must have “Manage roles” (to give the role) and “Manage messages” (to delete messages) permissions.

2.3 Slowmode

Syntax: `slowmode <seconds>` or `slowmode off`

Slowmode keeps your text channel quiet when excited people have decided to talk a little too fast. More precisely, it prevents members from posting messages too often. The frequency between two consecutive messages from the same member is indicated in the command.

Note: The system uses a brand new feature released on September 8th in Discord beta. It therefore is a completely new (as in very few bots have it) feature and can be highly integrated into your applications. It is even better than just deleting messages.

2.4 Clear

Syntax: `clear <number> [parameters]`

This command allows you to efficiently delete messages, with a list of possible parameters for more accuracy. You can thus specify a list of members to check by mentioning them, `+i` to delete all messages containing files/images, `+l` for those containing links or Discord invitations, `+p` for pinned messages. By default, the bot will not delete pinned messages.

Be careful, all specified settings must be validated for the message to be deleted. For example, if you enter `clear 10 @Z_runner#7515 +i`, the bot will check in the last ten messages if the message comes from `Z_runner#7515` AND if the message contains an image.

If you enter `clear 25 -p +l`, the bot will clear the last 25 messages if they contain a link AND if they’re not pinned, no matter the author.

If you enter `clear 13 -p -i @Z_runner#7515`, the bot will clear the last 13 messages if they are not pinned AND if they do not contain any file/image AND if the author is `Z_runner#7515`.

If you enter `clear 1000 @Z_runner#7515 @ZBot_beta#4940`, the bot will delete all messages contained in the last 1000 messages of the channel AND written by `Z_runner#7515` OR `ZBot_beta#4940`

Warning: The permissions “Manage messages” and “Read messages history” are required.

2.5 Kick

Syntax: `kick <user> [reason]`

The kick allows you to eject a member from your server. This member will receive a personal message from the bot to alert him of his expulsion, with the reason for the kick if it's specified. It is not possible to cancel a kick. The only way to get a member back is to send him an invitation (see the `invite` command) via another server.

Warning: For the command to succeed, the bot must have “Kick members” permissions and be placed higher than the highest role of that member.

2.6 Softban

Syntax: `softban <user> [reason]`

This command allows you to expel a member from your server, such as kick. But in addition, it will delete all messages posted by this member during the last 7 days. This is what explains its name: the bot bans a member by asking Discord to delete the messages (which is not possible with a kick), then unban immediately the member.

Warning: For this command, the bot needs “Ban members” permission, and you need to have a role to use the “kick” command

2.7 Ban/Unban

Syntax: `ban <user> [duration] [days_to_delete] [reason]`

The ban allows you to instantly ban a member from your server. This means that the member will be ejected, and will not be able to return before being unbanned by a moderator. The ‘days_to_delete’ option represents the number of days worth of messages to delete from the user in the guild, between 0 and 7 (0 by default)

The duration of the tempban is the same as for the tempmute: use `XXd` for days, `XXh` for hours and `XXm` for minutes (replacing `XX` by the corresponding number, of course!)

To cancel this action, use the Discord interface or the `unban` command. The member will nevertheless have to decide for himself if he wishes to return to your server.

Syntax: `unban <user> [reason]`

This command allows you to revoke a ban, whether it was made via this bot or not. Just fill in the exact name or the identifier of the member you wish to be unbanned so that the bot can find the member you choose in the list of banned members for the member in question.

The persons authorized to use this command are the same as for the `ban` command(see the `config` command).

Warning: For both commands to succeed, the bot must have “Ban members” permissions (as well as be placed higher than the highest role of the member to ban).

2.8 Banlist

Syntax: `banlist`

If you ban so many people that you don't remember the exact list, and you have the laziness to look in your server options, this command will be happy to refresh your memory without too much effort.

The 'reasons' argument allows you to display or not the reasons for the bans.

Note: Note that this command will be deleted after 15 minutes, because privacy is private, and because we like privacy, it is only available for your server administrators. Ah, and Discord also likes privacy, so the bot can't read this list if he doesn't have permission to "ban people".

2.9 Handling cases

2.9.1 View list

Syntax: `cases list <user>`

If you want to know the list of cases/logs that a member has in this server, you can use this command. Note that to select a member, you must either notify him/her, retrieve his/her ID or write his/her full name.

The persons authorized to use this command are the same as for the *warn* command.

Warning: The list of cases is returned in an embed, which means that the bot must have "Embed Links" permission.

2.9.2 Search for a case

Syntax: `cases search <case ID>`

This command allows you to search for a case from its identifier. The identifiers are unique for the whole bot, so you can't see them all. However, the ZBot support team has access to all the cases (without being able to modify them)

Warning: The case is returned in an embed, which means that the bot must have "Embed Links" permission to send it correctly.

2.9.3 Edit Reason

Syntax: `cases reason <case ID> <new reason>`

If you want to edit the reason for a case after creating it, you will need to use this command. Simply retrieve the case ID and enter the new reason. There is no way to go back, so be sure to make no mistake!

The persons authorized to use this command are the same as for the *warn* command.

2.9.4 Remove case

Syntax: `cases (remove|clear|delete) <case ID>`

This is the only way to delete a case from the logs for a user. Just to make sure you don't forget the command name, there are three aliases for the same command.

The locker will be deleted forever, and forever can be very, very long. So be sure you're not mistaken, there's no backup!

The persons authorized to use this command are the same as for the `warn` command.

2.10 Anti-raid

Not a command, but a server option.

This option allows you to moderate the entry of your server, with several levels of security. Here is the list of levels:

- 0 (None): no filter
- 1 (Smooth): kick members with invitations in their nickname
- 2 (Careful): kick accounts created less than 5min before
- 3 (High): ban members with invitations in their nickname, and kick accounts created less than 30min before
- 4 (°°): ban members created less than 30min before, and kick those created less than 2h before

Note: Note that the levels are cumulative: level 3 will also have the specificities of levels 1 and 2

Warning: The bot must have access to “Kick members” and “Ban members” permissions

2.11 Anti-bot verification

How does it work?

The verification system works with a simple command and a role, and filters most of the selfbots that attack your servers.

Zbot uses a list of random questions he asks the user to test it, and if the answer is correct, the user is removed from the defined role (if he has it). The command to type to “verify” is `verify`, and to define which role to remove, it is the configuration option `verification_role`, configurable using the command `config change verification_role <role>`.

It is recommended to give this role to all new members via the `welcome_roles` option, then block access to the server for this role, in order to force the new members to check themselves.

List of commands: `verify`: ask a question to check the member `config change verification_role <role>>` configures the role to be removed from the verified members

Warning: For this system, the bot **must** have “Manage Roles” permission. The roles to be removed **must** also be lower than the role of Zbot in your server hierarchy (Server Settings > Roles tab).

2.12 Miscellaneous

2.12.1 Emoji Manager

With this command, you can become the undisputed master of the Emojis and handle them all as you please. You can even do something that no one has ever done before, a beta exclusivity straight out of the Discord labs: restrict the use of certain emojis to certain roles! **YES!** It's possible! Come on, let's not waste any time, here's the list of commands currently available :

- `emoji rename <emoji> <new name>` : renames your emoji, without going through the Discord interface. No more complicated thing.
- `emoji restrict <emoji> <roles>` : restrict the use of an emoji to certain roles. Members who do not have this role will simply not see the emoji in the list. Note that there is no need to mention, just put the identifier or the name.
- `emoji clear <message ID>` : instantly removes reactions from a message. This message must be indicated via its identifier, and belong to the same chat as the one where the command is used. The bot must have “Manage Messages” and “Read Message History” permissions.
- `emoji list` : lists all the server's emojis, in an embed, and indicates if some of them are restricted to certain roles. The bot must have “Embed Links” permission.

Warning: The bot needs the [Manage Emojis](#) permission to edit these pretty little pictures. And you, you need Administrator permission to use these commands.

2.12.2 Role Manager

Nice command that allows you to do different things with the server roles (other subcommands will be created later). The permissions required to execute them depend on the subcommands, ranging from anyone to the administrator. If you have any ideas or other suggestions, feel free to contact us via [our Discord server](#), or in PM at the bot!

- `role color <role> <colour>` (alias *role colour*): Changes the color of the given role. The color must be in hexadecimal form, although some common names are accepted (red, blue, gold...). To remove the color, use the name *default*. Please check notes 1. and 2.
- `role give <role> <user(s) | role(s)>`: Give a role to a list of people. You can target as many users or roles as you want, so for example to target your friends Joe and Jack, plus the Admin role, use `role give superRole Joe Jack Admin`. Please check note 2.
- `role remove <role> <user(s) | role(s)>`: Same as above, but instead of giving them, it takes them away. Please check note 2.
- `role list <role>`: List every members who are in a specific role, if this number is under 200. The bot must have “Embed Links” permission to display the result. Please check note 2.
- `role server-list`: Liste every role of your server, with the members count. The bot must have “Embed Links” permission to display the result. Please check note 2.

Warning:

- (1) The bot need the “[Manage roles](#)” permission, also his highest role need to be higher than the role he's trying to edit.

(2) You need to have the “Manage roles” permission (or be an administrator) to use this command. Else, Zbot won’t react.

2.12.3 Unhoist members

People like to put strange characters in their nicknames to appear at the top of the membership list. With this command you will be able to put an end to this habit. Simply type the command without argument to remove all non-alphabetic characters (a-z A-Z 0-9) at the beginning of the nickname, and you can give your own characters via an argument. Easy, isn’t it?

Syntax: `unhoist [characters]`

Warning: It is necessary that the bot has “Manage nicknames” permission, and that its role is above the roles of the members to be renamed.

Server configuration

3.1 Config options

Recently, ZBot has features that can be modified per server, to make each server unique. You will find the language of the bot, the activation of certain options (such as the xp system), the list of roles authorized to use certain commands (ban, clear. . .), the welcome messages, and many other options.

The list of options continues to grow as development progresses, that's why a website is planned to make server configuration easier.

Note: For the curious, know that all the configuration of each server is entirely saved in a MySQL database file, which makes its use easier than a simple txt or csv file.

3.1.1 Watch

Syntax: `config see [option | page]`

The *see* subcommand allows you to see the value of a configuration, with a mini explanatory sentence. If no option is specified, the entire configuration will be displayed in a single message. Note that the bot can suffer a slight latency since this data is stored in an external database.

The page number is used because the options are too numerous to be displayed on a single screen. They have therefore been grouped by page of 20, accessible via this number (default 1).

A detailed list of all options is available *below*.

Warning: To display this command correctly, ZBot only needs Embed links permission.

Syntax: `config list`

This command will give you a list of all the bot configuration options, in case you are looking for a specific name, or if you just want to make sure you know them all. However, no details will be given, to know what these options are used for you will have to go to the documentation.

3.1.2 Modify

Syntax: `config change <option> <value>`

This subcommand allows you to modify the value of an option. Just enter the exact name of the option and its value. A validation message will then be sent if the request has been correctly executed.

If the value contains several objects, such as a list of roles or channels, they must be separated by commas, like this:
`config change clear Admin, Moderators, Special role for Special people.`

Note: When the value takes the form of roles, for more comfort you are not obliged to mention them: the exact name or the identifier of the role is enough. The same goes for channels.

3.1.3 Delete

Syntax: `config del <option>`

This subcommand can be useful to reset an option to its default value. By executing this command, the option will be deleted and will take the same value as originally.

3.1.4 List of every option

- **prefix:** Character string that will be the bot prefix, for all commands, beginning with the validation message. The prefix must be between 1 and 5 characters long. By default, `!`.
- **language:** Language of the bot. Currently only the languages `fr` (French), `en` (English), `fi` (Finnish) and `de` (German) are available (also you can use `lolcat` or `fr2` for more fun). The change takes place as soon as the order is validated by the system. Default `fr`.
- **description:** Description of the server, used for the `info server` command and potential partners. Default empty.
- **clear:** List of roles allowed to use the `clear` command. By default, none.
- **slowmode:** List of roles allowed to use the `slowmode` and `freeze` commands. By default, none.
- **mute:** List of roles allowed to use the `mute` command. By default, none.
- **kick:** List of roles allowed to use the `kick` command. By default, none.
- **ban:** List of roles allowed to use the `ban` command. By default, none.
- **warn:** List of roles allowed to use the `warn` and `cases` commands. By default, none.
- **say:** List of roles allowed to use the `say` command. By default, none.
- **welcome_channel:** List of channels where messages when a member joins/leaves the server will be sent. By default, none.
- **welcome:** Message sent when a member joins your server. Some variables are usable, enter the `welcome` command to see them.
- **leave:** Message sent when a member leave your server. Some variables are usable, the same as for the welcome message.

- `welcome_roles`: List of roles automatically given to members when they join the server. It is necessary that the bot is above the roles in question, and that it has the permission “Manage roles”.
- `bot_news`: List of channels to which new bot products will be sent. These are the new bugs found as well as the new features added. None by default.
- `modlogs_channel`: Channel where all moderation logs (ban, warn, clear...) will be sent.
- `poll_channels`: List of channels in which the bot will add the reactions `and` to each message
- `enable_xp`: Boolean indicating whether the xp system is activated. Default is `True`.
- `levelup_msg`: Message to send when someone reaches a new XP level. You can use `{level}` variable to include the reached level, and `{user}` to mention the user (or `{username}` if you only want the name). Default is a random sentence.
- `levelup_channel`: Channel where the bot will send every levelup announcement message. It can be either a text channel, or “none” for no channel (Zbot won’t send any levelup channel), or “any” if you want it in the same channel as the message. Default to any.
- `xp_rate`: Exp modifier, which multiplies the gain of xp by this number. It must be between 0.1 and 3, rounded to the nearest 1/100.
- `xp_type`: Type of XP system to use: `global` if you want to use the accross-server system, common with every other servers which use it, or `local` if you want a more private system. There is also a `mee6-like` system, which uses the same rules as the MEE6 bot, and is also local. Default to `global`.
- `noxp_channels`: List of text channels where members will not be able to earn any exp. Not necessary if XP is disabled in your server.
- `anti_caps_lock`: Boolean indicating whether the bot should send a warning message when a message with too much capitalization is sent. Default is `True`.
- `enable_fun`: Boolean indicating if the fun part (documentation in preparation) is activated. If so, all commands in this section will be available. Default is `True`.
- `membercounter`: A voice salon whose name displays the number of members on the server
- `anti_raid`: Anti-raid protection with some useful features. More information [here](#). Default level: 0
- `vote_emojis`: List of emojis that the bot will have to use when there is a voting message. This case may occur when using the vote command, or in a poll channel.
- `help_in_dm`: Boolean indicating whether the help command message should be sent as a private message, or in the server. If the value is set to `True`, the message will be sent in DM.
- `muted_role`: Role used to mute your members. If no role is specified, Zbot will check for any role called “muted”, and create one if needed, with basic permissions.
- `partner_channel`: One channel where every partners of the server will be displayed. Default to `None`.
- `partner_color`: The color of the partners embeds. Can be hex, integer or common english names. Default to `#a713fe`.
- `partner_role`: A role given to every administrator of a partner server. Default to `None`.
- `update_mentions`: A list of roles which will be mentioned in each update changelog. You can enable those changelogs with the `bot_news` option. Default to `None`.

3.2 XP System

The xp system is a system for evaluating a person's activity on a server using a point system. Each message brings a certain number of points to its author, allowing him to gain in level and to rise in the ranking. To avoid having a too easy system, each level is a bit more difficult to reach than the previous one, and security measures have obviously been taken against spam or cheating.

3.2.1 Configure your server

There are several ways to customize your xp system. In particular, you have 4 [configuration options](#), each one modifying a characteristic. And more are to come!

- **Enable/disable xp:** it is possible to enable or disable the entire xp system for your server via the option `enable_xp`. If it is set to 'true' the system is enabled, otherwise it will be 'false'. By default 'true'.
- **Change the levelup message:** the bot automatically uses a long list of random messages for your members' level changes, but you can put a single one written by you via the option `levelup_msg`. It is up to you to use then `{user}` to mention the member, `{level}` for his level and `{username}` for his simple name (without notifications).
- **Select the type of xp:** there are natively three different xp systems at Zbot, modifiable with the option `xp_type`: a `global`, in common with all servers using this system (default), a `local` respecting the same calculations but without synchronization between the servers, and a `mee6-like` which uses the same rules as the famous [MEE6 bot](#).
- **Change the gain rate of xp:** if you find that your members are not earning xp fast enough (or too fast), or if you want to make a special event xp for a limited time, you can add a gain modifier between `x0.1` and `x3`, which will multiply by its value each point of xp earned. Not usable for the global xp system, of course. Option name: `xp_rate`.
- **Prevent xp in some channels:** although Zbot prevents people from earning xp with its commands, it cannot detect commands from other bots. So you can prevent your members from earning xp in certain channels via the `noxp_channels` option, which contains a list of all channels where your users can't have any experience points.
- **Select a channel where to send levelup messages:** sometimes levelup messages can be a bit spammy. So you have an option to select a single channel where to send level up messages. It is also possible to disable these messages via the same option. Enter the command `config change levelup_channel` followed by the name of your channel, or "none" to disable the message.

3.2.2 Roles rewards

Roles rewards are roles given to your members when they reach a certain level of xp. These levels are defined by you (or by anyone with "Manage Server" permission), and you can add up to 7 rewards per server.

The main command to manage these roles is `roles_rewards` (or its alias `rr`). Here is the list of commands currently available :

- `roles_rewards add <level> <role>` : allows you to add a new role to the list of roles-rewards. The level is at least 1, without maximum, and to give the role you can provide either the Identifier or the name.
- `roles_rewards remove <level>` : allows you to delete a role-reward at a certain level, to prevent the next people reaching that level from getting the role. People currently with this role will not lose it, unless you perform a reload via the following command.

- `roles_rewards reload`: reload all roles, to check that each member has the right roles. If a member has excess role-reward, they will be removed; similarly, if a member misses certain roles, they will be assigned to him.
- `roles_rewards list`: lists all currently configured roles-rewards, with their corresponding level, as well as the maximum number of roles allowed for your server. The bot must have “[Embed Links](#)” permission.

Warning: For these roles to work properly, the bot **must** have “Manage Roles” permission. The roles to be given or removed **must** also be lower than the role of Zbot in your server hierarchy (Server Settings > Roles tab).

3.3 Partners system

As a server grows, it is not uncommon to see partnerships formed with other servers. Some may even partner with bots. Zbot therefore offers a system to manage these partnerships in a clean and automatic way. Thanks to this system you can add, edit or remove partners in a few commands, and they will all be displayed in the same place, with the main information about them.

This information on partners is refreshed every 7 hours, starting at 1am (Paris time). It is currently impossible to reload the list yourself, only a Zbot administrator can do so.

3.3.1 Add a partner

Syntax: `partner add <invite> [description]`

Allows you to add a server or bot to your partner list. The invitation must be either a server invitation (starting with `discord.gg`) or a bot invitation (`discord.com/oauth`). This invitation will be used to synchronize the partner, so make sure it does not expire.

3.3.2 Change the embed color

Syntax: `partner color <new color>`

Modifies the color of the partner embed, i. e. the color of the bar to the left of the presentations. An alias exists with the subcommand “`colour`”.

3.3.3 Modify a description

Syntax: `partner description <ID> <new message>`

Adds or modifies the description of a partner. The identifier must be that of the partnership, obtainable via the command `partners list` or under the embed displayed in the partners’ lounge.

3.3.4 Change a server invite

Syntax: `partner invite <ID> [new invite]`

It often happens that for X reason an invitation becomes invalid. Problem: Zbot uses the partner invitation to synchronize partners with the channel. There is therefore a command to quickly change the invitation of a server.

Note: If no new invitation is given in the command, the bot will send you the one currently in use.

3.3.5 List every partners

Syntax: `partners list`

Lists all the partners that your server currently has. The bot will display the name of the partner, the type (server or bot), and the date of addition. You will even have the list of servers that have added you as a partner!

Warning: For a better display of the list, it is recommended to give “[Embed Links](#)” permission to the bot.

3.3.6 Reload your list

Syntax: `partner reload`

Allows you to remove a partner from the list. You will be asked for a confirmation, to avoid misuse. Once a partner is removed, you must reconfigure it completely if you want to put it back into the channel.

3.3.7 Remove a partner

Syntax: `partner remove <ID>`

Allows you to remove a partner from the list. You will be asked for a confirmation, to avoid misuse. Once a partner is removed, you must reconfigure it completely if you want to put it back into the channel.

3.4 Server backup

Zbot has a system to backup your server, saving your roles, channels, emojis, webhooks, icons, permissions, and much more. You will also find in this file the list of members and their permissions, although Zbot is not able to reinvite members if needed. This backup will avoid the most important damage, those little mistakes that can destroy your server as I myself experienced a few years ago. I hope to be able to save what is important to you.

When you load the backup, the bot may not be able to apply some changes. However, it will give you a complete list of what has and hasn't been changed so that you can fix it yourself.

Warning: The bot will need as many permissions as possible, which includes: Manage Roles, Manage Channels, Manage Webhooks, Ban Members, Manage Emojis.

3.4.1 Create a backup

Syntax: `backup create`

Creates a file containing as much information as possible on your server, within the limit of the permissions granted to the bot. You will have to keep this file carefully, it will be necessary for you when you will want to restore the backup.

3.4.2 Load a backup

Syntax: backup load

Uses the file attached to this message to load a backup, based on the data stored in the file. Be sure to send the file in the same message as the command, so that Zbot can easily find it. If the bot lacks permissions, it will try to skip this step and write it down in the logs. The set of logs is then sent at the end of the procedure.

User configuration

Users have a very important place in the bot code. Well, after all, without users, no bot, right? That's why we're currently working to give you as much as possible with ZBot, so you can fully enjoy your experience.

In this section, you will find the XP module, a classic in Discord bots, as well as a command to change your own preferences, such as the color of your xp card or the language used in your personal messages. Don't worry, other possibilities are planned and will be added later!

4.1 XP system

The xp system is a system for evaluating a person's activity on a server using a point system. Each message brings a certain number of points to its author, allowing him to gain in levels and climb in the ranking. To avoid having a too simple system, each level is a little harder to reach than the previous one, and security measures have obviously been taken against spam or cheating.

4.1.1 Check the XP of someone

Syntax: rank [user]

This command is used to view the number of xp, rank and level of a member. You can select this member either by mentioning it or by using his name or ID. If no user is given, your own XP will be displayed.

Note: The bot sends the format adapted to its permissions: if it can `send files`, it will display the xp card. If it can `send embeds`, it will display it in an embed, and otherwise by text.

4.1.2 Get the general ranking

Syntax: top [page] (guild|global)

If you want to know who is at the top of the ranking, or who is following you so closely, this command is the ideal function. It allows you to have the name, xp and level of 20 people per page. The operation of the pages is quite intuitive: page 1 shows 20 first users, page 2 between 21 and 40, page 3 between 41 and 60, and so on. Simple, right?

If you give the argument *guild* (or *server*, as you want), the top will only display users who are on the current server, instead of all bot users.

Note: The bot can send this message without special permission, but don't hesitate to give him permission to [send embeds](#) to make the result more aesthetic!

4.2 Your info

We were talking about adding customization options for each user. Here is the section that concerns them, where you can all configure some nice options to create your own identity on the bot. For the moment there are few (very few) options, but others will come later, we guarantee it!

4.2.1 Change your xp card

Syntax: `profile card [style]`

With this command, you can change the design of your xp card (the one used for the *rank* command). A long list of styles are available to everyone, but others are exclusively unlocked and reserved for certain people (more details on this part will come later).

If you use the command without argument, the bot will show you an example of a xp card with the style you currently have. But if you enter a style name that does not exist, it will give you a list of usable styles *for you*.

Note: Note that to be able to display your card, the bot needs [Attach Files](#) permission!

4.2.2 Allow or disallow an option

Syntax: `profile config <option> [true|false]`

You can allow or disallow one of the configuration options via this command, much like configuring a server. Here is the list of available options

Option 'animated_card'

This option allows you to enable the rendering of your xp card in.gif format if you have an animated profile image. Since the image processing time is much longer, with reduced quality, this option is disabled by default.

Option 'auto_unafk'

This option has been designed for people who don't want to manually type the *unafk* command as soon as they are back. So by activating this system, Zbot will automatically remove the AFK tag from you as soon as you send a message. As simple as that!

Option 'usernames_log'

By default Zbot records when you change your nickname, whether it is your main name or the nickname in a server. This data is then used for the results of the `usernames` command, and nowhere else. But for confidentiality reasons you can prevent your future changes from being saved at any time, via this option.

ZBot also allows you to retrieve information about the virtual world surrounding you. There you will find a single command that summarizes all the information about a channel/member/role/server/invitation/emoji, as well as a few other commands allowing you to study further.

5.1 About

Syntax: `about` or `botinfo`

This command sends a short presentation text of the bot, so that you know it a little better. It will also give you some links that may be useful to you (like the one to invite the bot, or to access its Discord server).

Note: For this command, ZBot doesn't need any specific permission! Good news, isn't it?

5.2 Bot invite

Syntax: `botinvite` (alias `botinv`)

Shorter than the 'about' command, this one only send an url to invite the bot. And this url will always be working, even if our web server crashes.

5.3 Help

Syntax: `help` [`command|cog`]

Allows you to know the list of all the orders currently available for you. The list is interactive, which means that you will only see the commands you are allowed to use.

You can specify a command (or subcommand) to get more details about it, or a cog (a code module) to see the list of commands related to that cog.

Note: For a better visual overview, it is recommended to give the permission “[Embed Links](#)” to the bot. In addition, you can configure the bot to [send the message as a private message](#).

5.4 Info

Syntax: `info [type] <object>`

This command is probably the most powerful in the information module. It allows you to find information on any item on your server: members, roles, text and voice channels, categories, emojis, invitations, as well as the server itself. Oh and also raw snowflakes (Discord IDs). Some information is even available about users who are not on your server!

You can enter the name, the mention, or the [identifier](#) of the object to be searched, and if the type of object (member, user, role...) is not specified, the bot will search itself to identify it. Note however that you are obliged to inform the type if your search includes spaces.

Note: Some fields may not appear under certain conditions. No need to worry, it’s just that Discord didn’t send the requested information to the bot. And there’s nothing we can do about it `_()_/_`

Warning: The necessary permissions for the bot depend on the desired result: for example “[Manage webhook](#)” is required to get the list of webhooks of a channel.

5.5 Membercount

Syntax: `membercount`

With this command, you can get the number of members on your server, but also the number of bots, of humans, people connected, and probably other numbers that will be added later. This is a small basic command without much functionality, but it allows you to quickly keep up with these statistics.

Note: Good news! The bot does not need any specific permissions for this command! Just keep in mind that the rendering is much prettier with “[Embed Links](#)” permission enabled.

5.6 Permissions

Syntax: `perms [channel] [user|role] or permissions [channel] [user|role]`

This small command allows you to see the list of permissions assigned to a member/role in a particular channel. The channel can be either a text or a voice chat, but if you don’t provide any, the bot will select the general permissions as set in the Server Configuration. To inform a member or a role, it is only necessary to enter his exact name, his [ID](#) , or his mention. If no name is given the targeted member will be the one who enters the command.

Warning: The only permission needed to grant the bot is “Embed Links”.

5.7 Ping

Syntax: `ping [ip adress]`

The ping command allows you to get the bot latency. It's useful if you want to check why your command takes too long to be read. The number corresponds with the delay between the moment when your message reaches Discord and the moment when the bot's response is received by the API.

If you give an ip address in the command, the bot will send a certain number of packets to this server to see if it is active, and know its latency. This may take a short time, depending on the server bandwidth and the number of packets to send.

5.8 Prefix

Syntax: `prefix or prefix change <new prefix>`

A nice shortcut to know the prefixes to which the bot responds. This is usually the prefix defined in the [configuration](#), plus the mention of the bot.

Note that this result may differ from the `config see prefix` command when the database is out of sync.

..note:: The subcommand `prefix change` is an alias of `config change prefix`

5.9 Usernames history

Syntax: `usernames <user> (aliases: username or usnrn)`

This command displays the history of all nickname changes of a member. The Discord API does not give this information, so Zbot records each change, therefore it is possible that some nicknames may not be displayed in the list.

If you don't want your names changes to be recorded, you can opt-out by using the [profile config usernames_log](#) command.

5.10 Welcome message

Syntax: `welcome or bvn`

This command helps you to define a message sent automatically by ZBot when a member joins or leaves your server (see the [config](#) command). You will find how to select the channel, as well as the variables that can be used in the messages.

At the very beginning, ZBot was a single server bot, and focused on the world-famous Minecraft game.

Even after diversifying, the bot has not forgotten its origins and remains very open to this cubic world, offering several commands related to the game. You will find a huge database on all the blocks, entities, items, commands, progress, potion effects, enchantments, and more. As well as a command to obtain the status of a Minecraft server (it is possible to display it permanently so that the information is refreshed regularly). And another one for the state of Mojang's servers. If you find this content is very low, don't worry: other orders are in preparation!

Note: The whole database comes from a single Minecraft site (French, like ZBot): fr-minecraft.net . The search engine and the information collected are therefore those appearing on this site. If you observe any error in this database, do not hesitate to contact me so that I relay it to the administrator of the site!

Warning: Most of these commands are reserved for certain roles only. To allow roles to use a command, see the *config* command

6.1 MC

Syntax: `mc <type> <name>`

This command is the main command of this module: the one that allows to search the information in the database, or to get those from a Minecraft server. To ask the bot to send the status of a server and to refresh this message regularly, use the *add* subcommand followed by the server ip. The bot will then try to edit the last message about this server, and if it can't, it will send a new one.

To search in the database, the command is disconcertingly simple: you just have to write the type of your search (entity, block, etc.) followed by its name (partial or total, French or English) or its identifier (numerical or textual). The rest does itself!

To see the list of available types, enter the help mc command in the chat. If you don't find what you're looking for, don't worry: this type is probably planned for later!

Warning:

- The bot needs the “[Embed links](#)” permission to send its search query, as well as “[Read message history](#)” and to display the status of a server (enabled with *add* subcommand)
- Adding server tracking automatically with *add* is considered the same way as an rss feed, which means that it takes a place in your feed list (limited to a certain number, except for a few special cases).

6.2 Mojang

Syntax: `mojang` or `mojang_status`

This command, much more basic, uses the Mojang API to get the status of its servers. For each server you will thus have its state, its url, as well as a short description.

Note: The bot does not need any specific permission for this command, however note that the appearance will look better if “[Embed links](#)” permission is enabled.

More and more bots offer the feature to follow news feeds, sometimes [rss atom feeds](#), but most often Twitter or YouTube profiles. ZBot allows you to track any rss/atom feed, as well as any Twitter/YouTube/Twitch/DeviantArt channel. For Reddit feeds, you can search for the url of the rss feed, but a command will be created to make your life easier!

With this bot you have two possibilities to follow a feed: manually request the last post, or configure an automatic follow-up in a text channel. In the case of automatic tracking, ZBot will scan all feeds every ten minutes to check for new posts, sending them in if there are any. Just be careful: this automatic tracking costs a lot of resources to the bot, so you are limited to a certain number of automatic feeds (same for rss, twitter, youtube or minecraft) !

To manage this plugin (add, edit or remove feeds), you will need at least the Manage Server permission.

7.1 See the last post

Syntax: `rss <youtube|twitter|twitch|deviant|web> <name|link>`

This command allows you to see the last post of a youtube channel, a user on Twitter or Twitch or DeviantArt, or an rss feed. You can enter `rss <type> help` to get a more complex guide to this command.

To go faster, aliases such as 'yt' or 'tw' are available! YouTube channel names or frequently used web links are already listed in the bot database. Remember to check it out!

Note: No specific permission is required for this command. Remember to allow the use of external emojis to get a prettier look.

7.2 Follow a feed

Syntax: `rss add <link>`

If you want to automatically track an rss feed, this command should be used. You can only track a maximum feeds, which will be reloaded every 10 minutes. Note that Minecraft server tracing also counts as an rss feed, and therefore will cost you a place.

For Twitter and YouTube channels, simply give the link of the channel, so that the bot automatically detects the type and name of the channel. If no type is recognized, the 'web' type will be selected.

Note: To post a message, the bot does not need any specific permission. But if it is a Minecraft server flow (see the [corresponding section](#)), don't forget the "Read message history" permission!

7.3 See every feed

Syntax: `rss list`

If you want to keep an eye on the number of rss/Minecraft feeds registered on your server, this is the command to use. The bot will search in the depths of its incomprehensible files to bring back the list of all the flows, and summarize them for you in a nice embed.

Warning: The bot needs "Embed Links" permission!

7.4 Mention a role

Syntax: `rss roles [flow ID] [roles]`

This rss flow tracking option allows you to notify a role when a new post arrives. The roles mentioned are different between rss flows, which allows you a greater handling.

The "flow ID" argument is the identifier of the flow (found with the command `rss list`). If you do not enter this argument, or if the feed can't be found, the bot will open a menu where you can choose which feed to modify. Then another menu will allow you to choose which roles you want to mention.

The "roles" arguments is the list of roles you want to mention, separated by spaces (if some of them contains spaces, you can use quotations "...") instead). If not specified, Zbot will ask you for the list. You can either use names or IDs, or put "none" to remove every mention.

Warning: For this command too, the bot needs "Embed Links" permission!

7.5 Change the text

Syntax: `rss text [flow ID] [new text]`

This command is particularly useful if you want to change the text of an rss flow tracking, for example to have a customized text, or in your native language. Many tools are at your disposal (also known as *variables*) that allow an optimal personalization of the message. That's right, we thought of you.

If the rss feed ID is not given, the bot will open a menu to select it. And for the text, if you have forgotten it, the bot will also ask you to know it, and will provide you the current text and a list of the usable variables.

Warning: Yo know what? For this command, the bot needs “Embed Links” permission!

7.6 Move a feed

Syntax: `rss move [flow ID] [new channel]`

If you want to move an rss feed without having to delete it, recreate a new one and then reconfigure it, you can use this command. It can also be useful to configure a flow in a secret chat room, then reveal it to your entire server without having to temporarily close your chat!

If no identifier is given, the bot will ask you which one to modify. As for the channel, if you do not specify any, it will select the one in which you type the command.

Warning: Here again, the bot needs “Embed Links” permission!

7.7 Setup a feed embed

Syntax: `rss use_embed [flow ID] [use embed]` or `rss embed <flow ID> [use embed] <parameters>`

Sometimes people want to have a lot of control over what is happening in the world. Since we are unable to offer it to them, we offer you a great control on the embeds sent by rss flows. The first command allows you to enable the use of embed instead of classic text, the second one allows you to choose a title, a color and a custom footer.

For the first command, if you do not give the feed identifier or a boolean value, the bot will ask you for it. However, you must give it yourself for the second command (you can find it via the *rss list* command).

The syntax of the color/text customization parameters is the same as for the `embed` command, i.e. in the form `key = "value"`, with the possible keys “color”, “footer” and “title”.

7.8 Delete a followed feed

Syntax: `rss remove [flow ID]`

With this command, you can stop following an rss/minecraft flow. And it’s also very easy to use. Just one command and *poof*, we shut down the machinery.

The “flow ID” argument is the identifier of the flow (found with the command *rss list*). If you do not enter this argument, or if the feed can’t be found, the bot will open a menu where you can choose which feed to modify.

Warning: ZBot needs “Embed Links” permission to send the selection list!

7.9 Reload every feed

Syntax: `rss reload`

If your favorite youtube channel has just posted a new cool video, and the bot takes too long to post it in your specially designed living room, you can force it to refresh the list of your youtube, twitter and other websites, in addition to Minecraft servers. This command will allow you in a few seconds to be again at the top of the latest news!

Note: Note that to avoid lags, a 10-minute cooldown is active on this command.

The reaction role system is increasingly used in Discord servers because of its simplicity. It allows your users to obtain or withdraw a role by simply clicking on a reaction. It is faster than a command to enter, more ergonomic, and it allows a better presentation of the obtainable roles... what more could you ask for?

To make it as easy as possible for users and moderators to use these reactions, we have tried to make it as easy as possible for you to configure this system.

8.1 Add and remove a reaction

Syntax: `roles_react add <emoji> <role> [description]`

Syntax: `roles_react remove <emoji>`

These two commands are used to add or remove a role to the list of reaction roles. Very simple to use, this makes it possible to link each role to a corresponding emoji, which will be used as a reaction for users. The emoji can be both a Discord emoji or a customized emoji of your server, even animated.

Note that it is not possible to give more than one role per emoji, and that you are limited to a certain number of roles in your server. This limit is visible via the 'list' subcommand.

8.2 Get or leave a role

Syntax: `roles_react get (alias display)`

Syntax: `roles_react join <role>`

Syntax: `roles_react leave <role>`

There are two ways to assign or withdraw a role: either via reaction or via a command.

For the first case, the 'get' command will display a list of available roles with their corresponding emojis, and will add each reaction at the bottom of the message. It is then sufficient to click once on a reaction to get the role, and a second time to remove it. As simple as that.

If you know exactly which role to get/remove, it is faster to use the ‘join’ and ‘leave’ subcommands followed by the role in question. You can give either the name of the role or its ID as a parameter. Note that it is not possible by this means to obtain or loose a role which is not in the list of reaction roles.

Warning: For the **first** command, the bot needs “Embed Links” and “Add Reactions” permissions for this command

8.3 List every roles-reactions

Syntax: `roles_react list`

To get a list of all the role-reactions without wasting time waiting for the bot’s reactions, this command will be very useful. It also allows you to have the number of roles currently used, and the maximum number of roles you can have on your server.

8.4 Update your embed

Syntax: `roles_react update <message ID> [changeDescription?] [emojis list]`

This command is very useful for those who have pinned the embed containing all the reactions of the bot. Instead of deleting and resending the message each time you add or remove a role, simply use the command to have the bot check the roles descriptions and usable reactions.

The second argument ‘changeDescription’ can be used when you don’t want Zbot to change the embed description. In this case it will only update the reactions (adding new ones if needed).

You can also use the third argument, a list of emojis, if you want your embed to contain only specific roles/emojis. Thus you can create different roles-reactions embeds with the same system.

Note: Note that there are two criteria for the bot to recognize the embed as its own: it must be sent by itself, and the footer text must be the same as in the official embeds. This means that you can use the *embed* command to send a custom embed, it will still work.

This section of the bot contains lots of fun commands, which will be less useful than the other sections, but will add a good atmosphere in your server. Because a server's life is not just about moderation or utilities, ZBot couldn't be without a few moments of fun.

However, if too much fun bothers you, this option can be disabled at any time with the `config` command (`enable_fun` parameter). The choice is yours!

9.1 List of every command

9.1.1 AFK

Syntax: `afk [reason]` or `unafk`

This command will be useful if you are often busy doing something while Discord is open. The principle is to put a tag[AFK] (which means Away From Keyboard) on you and notify anyone who tries to mention you. So there is a command to put you in AFK mode (the reason is optional, it will be indicated to people who are trying to mention you), and another command to exit AFK mode.

Note: Note that even if the system is specific to each server (it works with your nickname), the reason is global: if you change the reason in one server, it will be effective on all other servers having you as AFK!

9.1.2 Bigtext

Syntax: `bigtext <text>`

Here is a handy command for people who have trouble seeing small text of Discord. Or just for those who want to have some fun. In fact the bot will use your text to convert each letter into Discord Emojis, so that you have bigger text.

You want some good news? If you have permission to use the `say` command, the bot will delete your message after posting it !

Warning: The only permission required is “[Manage messages](#)” to possibly delete your own message.

9.1.3 Blame

Syntax: `blame <name|list>`

You’ve probably already gotten a chance to get mad at someone. It doesn’t matter why. And since it’s always better to do it together, you may like to protest collectively against a common evil. This command therefore allows you to blame someone without getting tired of spamming. You just have to enter a name and hope that its custom message exists in our database, and we’ll be happy to irritate ourselves for you.

You can access even more names by being on some secret servers! Enter the `list` argument to see which ones you have unlocked.

Note: Currently only a bunch of images exist, but if you have others to propose (that respect the theme), come see us!

Warning: ZBot needs “[Attach files](#)” permission to send its message. You’re not gonna ask him to be angry without giving him permission to do it?

9.1.4 Bubble-wrap

Syntax: `bubble-wrap [width=10] [height=15]` (aliases: `papier-bulle`, `bbw`)

Just bubble wrap. Which pops when you squeeze it. That’s all. `#Just4Fun`

Width should be between 1 and 150, height between 1 and 50.

9.1.5 Count messages

Syntax: `count_msg [limit] [user] [channel]`

A nice little order that counts the number of messages in the history of this channel posted by someone. The limit corresponds to the number of messages to study in the chat, 1000 by default. And since some very old chats can have a very many many MANY many messages (yes, 5 times *many*), we have a set limit on the number of messages you wish to search.

If no user is given as parameter, Zbot will count your own messages. Same for the channel, if you don’t provide any, Zbot uses the current one.

Warning: It seems obvious, but the bot needs “[Read message history](#)” permission to read the messages history. . .

9.1.6 Congrats

Syntax: `gg`

This is a nice little gif to use when you want to congratulate your friend. And remember to thank Gilderoy Lockhart for his charming smile!

Warning: ZBot needs “[Attach files](#)” permission to send this nice message.

9.1.7 Lmgtfy

Syntax: google <search>

Yes, that name is unpronounceable. On the other hand, commands are executed on written channels, not voice, so that’s good. And, for use, refer to the website: “For all those people who find it more convenient to bother you with their question rather than search it for themselves.”

Warning: ZBot needs “[Manage messages](#)” permission to delete the invocative message.

9.1.8 Loading

Syntax: loading

Do you think that the time is long? Or do you just need to express a veery loooooong loading time? Use this command, designed especially for this bot!

Warning: ZBot needs “[Attach files](#)” permission to send this message.

9.1.9 Me

Syntax: me <text>

When you need the bot to talk about you, like “*Z_runner runs away very fast*”, use this command. The “me” will be replaced by your nickname, and if you are allowed to use the [say](#) command, your original message will be deleted.

9.1.10 Money

Syntax: money

This command is perfect if you want to give the impression of literally swimming in piles of money. Can be placed in any discussion that is more or less related to this theme. Fortunately, the use of this command is not overtaxed.

Warning: ZBot needs “[Attach files](#)” permission to send this gif.

9.1.11 Nope

Syntax: nope

A small command to use when you do not agree with your interlocutor. Small, but it has the merit to be clear and to quickly cut short the discussion. And even better, if you have permission to use the `say` command, your invocation message will be deleted ! *Camouflage activated!*

Warning: ZBot needs “`Attach files`” permission to send its message, and “`Manage messages`” to delete yours.

9.1.12 Nuke

Syntax: `nuke`

The conversation’s getting hot, do you want to blow it up? Let off some steam with a nice little gif, just to get the point out and calm down negotiations!

Warning: ZBot needs “`Attach files`” permission to send this gif.

9.1.13 HEEELP

Syntax: `osekour`

This command is specially designed for French salons. In France, a call for help is sometimes called “au secours” (“osekour” in a very, very short version). If you need the bot to help you, type this command and see which random answer it will come out!

9.1.14 Party

Syntax: `party`

Do you party often at home? If so, you have enough power to prove it with this command. And if not, she’ll just put some good humor in the chat!

Warning: Two permissions are required for this command: “`Attach files`” and “`Use external emojis`”.

9.1.15 Pibkac

Syntax: `pibkac`

This is a quite well-known case in IT. To quote the definition of the [Urban Dictionary](#), “*Problem Is Between Keyboard And Chair: Another term used to refer to an id10t or other person who probably should not own a computer*”.

Warning: ZBot needs “`Attach files`” permission to send this gif.

9.1.16 Piece

Syntax: `piece`

Use it if you need to flip a coin, but you don’t have any money to show in front of your computer’s camera. Also, like real life, the piece can also fall on the edge! Isn’t that great?

9.1.17 Pikachu

Syntax: pikachu

Who doesn't know the world-famous Pokemon, Pikachu, who was for a long time the mascot of the Nintendo-owned company? Thanks to this command you can use gifs from this rabbit-eared Pokemon, randomly drawn from our ever-growing image bank!

Warning: ZBot needs “Attach files” permission to send any gif.

9.1.18 Pizza

Syntax: pizza

Any of you like pizza here? Those beautiful dripping pieces of melted cheese and tomato sauce, delicately placed on a crispy, light dough? If that's your case, why don't you take a look at this beautiful and mouth-watering gif?!

Warning: Yup. Here too, the bot needs “Attach files” permission to send this gif.

9.1.19 Pong

Syntax: ping

This is probably the most useless command in the bot. Try it, you may (maybe) not be disappointed!

9.1.20 Ragequit

Syntax: ragequit

Basically this command was designed for the sole use of the Creator. But since everyone has the right to get mad at something (and not just Python code), he decided to leave it open access. It's up to you to make good use of it!

Warning:

- The bot needs “Attach files” permission to send these images.
- We do not own the copyright of each of the images used in this command. If you want to design an image especially for ZBot, and are ready to give us all rights, thank you to contact us as soon as possible!

9.1.21 React

Syntax: react <messageID> <list of emojis>

This command allows you to force the bot to add reactions to a message, which is useful in certain situations. For example, if you organize a reaction vote and want to cheat a little on the statistics! Please note that only people who have access to the *say* command can use this one.

All reactions work, whether they are Discord or server reactions. All you have to do is separate them with a space. Just like magic!

Note: To find out how to find the ID of a message, follow [this link!](#)

Warning: To find the message, ZBot must have “Read Message History” permission, and “Add Reactions” permission to add reactions.

9.1.22 Roll

Syntax: roll <options>

If you can't agree with your friends, or if you want to leave the decision of a difficult choice to chance, this command will surely delight you. It allows you to select an option randomly from a list of options you provide, separated by semicolons (;). And you can put as many choices as you need!

Example: roll a little; a lot; passionately; madly; not at all!

9.1.23 Run

Syntax: run

Just... run... very... fast... εεεεε(>_<)

If you're tired of running, make the bot run for you!

Note: No specific permission is required!

9.1.24 Shrug

Syntax: shrug

Don't know the answer to a question? This is the opportunity to express it with a pretty gif straight out of our image bank! A simple command, but one which can be fun.

Warning: ZBot needs “Attach files” permission to send any gif.

9.1.25 Thanos

Syntax: thanos

I assume you know Thanos from the Avengers series. If not, to make it short, he's a bad guy who decided to kill half the universe with a single snap of his finger.

Well, if you want to know if you will be spared by this guy or not, check out the great oracle Zbot!

9.1.26 Tip

Syntax: `tip`

If you want to get some advice on how to use the bot, or just a funny fact, you will surely find what you are looking for here. This command returns a random phrase from a defined list of “Pro-tip” and “Did you know?”, to hopefully teach you something!

9.2 Bot events

From time to time, for special events of the year, Zbot has fun organizing an event where some small changes are made to the code. There is for example the tic-tac-toe whose symbols change, or many other small easter eggs of this kind... as well as the possibility to win event points!

To get event points, it is usually enough to use the bot: win games of tic-tac-toe, increase in xp level, or other actions of this kind.

9.2.1 Get info about the current event

Syntax: `events info`

You can have details about an event via this command. If an event is in progress, you will then have the explanatory summary, start and end dates, as well as any prizes to be won.

9.2.2 Get your current progress

Syntax: `events rank`

To know your progress in the event, as well as the prizes you can recover, this command is the one you need. You'll even get your ranking among all the players in the world!

CHAPTER 10

Miscellaneous

Some commands are difficult to classify in the categories of the site. They are not used for moderation, nor to get information, they can be fun but they are not listed in this category, they don't have much to do with configuration or rss...

Here is the list of these particular commands.

10.1 Bitly urls

Syntax: `bitly [create|find] <url>`

Bitly is a famous website for shortening web addresses (aka url). With this command you can create a shortcut yourself instantly using their services, and see to which address a bit.ly link refers without having to click on it. Practical, isn't it?

10.2 Book

This command allows you to manage your own library, saving your books easily. You can also only use the search function, for example to share a specific book with your Discord friends. Anything is possible!

Warning: This command is under construction, and is therefore only in its early stages. Feel free to give your ideas on the official [bot server](#)!

10.2.1 Search by ISBN

Syntax: `book search <ISBN>`

Used to search for a book from its ISBN, and displays its main information. ISBNs of length 10 and 13 are accepted.

10.3 Changelogs

Syntax: `changelogs [version]` or `changelogs list`

If you want to know the bot modification log, what has been changed in the last version or an older one, you can use this command. Introduced in version 3.5.5, it allows access to all bot changelogs from this version. For older versions, you will have to go directly to the channels of the official server!

Note: Giving the “[Embed Links](#)” permission to the bot can be useful if you want to get a better rendering. But it’s not mandatory!

10.4 Embed

Syntax: `embed <args>`

This command is particularly useful if the ‘say’ command is no longer enough for you, if you want something even bigger, with colors, images and everything that goes with it. You can send embeds (these pretty rectangles with colored bars), by customizing the title, content, image, title url, color and footer text!

Each argument is presented in the form `name="value"`. If you want a line break, you can use the character `\n`, and if you want to use quotation marks without closing the argument, you will have to escape them (with a `\"` in front). To better understand how it works, here is an example of how to use it: `embed title="Here is my title!" content="Blah blah \nBlah ?" footer="Do you mean \"Text\"? "`

Warning: Hey, this may sound weird, but Zbot needs “[Embed Links](#)” permission to send embeds...

10.5 Hour & Weather

Syntax: `hour <city>`

Syntax: `weather <city>`

With these two commands, you can get the time (and timezone) or weather for any city in the world! All you have to do is enter the name of the city, preferably in English format (London instead of Londres for example), and the magic does the rest!

Note: For the *weather* command, it is better to give the “[Embed Links](#)” permission to the bot, to get a better rendering. But it’s not mandatory!

10.6 Markdown

Syntax: `markdown`

The markdown is a set of formatting rules used everywhere, such as on GitHub or Discord. This command gives you these formatting rules, which will allow you to display spoilers, code blocks, or just simple strikethrough or italicized text.

Warning: Warning, this command contains an invitation link to the information server on the code blocks.

10.7 Reminders

If you have some issues with your memory like me, I think you should start using this command. With it, you can ask Zbot to remind you things to do later, between a minute and a few years. Like a *!d bump*, or anything else, up to you. And it also works in DM.

10.7.1 Create a new reminder

Syntax: remindme <duration> <message> or reminder create <duration> <message>

The duration argument is exactly the same as for tempmute/tempban: use XXw for weeks, XXd for days, XXh for hours and XXm for minutes (replacing XX by the corresponding number, of course!)

Warning: Zbot needs “[Embed Links](#)” permission to send the reminder.

10.7.2 List your reminders

Syntax: reminder list

Here you will get the full list of pending reminders, waiting for the end of their timers. Nothing but a list, really.

Note: Giving the “[Embed Links](#)” permission to the bot can be useful if you want to get a better rendering.

10.7.3 Delete a reminder

Syntax: reminder delete <ID>

Used when you want to stop a reminder, so Zbot will completely forget it. The reminder ID can be found with the *reminder list* command (see above).

10.8 Say

Syntax: say [channel] <text>

If you want to talk through the bot, as if it were sending your messages, this command will be a great help. Just indicate the text to send, and voilà, it’s over. If a channel is specified, the message will be sent there. Otherwise, it will be sent in the current channel.

Note: Note that this command is reserved for certain roles, which you can define in the [configuration section](#).

Warning: In addition, “[Manage Messages](#)” permission is required if you want the bot to delete your message as soon as it has posted its copy.

10.9 Tic-tac-toe

Syntax: `tic-tac-toe (alias morpion) or tic-tac-toe leave`

Yes, we did it! A first mini-game for our bot, the crab! You can play against the bot in this fast and simplistic game, just by entering the command and following the instructions (enter a number between 1 and 9 corresponding to the chosen cell). And the best part is that the only special permission required is to use the external emojis!

By using the *leave* argument you can instantly stop a game. This can be useful if you are stuck by a bug and can't join a game for example.

10.10 Vote

Syntax: `vote [number] <text>`

This command will add a little interactivity in your server by allowing the creation of votes or polls. Zbot will send a message containing your text and then add reactions to it, before deleting your original message.

If no number of choices is given, or if this number is 0, the vote will be a yes/no type. Otherwise, it will be a question of choosing between the choices using numbers. Note that it is not possible at this time to put more than 10 choices.

For this command the bot needs “[Add Reactions](#)” (add reactions to its message), “[Read message history](#)” (find its message in the chat room) and “[Manage Messages](#)” (delete your message) permissions.

Note: A big thank to the member Adri526, for his emojis specially designed for ZBot!

CHAPTER 11

Indices and tables

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