**NOTES** [things to do]

* **Look into pooled connections for fast querying**
* **Consider writing bot in Rust with Serenity and use sqlx for compile-time checked queries**
* **At each step, consider what data you can leverage for analytics and AI purpose. Perhaps a recommendation system for joining a certain ‘group’ in a server?**
* [**https://www.pinecone.io/learn/vector-database/**](https://www.pinecone.io/learn/vector-database/)

**List of commands:**

**Admin:**

**!create** [group\_type, channel\_id, or @role\_id] [group\_color] [emoji]   
[std\_admin\_msg]

**!set type** [group\_name] [group\_type]

**!set color** [group\_name] [group\_color]

**!set emoji** [group\_name] [group\_emoji]

**!disband** [channel\_id]

**User:**

**!form** [group\_type] [group\_name] (optional message)

**!disband**

**!set** [command]  
Parameters: [group\_type], [group\_type\_color, default=], [emoji, default=👍], [std\_admin\_msg, default=]

!set Command List:

* !set type
* !set color \*\*\*\*\*
* !set emoji \*\*\*\*\*

!set Admin Command(s):

1. Set a Group Type   
**Command:**   
!set type [group\_type] [group\_type\_color, default=], [emoji, default=👍]  
[std\_admin\_msg, default=“”]

* example admin usage:

!set team yellow

\*\*This is a project team group.\*\*

!set User Command(s):

**!form** [command]  
Parameters: [group\_type], [group\_name]

!form Command List:

* !form group

!form Admin Command(s):

!form User Command(s):

1. Form a Group  
**Command:**

!form group [group\_type] [group\_name]  
(optional variable user message)

* example user usage:

!form team Jack’s Project

Welcome to my project’s team! Here are some resources that you might need to get started:

* xyz.com
* www.com

**Standard Message Format:**

Welcome to our [group\_type] “[group\_name]”!   
Please react with [emoji, default=👍] to board this group.

(optional [std\_admin\_msg])

(optional variable user message)

[group\_type] by @user

* example output: *font size and spacing may vary*

Welcome to my team “Jack’s Project”!   
Please react with 👍 to board this group.

**This is a project team group.**  
Welcome to my project’s team! Here are some resources that you might need to get started:

* xyz.com
* www.com

Team by @user

**!disband** [command]

[group\_owner], [group\_type], [group\_name], [message\_id], [role\_id], [channel\_id]

Admin Command(s):

1. Disband a Group  
**Command:**

!disband [group\_name]

User Command(s):

1. Disband a Group  
**Command:**

!disband

**Function:** Only [group\_owner] can call this method inside the [channel\_id] where the appropriate group exists. When invoked, entire channel data is archived, and sent to the [group\_owner] in a private DM. Then, the [role\_id] and [channel\_id] is erased from the channel. The message with [message\_id] is cleared from the post location. The admin can choose to be notified that a group has been disbanded in the form of a log.

**!set** [database dev-guide]

Relevant Data: [server\_id], [group\_type], [group\_type\_color, default=], [emoji, default=👍], [std\_admin\_msg, default=]

!set Command List:

* !set [group\_type]

!set Admin Command(s):

1. Set a Group Type   
**Command:**   
!set type [group\_type] [group\_type\_color, default=] [emoji, default=👍]  
[std\_admin\_msg, default=“”]

**Method:** A table stores predefined server\_id, group\_type(s), group\_type\_color(s), emoji and std\_admin\_msg set by the admin. When a **user** invokes the !form group\_type command, it references a group\_type and this table provides the associated reaction role emoji, group\_type\_color (translates to role color) and std\_admin\_msg.

**Table Name:** ‘form\_type’

**Data Variables:**   
[server\_id] – Reference to server where command was invoked.   
[group\_type] – The type/category of group set by admin.   
[group\_type\_color, default=] – Role color associated with that group type.  
[emoji, default=👍] – Reaction role emoji associated with type.   
[std\_admin\_msg, default=“”] – Standard message by admin for that type.

**Table Relationships:** [server\_id](Foreign Key) references primary key in the Servers Index.

User Command(s):

**!form** [database dev-guide]

[group\_owner], [group\_type], [group\_type\_color, default=], [group\_name],

[emoji, default=👍], [std\_admin\_msg, default=], [message\_id], [role\_id], [channel\_id]

Admin Command(s):

User Command(s):

1. Form a Group  
**Command:**

!form group [group\_type] [group\_name]  
(optional variable user message)

**Method:** A table stores server\_id (foreign key), group\_owner, group\_name, role\_id, message\_id, channel\_id and created\_date. When a user invokes !form [group\_type], it creates a new [role\_id] and [channel\_id] in the server, both with the same [group\_name]. The user’s id is stored as [group\_owner]. The associated role color is updated to the [group\_type\_color] set by admin. The channel’s permissions are set to private, only viewable only by users with corresponding [role\_id]. A message is also posted with a reaction role [emoji] which, when reacted to, assigns users to the new role. A [created\_date] is formed to keep track of data retention/cleanup tasks. After 30 days of inactivity within a group, an admin and/or [group\_owner] will have 7-day grace period to keep the group up or it will be automatically removed.

**Table Name:** ‘form\_group\_type’  
**Data Variables:**

[server\_id] – Reference to server where command was invoked.

[group\_owner] – Reference to user who created the group.  
[group\_name] – Name of the group, role, and channel.   
[role\_id] – Role ID create for the group\_name.   
[message\_id] – ID of the message with the reaction role.   
[channel\_id] – ID of the private channel created for the group.

[created\_date] – Date the group was created.  
**Table Relationships:** [server\_id](Foreign Key) references primary key in the Servers Index. [group\_type](Foreign Key) references ‘group\_type’ in the ‘form\_type’ table, indicating the type of group, [std\_admin\_msg] and [emoji] associated with type.

**!disband**[database dev-guide]

[group\_owner], [group\_type], [group\_type\_color, default=], [group\_name],

[emoji, default=👍], [std\_admin\_msg, default=], [message\_id], [role\_id], [channel\_id]

Admin Command(s):

User Command(s):

1. Disband a Group  
**Command:**

!disband

**Method:** In the database table ‘form\_group\_type’, any rows with [channel\_id] (foreign key) are removed.

**Table Name:** ‘form\_group\_type’  
**Data Variables:**

[server\_id] – Reference to server where command was invoked.

[group\_owner] – Reference to user who created the group.  
[group\_name] – Name of the group, role, and channel.  
[role\_id] – Role ID create for the group\_name.   
[message\_id] – ID of the message with the reaction role.   
[channel\_id] – ID of the private channel created for the group.  
**Table Relationships:** [server\_id](Foreign Key) references primary key in the Servers Index. [group\_type](Foreign Key) references ‘group\_type’ in the ‘form\_type’ table, indicating the type of group, [std\_admin\_msg] and [emoji] associated with type.