

Wordlists (Banned Words)

REQUIRES: Manage messages

 Banned word example	 Banned word example with variables.
--	---

Banword/Unbanword

Bans a word by adding it to a specified wordlist. If no word list is specified, then banned words will be added to the **"local"** wordlist. You may upload a text file (.txt) with a list of words **separated by a new line** instead.

Command	Descriptive Example
<code>!banword word/phrase</code>	<code>!banword potato</code> - Bans the word and adds it to the "local" wordlist
<code>!banword name ListName word/phrase</code>	<code>!banword name NaughtyWords nibba</code> - Unbans the word and adds it to the wordlist
<code>!banword insert your regex here %regex</code>	<code>!banword b[il1]+t{2,}(er a)' %regex</code> - Bans a custom regex <div style="background-color: #ffe4c4; padding: 5px;">Regex is a complicated matter, which may result in unexpected filter behavior.</div>

!unbanword <i>word/phrase</i>	! unbanword potato - Unbans the word and removes it from the "local" wordlist
!unbanword name <i>ListName word/phrase</i>	! banword name NaughtyWords nibba - Unbans the word and removes it from the word list
!unbanword id <i>WordID#</i>	! unbanword is 12345 - Unbans the word based on its unique ID number

Variables

Variable	Description
%whole	Only the exact match
%begin	Only from the beginning of words
%end	Only from the end of words
%case	Case-sensitive (100% match only)
%mute	Auto permanent mute
%mute #	Auto mute for certain amount of minutes
%warn	Auto warn
%kick	Auto kick
%softban	Auto softban
%ban	Auto ban
%name	Only applies to names/nicknames
%regex EXAMPLE: !banword b[il1st]{2,}(er a)' %regex	Applies your given regex EXAMPLE: Blocks common variations of the word bitter

<code>%notify</code> (REQUIRES: Notifications logging enabled)	Posts a log through the Notifications logging channel
<code>%notifynd</code> (REQUIRES: Notifications logging enabled)	Posts a log through the Notifications logging channel <u>This will not delete the banned word.</u>

You can filter usernames based on your banned words with `! auto name` which checks whenever someone joins your server or whenever a name is updated.

[\(Learn more\)](#)

Filter

Command	Description
<code>!filter ignore [invert] #channel url/invite/text/auto/activity</code>	<pre>!filter ignore #general url</pre> <ul style="list-style-type: none"> - Ignores url filters in the channel <hr/> <pre>!filter ignore invert #general text</pre> <ul style="list-style-type: none"> - Ignores text filters in all channels except for #general
<code>!filter recog [invert] #channel url/invite/text/auto/activity</code>	<pre>!filter recog #general url</pre> <ul style="list-style-type: none"> - Re-enables url filters in the channel <hr/> <pre>!filter recog invert #general invite</pre> <ul style="list-style-type: none"> - Re-enables url filters in all channels except for #general
<code>!filter view [ignores]</code>	<pre>!filter view</pre> <ul style="list-style-type: none"> - Displays currently active filters in the server <hr/> <pre>!filter view ignores</pre> <ul style="list-style-type: none"> - Displays all currently active filter ignores in the server
<code>!filter url/invite</code>	<pre>!filter url</pre> <pre>!filter invite</pre> <ul style="list-style-type: none"> - Toggles the url or invite filter

Wordlist

Command	Descriptive Example
!wordlist list	<pre>! wordlist list</pre> <p>- Displays the list of current wordlists in the server along with their info</p>
!wordlist view <i>name</i> [<i>serverid</i>] [readable]	<pre>! wordlist view NaughtyWords 270438133584232449 readable</pre> <p>- Sends a list of banned words in an editable .json file</p> <p>It is entirely optional to use a server ID and "readable" in the command. The server ID is primarily meant for cases in which the wordlist did not originate in your server, and that the wordlist is added to your server with the <code>! wordlist add</code> command.</p> <p>Typing "readable" at the end of the command will output a .txt file instead of .json. This is easier for when you only want to look at it and not re-upload an edited version of the file.</p>
!wordlist create ListName	<pre>! wordlist create NaughtyWords</pre> <p>- Creates a new wordlist with a name</p> <p>It is possible to use an edited .json file, template can be found with <code>! wordlist formfile</code>, to ban words in bulk</p>
!wordlist delete <i>ListName</i>	<pre>! wordlist delete NaughtyWords</pre> <p>- Deleted the wordlist form the server</p>
!wordlist formfile	<pre>! wordlist formfile</pre> <p>- Sends a .json template for banning words in bulk when uploading a .json file</p> <p>This is useful for when using the <code>! wordlist create</code> command.</p>
!wordlist add/rem <i>name serverid</i>	<pre>! wordlist add NaughtyWords 270438133584232449</pre> <p>- Removes another server's non-locked wordlist to yours</p> <hr/> <pre>! wordlist rem NaughtyWords 270438133584232449</pre> <p>- Removes another server's non-locked wordlist from yours</p>
!wordlist merge/unmerge <i>name serverid</i>	<pre>! wordlist merge NaughtyWords 270438133584232449</pre> <p>-Merges another server's non-locked wordlist to yours</p> <hr/> <pre>! wordlist unmerge NaughtyWords 270438133584232449</pre> <p>- Unmerges another server's non-locked wordlist from yours</p>

Updated Sat, Oct 23, 2021 11:40 PM by [Fairy](#)