
Laggron's Dumb Cogs Documentation

El Laggron

Aug 05, 2019

1	InstantCommands	3
1.1	Installation	3
1.2	Usage	4
1.3	Frequently Asked Questions	6
2	RoleInvite	9
2.1	Installation	9
2.2	Usage	9
2.3	Frequently Asked Question	13
3	Say	15
3.1	Installation	15
3.2	Usage	15
3.3	Frequently Asked Questions	17
4	WarnSystem	19
4.1	Installation	19
4.2	Usage	20
4.3	Commands	21
5	RoleInvite	33
5.1	API Reference	33
5.2	Errors	35
6	WarnSystem	37
6.1	API Reference	37
6.2	Errors	37
7	Indices and tables	39
	Python Module Index	41
	Index	43



31 ONLINE



Important: Please make sure that you first installed [Red DiscordBot V3](#). This is needed if you want to use these cogs.

CHAPTER 1

InstantCommands

Note: These docs refers to the version **1.0.0**. Make sure you're under the good version by typing `[p]cog update`.

This is the guide for the `instantcmd` cog. Everything you need is here.

`[p]` is considered as your prefix.

1.1 Installation

To install the cog, first load the downloader cog, included in core Red.:

```
[p]load downloader
```

Then you will need to install the Laggron's Dumb Cogs repository:

```
[p]repo add Laggrons-Dumb-Cogs https://github.com/retke/Laggrons-Dumb-Cogs v3
```

Finally, you can install the cog:

```
[p]cog install Laggrons-Dumb-Cogs instantcmd
```

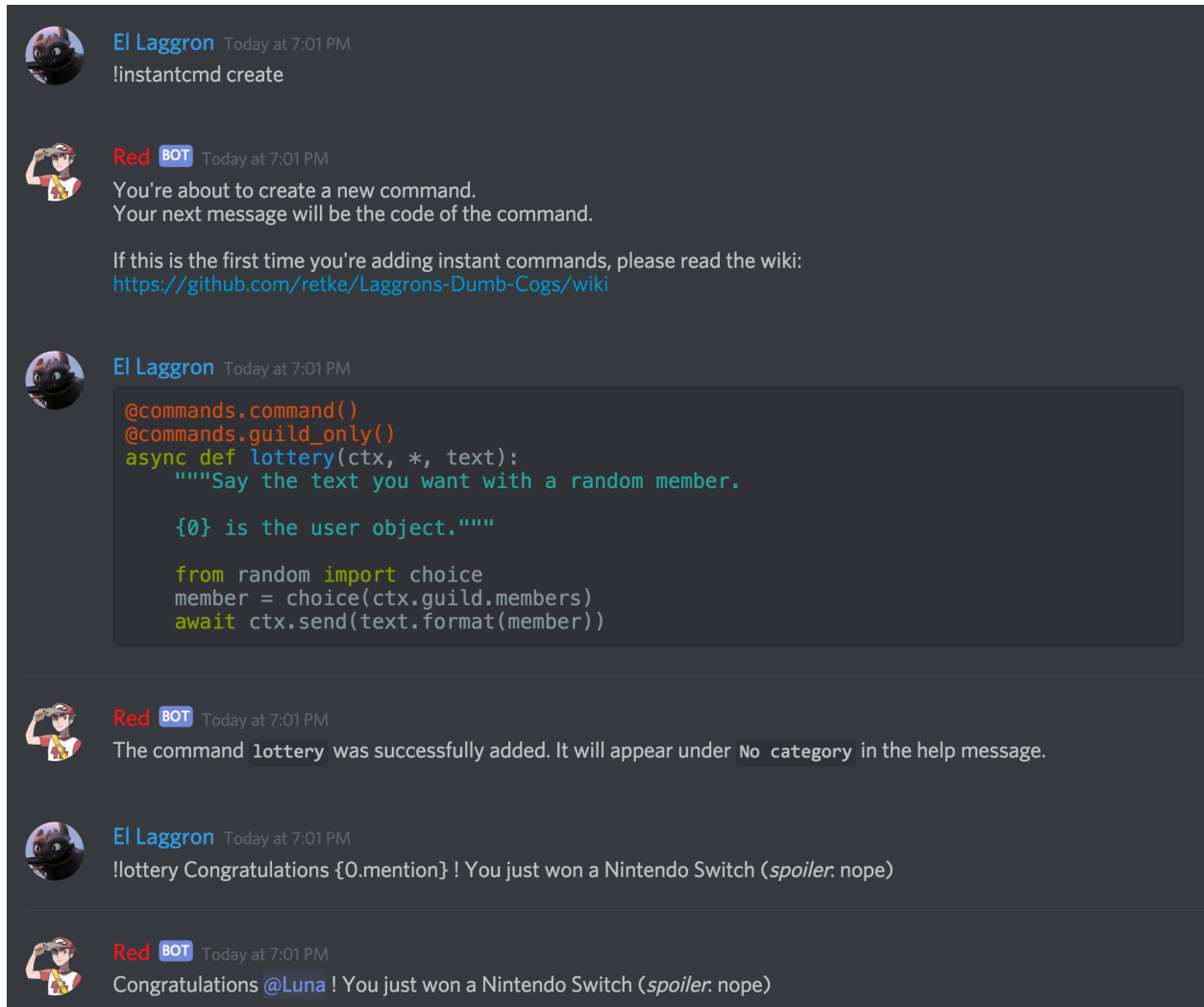
Warning: The cog is not loaded by default. To load it, type this:

```
[p]load instantcmd
```

1.2 Usage

InstantCommands is designed to create new commands and listeners directly from Discord. You just need basic Python and discord.py knowledge.

Here's an example of how it works:



The screenshot shows a Discord chat interface with a dark theme. It contains a sequence of messages:

- El Laggron** (Today at 7:01 PM): `!instantcmd create`
- Red BOT** (Today at 7:01 PM):
You're about to create a new command.
Your next message will be the code of the command.

If this is the first time you're adding instant commands, please read the wiki:
<https://github.com/retke/Laggrons-Dumb-Cogs/wiki>
- El Laggron** (Today at 7:01 PM):

```
@commands.command()
@commands.guild_only()
async def lottery(ctx, *, text):
    """Say the text you want with a random member.

    {0} is the user object."""

    from random import choice
    member = choice(ctx.guild.members)
    await ctx.send(text.format(member))
```
- Red BOT** (Today at 7:01 PM):
The command `lottery` was successfully added. It will appear under `No` category in the help message.
- El Laggron** (Today at 7:01 PM):
`!lottery Congratulations {0.mention} ! You just won a Nintendo Switch (spoiler: nope)`
- Red BOT** (Today at 7:01 PM):
Congratulations `@Luna` ! You just won a Nintendo Switch (spoiler: nope)

Here's a list of all commands of this cog:

1.2.1 instantcommand

Syntax:

```
[p] [instacmd|instantcmd|instantcommand]
```

Description

This is the main command used for setting up the code. It will be used for all other commands.

1.2.2 instantcommand create

Syntax:

```
[p]instantcommand create
```

Description

Creates a new command/listener from a code snippet.

You will be asked to give a code snippet which will contain your function. It can be a command (you will need to add the `commands` decorator) or a listener (your function name must correspond to an existing discord.py listener).

Tip: Here are some examples

```
@roleset.command()
@commands.command()
async def command(ctx, *, argument):
    """Say your text with some magic"""

    await ctx.send("You excepted to see your text, "
                  "but it was I, Dio!")

return command
```

```
async def on_reaction_add(reaction, user):
    if user.bot:
        return
    await reaction.message.add_reaction('')
    await reaction.message.channel.send("Here's some love for " + user.mention)

return on_reaction_add
```

Note: Here are the available values for your code snippet:

- bot (client object)
- discord
 - ext.commands
- asyncio
- inspect
- traceback
- random
- redbot.core
 - checks
 - Config
 - utils.chat_formatting.pagify

It isn't recommended to use the `Config` value for now. A future release should give a ready `Config.Config` object.

1.2.3 instantcommand delete

Syntax

```
[p]instantcommand [delete|del|remove] <name>
```

Description

Remove an instant command or a listener from what you registered before.

Arguments

- <name> The name of the command/listener.

1.2.4 instantcommand info

Syntax

```
[p]instantcommand info [command]
```

Description

List all existings commands and listeners created with InstantCommands.

You can give a command/listener name to show its source code.

Note: This only works with InstantCommands' commands and listeners.

Arguments

- [commands] The command/listener name to get the source code from. If not given, a list of existing commands and listeners created with the cog is shown.

1.3 Frequently Asked Questions

Note: Your question is not in the list or you got an unexpected issue?

You should join the [Discord server](#) or [post an issue](#) on the repo.

1.3.1 It's written in the help message that I can add a listener. How can I do so?

Instead of giving a `Command` object, just give a simple function (don't put the command decorator) and make sure its name is matching the lowercased [Discord API listeners](#).

Warning: Do not use the new `@commands.Cog.listener` decorator introduced in Red 3.1. The bot uses `bot.add_listener` which doesn't need a decorator.

1.3.2 My command was added but doesn't respond when invoked.

If a command is not invoked, this is most likely due to missing arguments. Please check that you only have the `ctx` argument and **no self argument**.

1.3.3 Can I use Config in my command?

Yes you can. The `Config` module is already imported, you just need to use it as in a cog.

..tip:: Here's an example

```
@commands.command(name="test")
async def my_command(ctx):
    config = Config.get_conf(cog_instance="InstantCommands", identifier=42)
    # use anything but 260 for the identifier
    # since it's the one used for the cog settings
    config.register_guild(**{
        "foo": None
    })

    await config.guild(ctx.guild).foo.set("bar")
    await ctx.send("Well done")

return my_command
```

1.3.4 How can limit a command for some users?

You can use the `checks` module, like in a normal cog.

Tip: Here's an example

```
@commands.command()
@checks.admin_or_permissions(administrator=True)
async def command(ctx):
    # your code

return command
```

1.3.5 How can I import a module without problem?

You can import your modules outside the function as you wish.

Tip: Here's an example

```
from redbot.core import modlog
import time

@commands.command()
async def command(ctx):
    # your code
```

(continues on next page)

(continued from previous page)

`return` command

CHAPTER 2

RoleInvite

Note: These docs refers to the version **2.0.0**. Make sure you're under the good version by typing `[p]cog update`.

This is the guide for the `roleinvite` cog. Everything you need is here.

`[p]` is considered as your prefix.

2.1 Installation

To install the cog, first load the downloader cog, included in core Red.:

```
[p]load downloader
```

Then you will need to install the Laggron's Dumb Cogs repository:

```
[p]repo add Laggrons-Dumb-Cogs https://github.com/retke/Laggrons-Dumb-Cogs v3
```

Finally, you can install the cog:

```
[p]cog install Laggrons-Dumb-Cogs roleinvite
```

Warning: The cog is not loaded by default. To load it, type this:

```
[p]load roleinvite
```

2.2 Usage

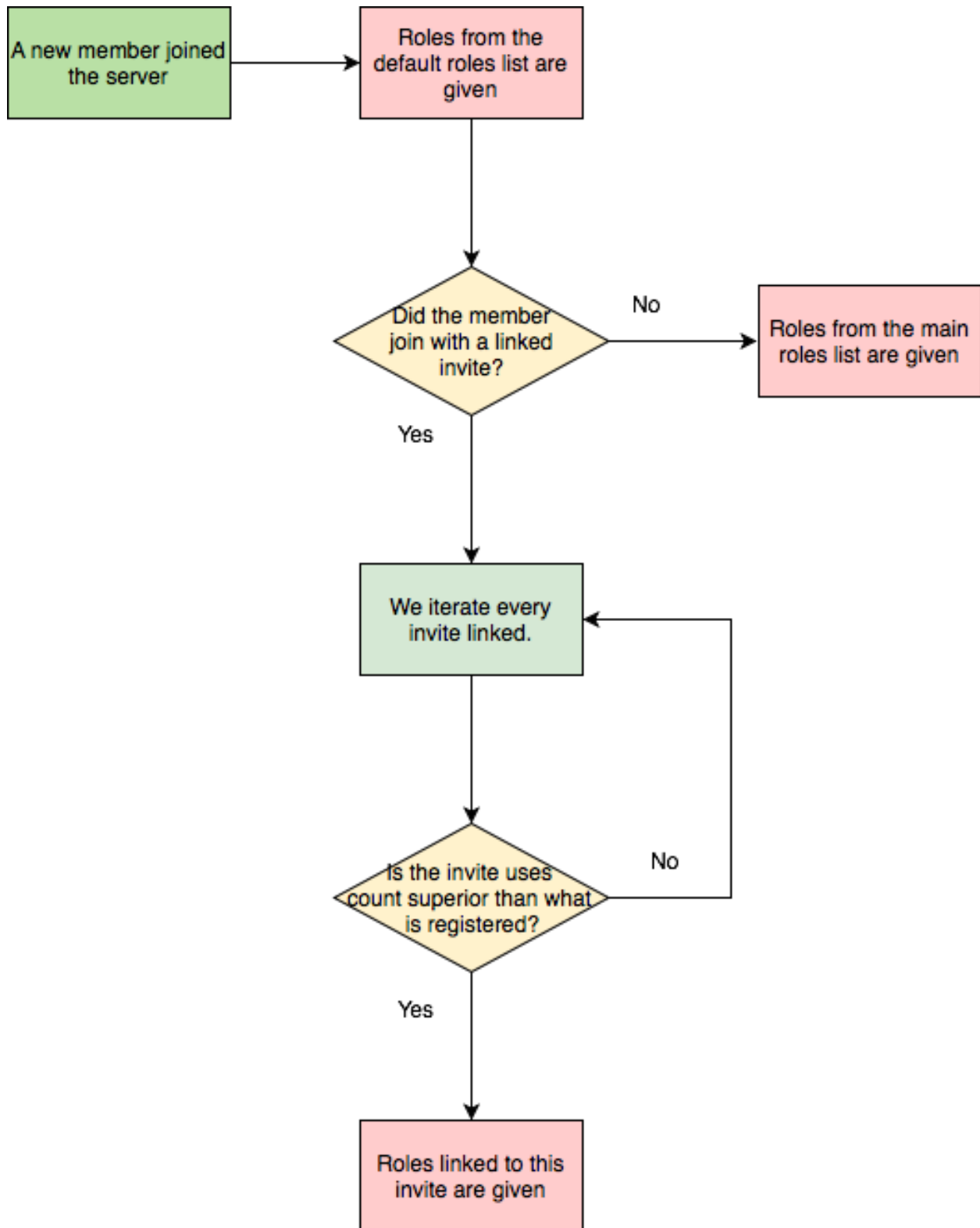
Before giving the commands list, I'd like to show you how the cog is working.

The cog works with what I call invite links. Each invite link is linked to one or more roles. This mean that, every time a new user join the server, if he used the invite A to join the server, he will get the list of roles linked to the invite A.

You can link many roles to multiple invites, so you can imagine something like “click **here** if you are an engineer, else click **here** if you're an architect”, and make roleinvite give the engineer or architect roles.

You can also link roles to default or main autorole. If you link roles to the main autorole, the new member will get these roles if he joined with an unlinked invite. If you link roles to the default autorole, new users will always get these roles, whatever invite he used.

Here's a schema for a better understanding:



Here's the list of all commands of this cog.

2.2.1 inviteset

Syntax:

```
[p]roleinviteset
```

Description

This is the main command used for setting up the code. It will be used for all other commands.

2.2.2 inviteset add

Syntax:

```
[p]roleinviteset add <invite|main|default> <role>
```

Description

Link a role to a Discord invite or a default autorole.

- If `invite` is specified (a discord invite link), a new invite link will be created with the role you gave.
- If `main` is specified, the role will be linked to the main autorole.
- If `default` is given, the role will be linked to the default autorole.

You can link more roles by typing the command with the same argument.

Arguments

- `<invite>` The object to link the role to.
 - If it is a Discord invite URL, the role will be linked to it.
 - If it is `main`, the role will be linked to the main autorole (role given if the invite used is not linked to any roles).
 - If it is `default`, the role will be linked to the default autorole (role always given, whatever invite the user used).
- `<role>` The role to be linked. Please give the **exact role name** or the ID.

2.2.3 inviteset remove

Syntax:

```
[p]roleinviteset remove <invite|main|default> [role]
```

Description

Unlink a role from an autorole. If the role is not given, the entire autorole will be removed.

Arguments

- `<invite>` The object that will be edited.
 - If it is a Discord invite URL, the role will be unlinked from it.
 - If it is `main`, the role will be unlinked from the main autorole.
 - If it is `default`, the role will be unlinked from the default autorole.

- `[role]` Optional. The role to be unlinked. Please give the **exact role name** or the ID. If not given, the entire autorole will be removed.

2.2.4 inviteset list

Syntax

```
[p]roleinviteset list
```

Description

List all of the existing autoroles on the guild, with their linked roles.

2.2.5 inviteset enable

Syntax

```
[p]roleinviteset enable
```

Description

Enable or disable the autorole system.

Note: If it was removed without your action, that means that the bot somehow lost its permissions. Make sure it has the good permissions and enable it again.

2.3 Frequently Asked Question

Note: Your question is not in the list or you got an unexpected issue?

You should join the [Discord server](#) or [post an issue](#) on the repo.

2.3.1 Can I make it so the bot adds x roles if the invite used is unknown?

Yes, by using the `main` value instead of using a discord invite when creating a new invite link. See *roleinviteset add* command's arguments for more informations.

2.3.2 Can I make it so the bot always adds x roles, regardless of the invite used?

Yes, by using the `default` value instead of using a discord invite when creating a new invite link. See *roleinviteset add* command's arguments for more informations.

2.3.3 Can I make a custom welcome message for each invite link?

Not for now. I'm thinking about interacting with another package for that, but that'll require an API, which is rare with cog creators, and creating my own welcomer system is a lot of work.

This may be available in a future release.

2.3.4 The bot suddenly stopped adding roles to the new members

The bot will automatically turn off the autorole system if he lose the `Manage server` or the `Add roles` permissions, which are absolutely necessary for the cog.

If you added the permissions back, enable the autorole again with the command *enable*.

2.3.5 Some roles are not added to the new members

This can happens if the role hierarchy is modified after the roles got linked. Remember that a bot/member can only add roles that are below him in the role hierarchy.

Modify the role hierarchy and make sure all necessary roles are **below** the bot's highest role. If it still doesn't work, try to link the role again.

2.3.6 An invite link was removed without any action

Invite links will be deleted if the invite doesn't exist anymore (manual delete or invite expired).

CHAPTER 3

Say

Note: These docs refers to the version **1.4.8**. Make sure you're under the good version by typing `[p]cog update`.

This is the guide for the `say` cog. Everything you need is here.

`[p]` is considered as your prefix.

3.1 Installation

To install the cog, first load the downloader cog, included in core Red.:

```
[p]load downloader
```

Then you will need to install the Laggron's Dumb Cogs repository:

```
[p]repo add Laggrons-Dumb-Cogs https://github.com/retke/Laggrons-Dumb-Cogs v3
```

Finally, you can install the cog:

```
[p]cog install Laggrons-Dumb-Cogs say
```

Warning: The cog is not loaded by default. To load it, type this:

```
[p]load say
```

3.2 Usage

Here's the list of all commands of this cog.

3.2.1 say

Syntax

```
[p]say [channel] <text>
```

Description

Make the bot say <text> in the channel you want. If specified, it is send in a different channel.

Tip: Examples

```
[p]say Hello it's me, Red bot!  
[p]say #general Hello, it's still me but from a different channel!
```

Arguments

- `[channel=ctx]`: The channel where the bot will send a message. Default to where you typed the command.
- `<text>`: The text that the bot will say in the channel.
- `attachment`: The file you want to make the bot send. This is optional.

3.2.2 saydelete

Syntax

```
[p][sayd|saydelete] [channel] <text>
```

Description

Exact same as `say` command, except it deletes your message.

Warning: The Manage message permission is needed for the bot to use this function.

3.2.3 interact

Syntax

```
[p]interact [channel]
```

Description

Starts a rift between the channel and your DMs. The messages you send to the bot in DM will make him post your messages in the channel. It will also post every message send in that time lapse.

Note: Click on the reaction on the first message to cancel the interaction.

Arguments

- `[channel=ctx]`: The channel where you want to start the interaction. Default to where you typed the command.

Tip: This can be used directly from DM. Then it will be cross-server.

Just make sure you give an ID as the channel. Giving the channel name can lead to a different server. Get the channel ID by enabling the developer mode (under Appearance section in the Discord user parameters), then right click on the channel and copy the ID.

3.3 Frequently Asked Questions

Note: Your question is not in the list or you got an unexpected issue?

You should join the [Discord server](#) or [post an issue](#) on the repo.

3.3.1 Can I send messages in another channel than the one where I typed the command?

Yes, by giving the channel as the first argument, like that:

```
[p]say #my-channel Hello!  
[p]say my-channel Hello!
```

You can also use the command in DM. It is recommended to give the channel ID as argument, since there may be many channels that has the same name in the bot servers.

```
[p]say 363031186504941578 Hello!
```

Tip: Get the ID by enabling the developer mode (User settings -> Appearance), then by right-clicking on the channel.

3.3.2 Can I make the bot upload links?

Yes, just attach a file to your message, it will be reposted with the same content. You can also add or not a comment.

3.3.3 Can I make the bot delete my message?

Yes, there's a command called *sayd* (for say delete) that will delete your message before posting.

3.3.4 My bot is slow to delete messages

If your bot is slow, that is an issue with your discord connection. Try changing the host machine.

Tip: You should use the *interact* command that let you tell what the bot should say in DM, so users won't see you typing.

3.3.5 I am not allowed to use the command

The command is only available for server owners and bot owner by default. You can modify this by using the core permissions cog.

CHAPTER 4

WarnSystem

Note: These docs refers to the version **1.0.1**. Make sure you're under the good version by typing `[p]cog update`.

This is the guide for the `warnsystem` cog. Everything you need is here.

`[p]` is considered as your prefix.

4.1 Installation

To install the cog, first load the downloader cog, included in core Red.:

```
[p]load downloader
```

Then you will need to install the Laggron's Dumb Cogs repository:

```
[p]repo add Laggrons-Dumb-Cogs https://github.com/retke/Laggrons-Dumb-Cogs v3
```

Finally, you can install the cog:

```
[p]cog install Laggrons-Dumb-Cogs warnsystem
```

Warning: The cog is not loaded by default. To load it, type this:

```
[p]load warnsystem
```

4.2 Usage

This cog is an alternative to the Mod core cog. It provides a moderation system similar to Dyno. Actions are stored and can be accessed at any time. This is the rewrite of the BetterMod cog for Red V3. Here is a quick start guide.

1. Define a modlog channel

You can define a channel where all actions will be logged, either with the `[p]warnset channel`` command or with the ```[p]modlogset modlog` command, from Modlog cog.

2. Set your moderators

All members with the moderator role will be able to use the `[p]warn` command. You can set the moderator and administrator role with the `[p]set modrole` and `[p]set adminrole` commands.

3. (Optional) Set up the mute role

The mute from WarnSystem uses roles instead of separate channel permissions. Type `[p]warnset mute` to create the mute role. It will be placed below the bot's top role and all channel permissions will be edited so those who have this role cannot send messages and add reactions.

You can edit this role as you want, as long as it is below the bot's top role so it can assign it to users.

4. Warn members

Once this is setup, moderators and administrators will be able to use the `[p]warn` command, with 5 different levels:

1. Simple warning
2. Server mute (can be temporary)
3. Kick
4. Softban (ban then quickly unban the member, to clean his messages)
5. Ban (can be temporary, and also ban members not on the server)

Each warn will send a DM to the warned member, a log in the modlog channel, then the bot will take actions. You can check, edit and delete a member's warnings with the `[p]warnings` command.

You now have the basic setup ready! If you want, you can setup more features for your bot:

- **Substitutions:** If you own a huge server, you might repeat yourself in the reasons of your warnings. You can setup substitutions, so you can include small words that will be replaced by a defined sentence. For example, if you set "Advertising for a Discord server." as a substitution of `ad`, type this: `[p]warn 3 @ElLaggron#0260 [ad] This is your last warning!` and the reason of the warn will be "Advertising for a Discord Server. This is your last warning!". Get started with the `[p]warnset substitutions group` command.
- **Reinvite:** Enabling this feature will try to send a DM to all unbanned members after their temporary ban, including an invite for your server. Note that the bot must share a server in common with the unbanned member.
- **Hierarchy:** To make sure your moderators doesn't abuse with their permissions, you can enable hierarchy protection. This means that the bot will block a moderator trying to warn a member higher than him in the role hierarchy, like with the manual Discord actions.
- **Multiple modlogs:** If you want to send all warnings, mutes, kicks and softban in a private channel, but you want to make the ban public, you can set a different channel for a specific warning level. Type `[p]warnset channel #your-channel 5` to make all bans go into that channel. Just change the number for the warn level.

- **Hide responsible moderator:** Sometimes, moderators want to keep their action anonymous to the warned member. If you want to stay transparent, type `[p]warnset showmod` to show the author of a warn to the warned member in DM.
- **Set number of days of messages to delete:** A Discord ban allows to set a specific number of days of messages sent by the banned member to delete, up to 7 days. By default, softbans will delete 7 days of messages and bans won't delete any. You can customize this with the `[p]warnset bandays` command.
- **Custom embed description:** If you want to customize your modlog and set your own sentence for logs sent to the modlog channel and to the warned member, you can do this with the `[p]warnset description` command.
- **Convert your old BetterMod logs:** If you're migrating to V3 and you were using the BetterMod cog on your V2 bot, you can migrate the logs for V3! Get the file of your modlog history (located at `/data/bettermod/history/<your server ID>.json`) and use the `[p]warnset convert` command.

4.3 Commands

Here is a list of all commands from this cog.

4.3.1 warn

Syntax

```
[p]warn
```

Description

The base command used to warn members. You must either have the moderator role, administrator role, have the administrator permission or be the server owner.

Warning: You **must** setup a modlog channel before using warn, either with the core Modlog cog (`[p]modlogset modlog`) or with WarnSystem (`[p]warnset channel`).

Each warning will be logged to the modlog channel, and a DM will be sent to the warned member. If the bot cannot send a message to that member (the member may have blocked the bot, disabled DMs from this server, or doesn't share a server in common with the bot), it will be showed in the modlog.

You can check the warnings set on a specific member later with the `[p]warnings` command. This command also allows to edit the reason of the warning, or delete them.

Tip: The warn level defaults to 1 if you omit it.

warn 1

Syntax

```
[p]warn <1|simple> <member> [reason]
```

Description

Sets a simple warning on a member. This does not take any action, but the warn will be showed to the member and stored.

Example

- `[p]warn 1 @El Laggron#0260 Rude behaviour.`

This warns El Laggron for the following reason: Rude behaviour.

Arguments

- `<member>`: The member to warn. Can either be a mention, the name + tag, the name, the nickname or an ID.
- `[reason]`: The reason of the warn. Omitting this will set the reason as “No reason set.”.

warn 2

Syntax

```
[p]warn <2|mute> <member> [duration] [reason]
```

Description

Mutes the member with a role on the server.

Warning: You **must** have the mute role setup. Use the `[p]warnset mute` command to create/assign the role.

The member will get the mute role for the specified time. You can edit this role as you like to allow him some channels for example. Removing his role manually will cancel his mute without problem, but the warn will still exist. Removing the warn with the `[p]warnings` command will also remove the role if needed.

You can set a duration to the mute with the first word of the reason, which should be a number followed by the unit. Examples:

- `20s = 20secs = 20seconds: 20 seconds`
- `5m = 5minutes = 5min: 5 minutes`
- `2h = 2hours = 2hrs: 2 hours`
- `1d = 1day: one day`
- `7d = 7days: a week`

You can also stack them like this:

- `5m30s: 5 minutes and 30 seconds`
- `1d12h: One day and a half`
- `1h45m: 1 hours and 45 minutes`

Examples

- `[p]warn 2 @El Laggron#0260 Hacked account.`

This will mute El Laggron for an undefined duration.

- `[p]warn 2 @El Laggron#0260 2h Spam for exp.`

This will mute El Laggron for two hours, then remove his role.

Arguments

- `<member>`: The member to warn. Can either be a mention, the name + tag, the name, the nickname or an ID.
- `[reason]`: The reason of the warn. Omitting this will set the reason as “No reason set.”.

warn 3

Syntax

```
[p]warn <3|kick> <member> [reason]
```

Description

Kicks the member from the server.

Example

- `[p]warn 3 @El Laggron#0260 Selfbot.`

This will just kick the member.

Arguments

- `<member>`: The member to warn. Can either be a mention, the name + tag, the name, the nickname or an ID.
- `[reason]`: The reason of the warn. Omitting this will set the reason as “No reason set.”.

warn 4

Syntax

```
[p]warn <4|softban> <member> [reason]
```

Description

Bans the member from the server, then unbans him, to mass delete his messages. This can be considered as a kick with a massive cleanup of messages.

The bot will delete 7 days of messages by default, this can be changed with the `[p]warnset bandays` command.

Example

- `[p]warn 4 @El Laggron#0260 NSFW in inappropriate channels.`

This will kick El Laggron and delete all of his messages sent in the last 7 days.

Arguments

- `<member>`: The member to warn. Can either be a mention, the name + tag, the name, the nickname or an ID.
- `[reason]`: The reason of the warn. Omitting this will set the reason as “No reason set.”.

warn 5

Syntax

```
[p]warn <5|ban> <member> [duration] [reason]
```

Description

Bans the member from the server, can be a temporary ban. It can also be a hackban (banning a member which is not on the server).

If you want to perform a hackban, get the ID of the user and provide it for the <member> argument. You can get a user ID by enabling the developer mode (User Settings > Appearance > Developer mode), then right-clicking on that user and clicking on “Copy ID”.

The bot won't delete any message by default, this can be changed with the `[p]warnset bandays` command.

You can set a duration to the mute with the first word of the reason, which should be a number followed by the unit. Examples:

- `20s = 20secs = 20seconds: 20 seconds`
- `5m = 5minutes = 5min: 5 minutes`
- `2h = 2hours = 2hrs: 2 hours`
- `1d = 1day: one day`
- `7d = 7days: a week`

You can also stack them like this:

- `5m30s: 5 minutes and 30 seconds`
- `1d12h: One day and a half`
- `1h45m: 1 hours and 45 minutes`

Attention: Deleting the warning through the `[p]warnings` command does not remove the ban.

Examples

- `[p]warn 5 @El Laggron#0260 Harassing`

Bans El Laggron forever from the server.

- `[p]warn 5 @El Laggron#0260 7d Doesn't respect the previous warnings`

Bans El Laggron for a week from the server, then unbans him.

- `[p]warn 5 348415857728159745 Advertising for a weird dating website, then leaves.`

Bans El Laggron forever while he is not on the server.

Arguments

- <member>: The member to warn. Can either be a mention, the name + tag, the name, the nickname or an ID.
- [reason]: The reason of the warn. Omitting this will set the reason as “No reason set.”.

4.3.2 masswarn

Syntax

`[p]masswarn`

Description

Warn multiple members at once. This advanced command allows you to filter members to warn with UNIX-like arguments, called flags.

Each “flag” is one more condition for the search. For example, `[p]masswarn --has-role "New Member" --joined-after "16 june 2019"` will filter the member who have the “New Member” role **and** who joined after the 16th of June of 2019. The search begins with all members on the server, then each condition is checked on each member to know if it should be kept in the masswarn or not.

You also have to tell to the bot what to do. Unlike the warn command where it takes actions, sends a message to the member and one in the modlog, you can decide what the bot should do, to make it faster or prevent spam.

- `--take-actions` will perform the action related to the warn (add the mute role, kick or ban a member...)
- `--log-modlog` will send a message in the modlog
- `--log-dm` will send a DM to the member

Warning: You have to put at least one of those flags.

You can then put the optional `--reason` flag to set the reason of the warning. Be sure to put it enclosed in quotes. If you're performing a level 2 or 5 warning, you can also use the `--time` flag to define the duration of the mute/ban if you want to make it temporary, the format of the time is the same as for the simple warnings.

Some flags needs an input with them, it can be a date, a set of roles, a regex expression... We will explain how input works for those flags. Note that if you need to put multiple words, you'll have to use quotes.

Date input

For the flags `--joined-before` and `--joined-after`, you will need to put a specific date. A lot of formats are supported, here are some examples:

- 27 june 2018
- 13/2/18
- august 2019 (will be the first day of the month)
- 2017 (will be the first day of the year)
- monday (will be the first monday of the month)
- 23 jun 12:00 (you can also specify the hour)
- Wednesday, 19th of September of 2018 (if you really want to lose time, that works too)

Role input

The flags `--has-role`, `--has-any-role`, `--has-all-roles`, `--has-none-roles`, `--above` and `--below` requires you to type one or more roles. You can provide the role ID or the role name, in quotes if there are spaces. Here are some examples:

- `--has-role Moderator`

- --has-any-role Member Staff "Nitro Booster" 168091848718417920
- --has-none-roles "Reddit Moderator"
- --below Administrators

Permission input

The flags --has-perm, --has-any-perm, --has-all-perms and --has-none-perms requires discord permissions, formatted as provided by the API. Here are the names you have to use:

General permissions:

- administrator
- view_audit_log
- manage_guild
- manage_roles
- manage_channels
- kick_members
- ban_members
- create_instant_invite
- change_nickname
- manage_nicknames
- manage_emojis
- manage_webhooks

Text permissions:

- read_messages
- read_message_history
- send_messages
- send_tts_messages
- attach_files
- embed_links
- external_emojis
- mention_everyone
- manage_messages
- add_reactions

Voice permissions:

- connect
- speak
- stream
- use_voice_activation
- priority_speaker
- move_members
- mute_members
- deafen_members

Here are some examples:

- --has-perm send_messages
- --has-any-perm manage_messages manage_channels manage_roles
- --has-none-perms administrator manage_guild
- --has-all-perms send_messages connect

Member input

The flags --include and --exclude requires you to pass multiple members, either with their name, their nickname, their name+tag, their ID or by mentioning them. Here are some examples:

- --include "El Laggron#0260" 133801473317404673 Twentysix

- `--exclude aikaterna#1393 "Kowlin, That silver Yuumi main"`

Regual expressions input (regex)

The flags `--name`, `--nickname` and `--display-name` requires regular expressions. Not going to explain how those work here, you can learn how to use those on [Python's guide](#) and test your expressions with [regex101](#). Just keep in mind you have to keep your expression enclosed in quotes.

Now it's time to list all of the flags.

- **Actions**

- `--take-action take-actions` *Defines if the bot should take an action (add the mute role, kick/ban the member)*
- `--send-dm` *Defines if the bot should send a DM to the warned members*
- `--send-modlog` *Defines if the bot should send a message in the modlog channel*
- `--reason <text>` *The reason of the masswarn, substitutions works*
- `--time --length` *The duration of the warn, for mutes and bans*

- **Member search**

- `--select [member, ...]` *Select multiple members to include in the masswarn, they are not affected by your search*
- `--exclude [member, ...]` *Select multiple members to exclude from the search, they won't be warned*
- `--everyone` *Includes everyone in the server; your search will therefore not be committed, the “--exclude” flag will also not be used*
- `--name <regex>` *Only includes the members which names validates to the given expression*
- `--nickname <regex>` *Only includes the members which nicknames validates to the given expression, this excludes members without nicknames*
- `--display-name <regex>` *Only includes the members which nicknames, or name if nickname isn't set, validates to the given expression*
- `--only-humans` *Excludes all bots from the search*
- `--only-bots` *Only includes bots in the search*
- `--joined-before <date>` *Members who joined after the given date will be excluded from the masswarn*
- `--joined-after <date>` *Members who joined before the given date will be excluded from the masswarn*
- `--last-njoins <number>` *Includes the last x members of the server; this is useful in case of a raid*
- `--first-njoins <number>` *Includes the first x members of the server; if you want to purge the elders you monster*

- **Permissions search**

- `--has-perm <permission>` *Includes the members with the given permission, this is based on roles, not channel permissions*
- `--has-any-perm [permission, ...]` *Includes the members who have any of the given permissions*

- `--has-all-perms [permission, ...]` *Includes the members who have all of the given permissions*
- `--has-none-perms [permission, ...]` *Include the members who have none of the given permissions*
- `--has-perm-int <number>` *Includes the members whose permission integer matches what you gave, you can calculate your permission integer on the 'permissions calculator' <<https://discordapi.com/permissions.html>> '_*

- **Role search**

- `--has-role <role>` *Includes the members who have the given role*
- `--has-any-role [role, ...]` *Includes the members who have any of the given roles*
- `--has-all-roles [role, ...]` *Includes the members who have all of the given roles*
- `--has-none-roles [role, ...]` *Include the members who have none of the given roles*
- `--has-no-role` *Excludes the members with any custom role*
- `--has-exactly-nroles <number>` *Includes the members who have the number of roles given, this doesn't count the @everyone role*
- `--has-more-than-nroles` *Includes the members who have more roles than the number given, this doesn't count the @everyone role*
- `--has-less-than-nroles` *Includes the members who have less roles than the number given, this doesn't count the @everyone role*
- `--above <role>` *Includes the members whose top role is above the given role*
- `--below <role>` *Includes the members whose top role is below the given role*

Enough info, time for explained examples.

- `[p]masswarn 2 --take-actions --send-dm --send-modlog --reason "Potential raid" --time 24h --joined-after "12 august 14:30" --has-no-roles --only-humans` This will mute for a day all members who joined after the 12th of august at 2:30 p.m. without roles and excluding bots. Everyone will receive a message and this will be logged in the modlog.
- `[p]masswarn 5 --take-actions --send-dm --reason "toxic potatoes" --has-role Starbucks` Just bans everone with the role "Starbucks"

4.3.3 warnset

Syntax

`[p]warnset`

Description

Base command used for all WarnSystem settings.

warnset settings

Syntax


```
[p]warnset settings
```

Description

Lists all settings defined on the current server.

warnset channel

Syntax

```
[p]warnset channel <channel> [level]
```

Description

Defines the modlog channel for the cog. This is a required step before warning members.

Note: You can also use the core Red modlog by loading the modlogs cog, then using the `[p]modlogset modlog` command.

If you want to set a different modlog for a specific warning level (like, sending ban warnings in a different channel), you can provide the warning level after your channel to set it as the modlog channel for this specific warning level.

Arguments

- `<channel>`: The text channel where the modlog will be set.
- `[level]`: The warning level associated to the channel. If this is not provided, the channel will be set as the default modlog channel.

warnset mute

Syntax

```
[p]warnset mute [role]
```

Description

Creates a role used for muting the members, or set an existing one as the mute role. If you don't provide any role, the bot will create one below his top role, then deny the "Send messages" and "Add reactions" on all text channels. **Editing all channels takes a long time, depending on the number of text channels you have on the server**, so don't worry if nothing happens for about 30 seconds, it's doing the setup for the mute.

You can also provide an existing role to set it as the new mute role. **Permissions won't be modified in any channel in that case**, so make sure you have the right permissions setup for that role.

Arguments

- `[role]`: The exact name of an existing role to set it as the mute role. If this is omitted, a new role will be created.

warnset reinvoke

Syntax

```
[p]warnset reinvoke [enable]
```

Description

Enables or disables the DM sent to unbanned members. If you enable this, make sure the bot has the permission to create new invites.

This is enabled by default.

Arguments

- `[enable]`: The new status to set. If omitted, the bot will display the current setting and show how to reverse it.

warnset hierarchy

Syntax

```
[p]warnset hierarchy [enable]
```

Description

Enables or disables the hierarchy respect. If you enable this, the bot will make sure the moderator is allowed to warn someone with the Discord hierarchy rules (cannot warn someone if the warned member has a role equal or higher than the moderator's top role).

This is disabled by default.

Arguments

- `[enable]`: The new status to set. If omitted, the bot will display the current setting and show how to reverse it.

warnset showmod

Syntax

```
[p]warnset showmod [enable]
```

Description

Toggles if the bot should show or hide the responsible moderator of a warn to the warned member in DM.

This is disabled by default.

Arguments

- `[enable]`: The new status to set. If omitted, the bot will display the current setting and show how to reverse it.

warnset bandays

Syntax

```
[p]warnset bandays <ban_type> <days>
```

Description

Defines how many days of messages should be deleted when a member is banned or softbanned. The number of days can be between 1 and 7. You can set 0 to disable message deletion for the bans, not for softbans.

Arguments

- `<ban_type>`: The type of ban that should be edited. Either `ban` or `softban`.
- `<days>`: The number of days of messages that should be deleted. Between 1 and 7 only. 0 to disable for bans.

warnset substitutions

Syntax

```
[p]warnset substitutions add <name> <text>
[p]warnset substitutions [delete|del] <name>
[p]warnset substitutions list
```

Description

Group command for managing the substitutions. A substitution is used to replace a small word in brackets by a long sentence in your warn reason, to avoid repetitions when taking actions.

Use `[p]warnset substitutions add <name> <text>` to create a substitution, where `<name>` is the keyword and `<text>` is what will replace the keyword.

Use `[p]warnset delete` to delete a substitution and `[p]warnset list` to list them.

Example

```
[p]warnset substitutions add lastwarn This is your last warning!
This creates a substitution with the keyword lastwarn.
```

```
[p]warn 3 @El Laggron#0260 Racist insults. [lastwarn]
The reason of this warn will be: Racist insults. This is your last warning!
```

warnset description

Syntax

```
[p]warnset description <level> <destination> <description>
```

Description

Edits the description of an embed for the modlog or the warned member. The default description for the modlog is “A member got a level (x) warning.”, for the member, it is “The moderation team set you a level (x) warning.”.

You can use the following keys in your custom description:

- `{invite}`: Generates an invite for the server and place it.
- `{member}`: The warned member. You can use attributes such as `{member.name}`, `{member.id}`, `{member.nick}`...
- `{mod}`: The responsible mod of a warn. You can use the same attributes as for `{member}`.
- `{duration}`: The duration of a mute/ban if set.
- `{time}`: The current date and time.

Arguments

- `<level>`: The level of the warn to edit.

- `<destination>`: Either `user` for the warned member or `modlog` for the modlog.
- `<description>`: The new description.

warnset convert

Syntax

```
[p]warnset convert <path>
```

Description

Converts a V2 BetterMod history file to migrate its logs to WarnSystem V3.

The history file is located at the following path: `Red-DiscordBot/data/bettermod/history/<server ID>.json`. You can grab your server ID with the `[p]serverinfo` command.

You can decide to append or overwrite the logs to the current logs through the guided configuration. Append will get the logs and add them, while overwrite will reset the current logs and replace them with the migrated ones.

Example

- ```
[p]warnset convert /home/laggron/Desktop/Red-DiscordBot/data/bettermod/history/
↪363008468602454017.json
```

#### Arguments

- `<path>`: The path to your history file.

### 4.3.4 warnsysteminfo

---

**Note:** This command is locked to the bot owner.

---

#### Syntax

```
[p]warnsysteminfo
```

#### Description

Shows multiple informations about WarnSystem such as its author, its version, the link for the Github repository, the Discord server and the documentation, and a link for my Patreon if you want to support my work ;)

## 5.1 API Reference

**class** `roleinvite.api.API` (*bot, config*)

Interact with RoleInvite from your cog.

To import the cog and use the functions, type this in your code:

```
roleinvite = bot.get_cog('RoleInvite').api
```

**Warning:** If `roleinvite` is `None`, the cog is not loaded/installed. You won't be able to interact with the API at this point.

---

**Tip:** You can get the cog version by doing this

```
version = bot.get_cog('RoleInvite').__version__
```

---

**coroutine** `add_invite` (*guild: discord.guild.Guild, invite: str, roles: list*) → bool

Add an invite link to the autorole system.

**Parameters**

- **guild** (`discord.Guild`) – The guild to get the invites from.
- **invite** (`str`) – The invite link to create/extend. Give `main` or `default` if you want to edit the main/default autorole system.
- **roles** (`list`) – A list of roles ID to add to the roles list.

**Returns** `True` if successful

**Return type** `bool`

## Raises

- `NotInvite` – The invite given is not a discord invite, not is is main/default.
- `CannotGetInvites` – The bot doesn't have the permission to get the guild's invites
- `EmptyRolesList` – The list of roles given is empty
- `InviteNotFound` – The invite given doesn't exist in the guild.

**escape\_invite\_links** (*text: str*) → str

Return a Discord invite link that won't show an embed

**Parameters** `text` (*str*) – The text which needs to have invite links previews removes

**Returns** `text` – The cleared text

**Return type** `str`

**coroutine get\_invites** (*guild*) → dict

Return a `list` of the invites linked to the autorole system of the guild.

**Parameters** `guild` (`discord.Guild`) – The guild to get the invites from.

## Returns

A `dict` of invites linked to any role on the guild.

Example

```
{
 "main" : {
 "roles" : [
 987654321234567890
]
 },
 "https://discord.gg/example" : {
 "roles" : [
 012345678987654321,
 987654321234567890
],
 "uses" : 42
 }
}
```

**Return type** `dict`

**coroutine remove\_invite** (*guild: discord.guild.Guild, invite: str, roles: list = []*) → bool

Remove a `list` of roles from the invite links.

## Parameters

- `guild` (`discord.Guild`) – The guild to get the invites from.
- `roles` (`list`) – A `py:class:list` of roles ID to remove from the roles list. If it's empty, it will remove the invite from the autorole system.
- `invite` (`:py:class`str``) – The invite to remove roles from. Give `main` or `default` to edit the main/default autorole system.

**Returns** `True` if successful.

**Return type** `bool`

**Raises** `KeyError` – The invite given doesn't exist.

**coroutine** `update_invites()` → dict

Update all invites registered to keep their uses count good.

This is usually called on cog load since these values could have been modified while the bot or the cog was offline.

#### Returns

The updated dictionary.

---

**Note:** The value `enabled` may have been switched to `False` if the `manage_guild` permission was lost on the guild.

---

**Return type** dict

## 5.2 Errors

Custom error handling used for the cog and the API

If you need to prevent an exception, do it like this

```
errors = bot.get_cog('RoleInvite').errors

try:
 await api.add_invite(
 ctx.guild, 'main', [42]
)
except errors.CannotAddRole:
 print("Missing permissions")
except InviteNotFound:
 print("Invalid invite")
except:
 # occurs for any exception
 print("Fatal error")
else:
 # executed if the try succeeded
 print("All good")
finally:
 # always executed
 print("End of function")
```

**exception** `roleinvite.errors.CannotAddRole`

The bot isn't allowed to give a role. The role hierarchy was modified or a 3rd party module added the role without check.

**exception** `roleinvite.errors.CannotGetInvites`

The bot isn't allowed to get the guild invites. Manage server permission is needed.

**exception** `roleinvite.errors.EmptyRolesList`

The list of roles that needs to be linked to an invite is empty.

**exception** `roleinvite.errors.InviteNotFound`

The invite sent isn't in the guild's invite list.

**exception** `roleinvite.errors.NotInvite`

The invite sent is not found as a discord.Invite object.





## CHAPTER 6

---

WarnSystem

---

### 6.1 API Reference

### 6.2 Errors



## CHAPTER 7

---

### Indices and tables

---

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



**r**

`roleinvite.errors`, [35](#)



### A

`add_invite()` (*roleinvite.api.API method*), [33](#)  
*API (class in roleinvite.api)*, [33](#)

### C

`CannotAddRole`, [35](#)  
`CannotGetInvites`, [35](#)

### E

`EmptyRolesList`, [35](#)  
`escape_invite_links()` (*roleinvite.api.API method*), [34](#)

### G

`get_invites()` (*roleinvite.api.API method*), [34](#)

### I

`InviteNotFound`, [35](#)

### N

`NotInvite`, [35](#)

### R

`remove_invite()` (*roleinvite.api.API method*), [34](#)  
`roleinvite.errors` (*module*), [35](#)

### U

`update_invites()` (*roleinvite.api.API method*), [34](#)