**Gaiza: Discord Utility Bot**

User Manual

*Spring 2020*

**Slippery Rock University**

Cain Kaltenbaugh [cxk1047@sru.edu](mailto:cxk1047@sru.edu)

Table of Contents

1. Gaiza Information and Utilization
   1. Inviting Gaiza to a Server
   2. Requesting the Commands
2. Commands
   1. User Level Commands
      1. Ping
      2. Avatar
      3. Invite
      4. aHelp
      5. Define
      6. Gif
   2. Admin Commands
      1. Prefix
      2. Kick
      3. Ban
   3. Welcome Module
      1. Welcome
      2. Channel
      3. Enable
3. Listeners
   1. Passive Listeners
      1. UWU Response
      2. Bored Response
      3. Microwave Response

1 Gaiza Information and Utilization

*The following is a limited explanation of the Gaiza bot and how to invite and utilize its many features*

The Discord bot Gaiza, is a bot that features multiple different utilizations. The bot itself contains multiple functionalities while maintaining a single solidified presence. There only needs to be one instance of the bot running at a time to manage several servers and users with different collective information. Users can manage the bot itself and will be able to utilize more commands in future implementations

1.1 Inviting Gaiza to a Server

1. Create a Discord account and download the platform [here](https://discordapp.com/)
2. Create a server within the platform. For help with this, the platform creator has uploaded a guide [here](https://support.discordapp.com/hc/en-us/articles/204849977-How-do-I-create-a-server-).
3. Follow this link to invite the bot to the newly created server [here](https://discordapp.com/oauth2/authorize?client_id=369295519576489984&scope=bot&permissions=2146561111).

Note: The person accessing the link must be either an admin or owner of the server.

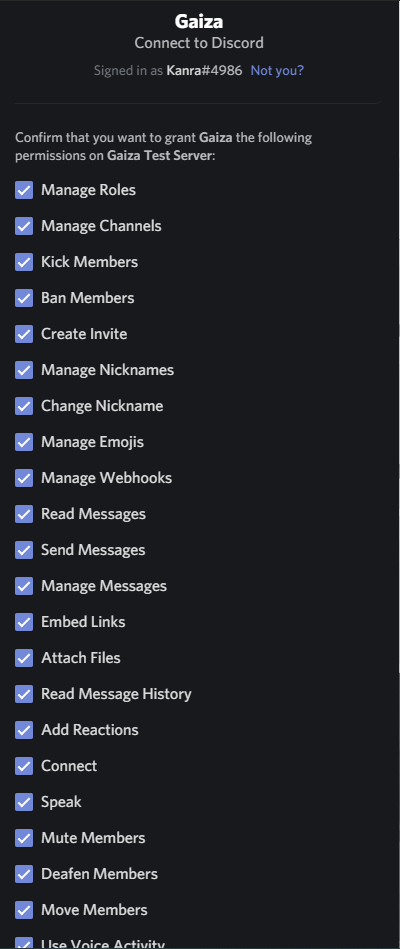


Figure 1 Invite Permissions

1. As seen in Figure 1, the bot will request what permissions it can have upon joining the server. A special role for the bot with all these roles will be created and can be managed after permissions have been applied.
2. When all the permissions have been specified, at the bottom of the list, press the authorize button and the bot will join the server

1.2 Requesting the commands

1. In order to see a list of commands, the initial command $help will need to be invoked. The bot must be able to see the channel where the message is sent to reply. Note: The prefix for all commands when a bot initially joins a server is $ until it is set to something else by the owner.

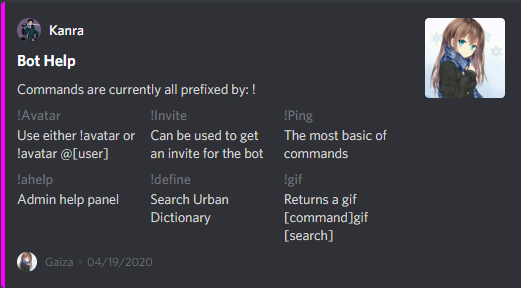


Figure 2 Help Embed Interface

1. Each command will have a name to it. Most commands will follow the syntax of “[prefix][command] <options>”. Most commands that have options for them will be detailed in the help menu.
2. In order to access admin commands, the command $ahelp will need to be invoked by a user that is either an admin or owner of the server. This will display all the commands that required a higher level of access than normal commands that can be used by everyone.

2. Commands

Commands are split into three categories. The categories are based on permission levels and how they are invoked. User commands are for basic users without privileges, admin commands are for users with admin level privileges, and listeners are commands that act regardless of user level and regardless of user invoked commands. All commands ignore case sensitivity. Any case sensitive titles are for clarity purposes only.

**For sake of simplicity, all commands will retain the default prefix of $.**

**Syntax: [required] <optional>**

2.1 User Level Commands

These user commands can be called by any user and the command can all be seen via the $help menu after start.

2.1.1 Ping

This command is typically just a beginning command for the bot in order to assure that commands are working, and the bot is running. This is the Hello World for the bot. Most typical responses with a command like this are sending back “pong”. This command simply returns a statement and mentions the user that invoked the command.

Syntax:

$ping

2.1.2 Avatar

This command can be called in two different ways. Users can simply invoke the command along or with a parameter. If invoked alone, it will return the user who called the command’s current profile avatar in an embedded format. If the user adds in a parameter by mentioning another user with @[username] it will return the mentioned user’s avatar in an embed. The image itself is enlarged and will return in larger size than Discord sets images to.

Syntax:

$avatar

$avatar <user>

2.1.3 Invite

The invite command will allow a user to invite the bot into a server. Much like how the original admin invites the bot to a server, this will return a link to allow the user to get a generic invite for the discord bot. This invite does not have permission nodes on it and therefore a role will not be created for it. The invite will need the user who accesses the link to be an admin or owner of a server they intend to invite it to.

Syntax:

$invite

2.1.4 aHelp

A general command that allows admins or the owner of the server to see the admin help menu. Since this command is not defined immediately and has different syntax than just help, it is listed on the regular help embed. If a user without higher privileges invokes the command, they will be returned with an error stating they need higher permissions to use this command.

Syntax:

$aHelp

2.1.5 Define

Urban dictionary searches can be completed with the bot using this command. Searches typically follow the format of [command][search]. The search terms are required for this command and will give an error if there are no search terms included. The command searches the Urban Dictionary website and returns the first definition found to the user in an embedded format. Searches do not need to be single words and can be more than one word.

Syntax:

$define [search]

2.1.6 Gif

Gif is a type of picture format. The graphics interchange format allows small loops of a video segment to be played within its lifetime. The command allows users to search a term from the tenor library and return a .gif through this command. Gif searches will randomly pick between the first fifteen search results and give one back to the user as an embed. The command requires there to be a search term and the search term does not need to be only one word. However, if the search term is too lengthy then there will be no appropriate gifs to return to the user.

Syntax:

$gif [search]

2.2 Admin Commands

Admin commands may only be accessed by an admin or the owner of the Discord server the bot is residing in. Most of these commands are silent and will not provide users who are not admins with any error message if they attempt to use one of the commands.

2.2.1 Prefix

Each server has its own data that pertains only to that server. This command is included in this storage. Every server will be able to have an individual prefix based on what the admins or owner of the server set it to. The prefix is what is used to invoke voluntary user and admin commands. The prefix can be changed to any symbol or phrase that the management desires. All help menus will be updated with the prefix and any future commands must use the updated prefix to call any commands. If the prefix is changed to ‘!’ then the help command would require the user to use ‘!help’.

Syntax:

$prefix [new prefix]

2.2.2 Kick

Members that are needed to be removed from the server temporarily or members that are being removed but are still allowed back in can be removed with this command. This will allow the admin or owner to remove them with or without a removal message. The message will be documented in Discord’s audit log if one is included, as well as sending back a message with the reason kicked in chat with confirmation of the kick. To utilize this functionality the bot’s personal role must be higher than the role it is removing from the server. If it is not higher than the role, it will not be able to kick the user. This is due to how Discord’s permission hierarchy is instantiated.

Syntax:

$kick [user]

$kick [user] <message>

2.2.3 Ban

Members that are no longer wanted in the server can be banned with this command. Users that are targeted by this command will not be allowed to return to the server until they have been unbanned. This command may take parameters to include a reason for the user being banned. This will record in the ban section when the user that was banned is selected. Just like the kick command, the bot’s role will need to be above the highest role of the user it is kicking to successfully ban them. Users can be banned without the reason and will default to no reason provided.

Syntax:

$ban [user]

$ban [user] <reason>

2.3 Welcome Module

This module contains information for managing the welcome message of the server. The bot provides functionality that can welcome members who have just joined the server. It needs setup before it will work and is defaulted to off for every new server joined. Each server can have its own independent welcome settings and can be managed independently. Only admins and the owner of the server will have access to this module.

2.3.1 Welcome

The server needs a welcome message and channel to start. These can be set in any order. This message will be the message that displays when a new user joins the server. Since this message will be what is set in the server, the message can also mention the joining user. To mention the user that is joining in the message, it must contain <<mention>> where the message should mention the user. The command will return what the message was set to if the message is set properly.

Syntax:

$welcome [message]

2.3.2 Channel

For the welcome message to work properly, it needs to have a channel configured. Otherwise the message will have nowhere to be posted. Server welcome channel is defaulted to nothing and will need set before enabling the module. The command requires a channel ID that is referenced when mentioning a channel. To mention a channel the user must type #[channel name] or the command will not accept the argument. A confirmation of the selected channel will be returned upon a proper channel being chosen. If the bot has permission to post and see the channel, it will post in this channel when a new user joins the server.

Syntax:

$channel [channel name]

2.3.3 Enable

The welcome message can only be enabled once both the welcome message and welcome channel have been set for a server. Otherwise, an error will return to the user in the channel. The command can be enabled with various Boolean expressions. True|false|t|f|0|1 are all valid arguments for enabling the welcome message. Once this is turned on, and a new user joins the server, it will promptly put out a message in the desired channel with the desired output. Other settings must be configured with the previous commands.

Syntax:

$enable [boolean expression]

3. Listeners

These listeners are simple what they are called. They listen for certain cues and return a response based on receiving those cues. So, if one were configured to respond to people saying Hello with World, then anytime a user said Hello in the Discord server, the bot would respond “World”. These are on by default and are typically not inhibiting to the user experience.

3.1 Passive Listeners

These listeners react passively and do not need any active prefix to trigger. They will react and compute values if there are any tasks that they are set to do.

3.1.1 UWU Response

This response will react to any user that enters uwu into the server as a message. The uwu can be anywhere in the message. Some examples are “;alskdjf;alsjfkljsuwua’sdkjfa”, “ wow I uwu there”, and “u w U”. These all will return an embedded image to react to the message. It does not catch everything, such as no space characters that can be placed in between the letters. Future implementations look to add a rate of charges to fine the users and the debt will accrue over time and be stored in the database.

3.1.2 Bored Response

The response gained from this is not as extensive as the uwu response. Activation on this requires the user to simply input “bored” or “im bored” into the server. This will prompt a three stage reply that sleeps for a certain amount of time in between the three messages. The middle statement is taken from a file randomly and sends back the reply in the messages.

3.1.3 Microwave Response

This listener is relatively simple. If it sees that someone sends a message that contains the word microwave in it, it will return a link to a YouTube video with five hours of a bass boosted microwave going off.