# nmathe.ws

Table of Contents

# Saber bot

Invite to Discord | Saber on GitHub | Get Support

Features

- Display events on dedicated schedule channels.
- Send announcements when event begins / ends.
- Send reminders before an event begins.
- Configure timezone per schedule.
- Sync schedules to Google Calendars
- Users can RSVP to events
- Custom command prefix

Roadmap

- Bug Fixes and Patches
- Feature revisions

Command	Action
init	Create a new schedule
create	Add a new event to a preexisting schedule

Command	Action
edit	Modify a preexisting event
delete	Delete an event or schedule
guild	Display and modify guild-wide settings
config	Display and modify schedule settings
zones	Show all timezones that the bot understands
help	Direct message the user a list of commands with additional information
sync	Sync a schedule to load events from a Google Calendar calendar
oauth	Authorize Saber access to your Google Calendar calendars
test	Test an event's notification message
sort	Chronologically reorder the events on the schedule
list	View list of users who have rsvp'ed for an event
skip	Cancel the next occurrence of an event
schedules	List guild schedules
events	List guild events

The default command prefix is `!`, however `@Saber` can also be used to trigger bot commands.

# Quick Start

1. Invite Saber-Bot and create a channel named **#saber_control**
2. Use `!init in #saber_control` to create a schedule.
    - If a channel named **#new_schedule** is not created, give Saber-Bot Manage Channel permissions.
3. Use `!config #new_schedule` to view schedule settings
4. Use `!guild prefix $` to change the command prefix to `$`.
5. Use `$zones "US"` to view available timezone codes for the United States.
    - Experiment with this command to find a timezone which best represents your locale.

6. Use `$config #new_schedule zone [timezone]`, where `[timezone]` is the timezone you found in step 5.
    - It is best to always configure your schedule's timezone before adding events.
7. Use `$create #new_schedule "Hi mom!" 5:45pm` to create a new event
    - If no message appears in #new_schedule, check to insure Saber-Bot has read, write, and manage message perms.
8. To delete the test schedule and event use `$delete #new_schedule` to remove the schedule.
    - Alternatively, the schedule channel can be manually deleted for the same affect. The same goes for the event messages.

Make use of the `help` command, read the docs, or ask in the support server to learn more about the commands. I hope @Saber fulfills your guild's scheduling needs!

# Server Setup

Saber bot is designed to save events of schedule on dedicated "schedule channels."

Scheduling channels require Saber be given some permissions, which should already have been requested if Saber joined your server using the above oath link. Insure that Saber has the Manage Channels permission and the ability to read/write on any channel you plan to have her announce on.

In addition to schedule channels, Saber bot is configured to listen for scheduling commands only on a dedicated channel. Create a channel with the name "saber_control," and allow Saber to view and send messages to it.

After the first `guild` command is issued in `#saber_control`, the name of the control channel can be customized without affecting the behavior of the bot.

# Commands

Saber commands are of the form `!command <arguments>`; an argument is any space separated, quotation enclosed phrase -- with the exception of single word phrases which may drop the quotations.

Here's an example : `!create #event_schedule "Rampage on Monday" 8:00am 9:00pm`. Saber will parse out four tokens from that command: `#event_schedule`, `Rampage on Monday`, `8:00am`, and `9:00pm`.

For each of the following commands, `<argument>` denotes an argument and `[argument(s)]` denotes optional arguments.

- `!create <#channel> <title> <start> [<end> <option(s)>]`

  Every new entry MUST be initialized with a title and a starting time. Start and end times should be of form h:mm with optional am/pm appended on the end (ex. `10:20pm`). The `<end>` parameter may be ommitted to create an event which does not end at any particular time.

  Events can be created with a variety of extra options by adding specific keywords followed by a required value to the `<option(s)>` section. Comments are the only exception to the previous rule. Comments may be added to an event's description by simply appending quotation enclosed phrases to the end of the command (see examples). A list of `<option>` keywords can be found later in this document.

  - Ex1. `!create #event_schedule "Party in the Guild Hall" 19:00 2:00 date 04/10`
  - Ex2. `!create "#event_channel Reminders" "Sign up for Raids" 4:00pm`
  - Ex3. `!create "#event_channel Raids" "Weekly Raid Event" 7:00pm 12:00pm repeat "Fr,Sa" "Healers and tanks always in demand." "DM @RaidCaptain with your role and level if attending."`

- `!edit <id> <option> <changes>`

  This command can be used to change any parameter of an event given the event's ID. The first argument is always the event ID, the second argument should be the keyword for the parameter you wish to change, and the third argument is the new value. The edit command uses the same option keywords as the create command.

  `comment` is the only exception to this rule. When 'comment' is the second argument, the third argument needs to be either `add` , `remove`, or `swap` (to denote the which action to take). If using the `add` or `remove` actions, a fourth argument representing the comment number or new comment needs to be provided. If using the `swap` action, two additional arguments representing the comment numbers to swap are required. See examples below.

  - Ex1: `!edit 3fa021 comment add "Attendance is mandatory"`
  - Ex2: `!edit 0abf58 start 21:15`

  - Ex3: `!edit 49a83f end 2:15pm`
  - Ex4: `!edit 80c012 comment remove 1`
  - Ex5: `!edit aa2134 comment swap 1 2`
  - Ex6: `!edit 90199a limit "Yes" 10`

- `!delete <id|#channel|all>`

  This command can be used to delete an event or schedule. Passing the event ID as the first argument will delete the individual event, passing a channel mention to a schedule's channel will delete the schedule, and passing the word 'all' will result in the deletion of every scheduled event on the server. This command is not necessary to safely delete events and schedules. Guild administrators can delete the discord message or discord channel which represents the event or schedule to safely remove both.

  - Ex1: `!delete 10ada9d`
  - Ex2: `!delete #new_schedule`
  - Ex3: `!delete all`

- `!config <#channel> <option> <new_config>`

  When only the first argument (#channel) is provided, the currently configured settings for that schedule is return. Those settings can be modified by issuing the settings name as the second argument and the new configuration as the third argument. The list of `<option>` keywords are detailed later in this documentation.

  - Ex1: `!config #schedule msg "@here Event %a: %t"`
  - Ex2: `!config #schedule chan general`
  - Ex3: `!config #schedule clock 12`
  - Ex4: `!config #schedule remind "30, 10, 5 min"`
  - Ex5: `!config #schedule rsvp on`
  - Ex6: `!config #schedule rsvp remove "Undecided"`
  - Ex7: `!config #schedule rsvp add "Maybe" :thinking:`

- `!guild <option> <new_config>`

  This command may be used to modify guild-wide settings, such as your guild's command prefix and control channel. Issuing the guild command without arguments will output the current settings for your discord server. With this command you can specify which commands are 'restricted' and 'unrestricted'. A restricted command can only be used in the designated bot control channel. An unrestricted command may be used in any channel and by anyone so long as the bot has appropriate permissions to view and post in that channel. The list of `<option>` keywords are detailed later in this documentation.

- Ex1: `!guild`
- Ex2: `!guild prefix $`
- Ex3: `!guild restrict init`
- Ex4: `!guild unrestrict help`

- `!sync <#schedule_channel> [<import|export>] <calendar address>`

  This command will replace all events in the specified channel with events imported from a public google calendar. Up to the next 7 days of events will be imported. To learn more, see the **Synchronize a Schedule with Google Calendar** guide below. The schedule will re-import from the google calendar once per day automatically. While a schedule is synchronizing, the schedule can not be modified and new events may not be added. If you have authorized the bot access to your Google Account's calendars using the `oauth` command, the `export` keyword may be used to export events from a Discord schedule to a Google calendar.

  - Ex1: `!sync #schedule [. . .]`
  - Ex2: `!sync #schedule import [. . .]`
  - Ex3: `!sync #schedule export [. . .]`

- `!oauth [<token>]`

  This command can be used to link a Google Accounts authorization token with your Discord User ID. Linking a token will allow the bot to view and modify public *and* private calendars. Using the command without any arguments will provide a url link to generate a new Google Accounts access token. After an access token has been created, the token may be provided as an argument to the command to link your account. To remove a previously linked access token, use the **off** keyword instead.

  - Ex1: `!oauth`
  - Ex2: `!oauth ZLjaoaafj_agFff`
  - Ex3: `!oauth off`

- `!zones <filter>`

  This command will output all timezones which match the provided filter. The filter is a character or string, any timezone whose name contains the the filter will be matched. The filter is required.

  - Ex: `!zones us`

- `!test <event_id>`

  The test command will send an test announcement for the event to the bot control channel. The announcement message format is controlled by the schedule to which the event belongs to, and can be changed using the config command. The `remind` and `end` keywords may be used to specify which announcement type's format to test.

  - Ex1: `!test 02ac9b`
  - Ex2: `!test 02ac9b remind`
  - Ex3: `!test 02ac9b end`

- `!list <event_id> [<mode> <filter(s)>]`

  The list command will show all users who have rsvp'ed for an event. The command takes a single argument which should be the ID of the event you wish query. The schedule holding the event must have 'rsvp' turned on in the configuration settings. RSVP can be enabled on a channel using the config command as followed, `!config #channel rsvp on`. Mobile discord clients may have trouble rendering the embedded output of this command. If so, the command 'mode' may be set to mobile by appending `mobile` to command arguments. For a view more conducive to copy and pasting the list of RSVPers else where, try the `id` mode. The output of this command can be filtered by role, user, and type. Reference examples 2 and 3 below to see how this can be done.

  - Ex1: `!list 39a0g3`
  - Ex1: `!list 390da3 mobile`
  - Ex2: `!list 109d2a "u: @notem`
  - Ex3: `!list 019a92 "t: yes" "r: @admin"`

- `!sort <#schedule_channel>`

  The sort command will re-sort the entries in a schedule. Entries are reordered so that the top event entry is the next event to begin. The schedule cannot be modified while in the process of sorting. Schedules with more than 10 entries will not be sorted. The order in which the schedule is sorted can be specified by appending 'asc' (ascending order) or 'desc' (descending order) to the command.

  - Ex1: `!sort #schedule`
  - Ex2: `!sort #my_schedule asc`
  - Ex3: `!sort #my_schedule desc`

- `!help [command]`

  Direct messages the user a list of commands that can be used in Saber's dedicated control channel. If a command is supplied as the optional first argument, additional usage information for that command is provided. The information provided by the `help` command is likely to be more up-to-date than the documentation here.

  - Ex1: `!help`
  - Ex2: `!help edit`

- `!skip <id>`

  This command cause an event to 'skip' over the next scheduled occurrence of the event. If the event is scheduled to repeat, the event will be rescheduled to the time of next recurrence. If the event is not scheduled to repeat, the event is will deleted.

  - Ex: `!skip a09fji`

# Configuring RSVP on a Schedule

Schedules may be configured to allow users to RSVP to events on that schedule. Users can RSVP to an event by adding one of the appropriate reactions to the event's display message (located on the schedule channel). The RSVP groups and the emoticon associated with that RSVP group can be customized.

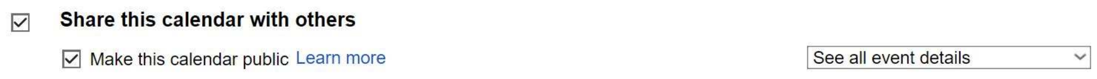
1. Enable RSVP on the schedule by using the `config` command.
   - Ex. `!config #schedule rsvp on`
   - This should cause three reactions to be added to every event on that schedule. All events' displays should update to show the current RSVP count for the three default RSVP groups: 'Yes', 'No', and 'Undecided'
2. To replace the RSVP groups with groups of your own, the default groups must be removed. Again the `config` must be used to remove the RSVP groups.
   - Ex. `!config #schedule rsvp remove "Yes"`
3. You can now add custom RSVP groups for your schedule using the `config` command.

- Ex. `!config #schedule rsvp add "DPS" :crossed_swords:`
- When adding RSVP groups, the rsvp 'add' options is used. The first argument after `add` needs to be the name of the new RSVP group. Spaces are allowed, but not recommend as the event display looks a bit distorted when spaces are used in the name. The second argument should then be the emoji to use as the reaction for that RSVP group. Both native discord emoticons as well as custom emoticons are acceptable.

4. For some events it may be desirable to set a limit to the number of individuals who may RSVP for particular group. This can be done using the `edit` command.
   - Ex. `!edit 0a8119d limit "DPS" 4`
   - The above command uses the 'limit' option for the `edit` command. The first argument after the `limit` needs to be the name of the group to add a player limit. The second argument should then be the limit. The limit MUST be a number/digit and cannot be represented as word (eg. `two`). There is one exception to this rule, to remove a previously set limit use `"off"` instead.

# Synchronize with Google Calendar

1. Create a public calendar with Google Calendar. To set the calendar public, you should find a sharing setting near that looks something like this:

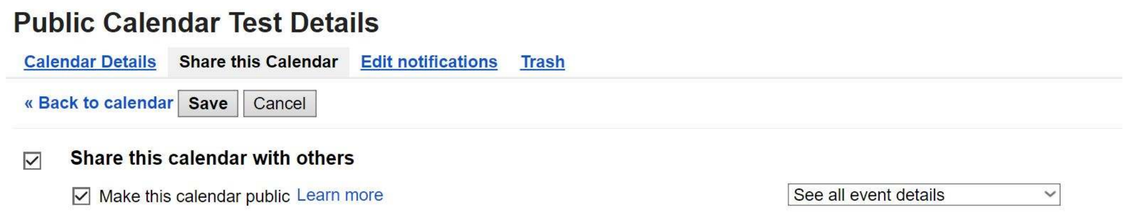


☑ Share this calendar with others

☑ Make this calendar public Learn more — See all event details

Or, if you wish to keep your calendar private, use the `oauth` command to provide access to your Google Account's calendars. For more information, reference the `oauth` command documentation.

2. (optional) If you already have a calendar setup, but not yet made it public, you can toggle the calendar's share settings in the 'Calendar settings'>>'Share this calendar tab' It should look something like this:



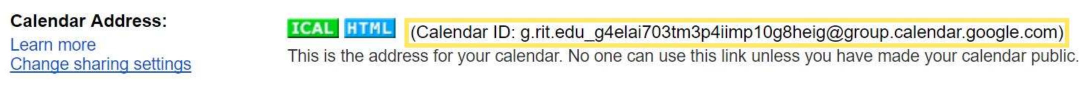
Public Calendar Test Details

☑ Share this calendar with others

☑ Make this calendar public Learn more — See all event details

Again, you may alternatively provide access to a private calendar with the `oauth` command.

3. Next, you'll need to find the calendar's public address. The calendar's public address is listed near the end in the calendar's settings webpage under the 'Calendar details' tab. It should look something like this:



Calendar Address:
Learn more
Change sharing settings

(Calendar ID: g.rit.edu_g4elai703tm3p4iimp10g8heig@group.calendar.google.com)
This is the address for your calendar. No one can use this link unless you have made your calendar public.

4. Finally, use the `!sync <#channel> <address>` command in your discord server's saber_control channel to sync the events in your public calendar to the schedule.

   The timer is automatically setup to resync the channel to your calendar once every day. However, if changes are made to the Google Calendar the channel will not show such changes until resync. Either wait until auto-sync, or reuse the `sync` command.

# Synchronizing Additional Parameters

Google Calendar events can be configured to sync some additional information to the Saber's discord schedules by including specific information in the description of the Google Calendar event. Special parameters are denoted by a special prefix string. When configure additional parameters, each parameter should be on it's own line in the Google Event's description.

Prefix	Suffix
image:	Url to an image
thumbnail:	Url to an thumbnail
url:	Valid url to use as event's title url
limit:	The name of a RSVP category and a limit to apply, separated by a space

Examples

- `thumbnail: https://nmathe.ws/thumbnail.png`

- `image: https://nmathe.ws/image.jpg`
- `url: https://nmathe.ws`
- `limit: Yes 8`

# Create and Edit Options

An edit command should be formatted like `!edit <ID> <keyword> <argument(s)>`

A create command differs in that several keywords and arguments may be used. So, a create command should look something like

```
!create <#channel> <title> <start> [end] [keyword] [argument(s)] ... [keyword] [argument]
```

The table below details what can be used in the [keyword] and [argument(s)] sections of the command

Shared Keywords

Keyword	Values
repeat	A comma separated list of days of week to repeat
interval	A (reasonably sized) number
date	A date in the format of yyyy/mm/dd
start-date	A date in the format of yyyy/mm/dd
end-date	A date in the format of yyyy/mm/dd
url	The url address for the event message's title
expire	A date in the format of yyyy/mm/dd
image	Image url to attach to the event's schedule display
thumbnail	Thumbnail url to attach to the event's schedule display

Edit Command Exclusive

Keyword	Values
title	The name given to the event
start	The time of day the event begins, formatted as hh:mm
end	The time of day the event ends, formatted as hh:mm

Keyword	Values
comment	First argument should be "add", "remove", or "swap". Later arguments are dependent upon the option used. Use the ``help`` command for specifics.
limit	First argument should be the RSVP group's name. Second argument should be the limit.
quiet-start	No arguments. This silences the start announcement.
quiet-end	No arguments. This silences the end announcement.
quiet-remind	No arguments. This silences any reminders.
quiet-all	No arguments. This will enable/disable ALL quiet options.

# Config Command Options

Announcement Settings

Keyword	Arguments
msg	A message format phrase
chan	The target channel for event announcements
end-msg	A message format phrase. This overrides `msg` for end announcements.
end-chan	The target channel for event announcements. This overrides `chan` for end announcements.

Reminder Settings

Keyword	Arguments
reminders	List of comma separated numbers
end-remind	List of comma separated numbers
remind-msg	A message format phrase. This overrides `msg` for reminders.

Keyword	Arguments
remind-chan	The target channel for event reminders. This overrides `chan` for reminders.

Misc Settings

Keyword	Arguments
clock	For 12 hour format use "12", for 24 hour format use "24"
zone	A zone from !zones
style	Use "full" for the full event display, "narrow" for a shortened display format.
sort	Use "asc" to auto-sort the entries in ascending order, "desc" for descending order, or "off" to disable auto-sort

Sync Settings

Keyword	Arguments
sync	A valid address, anything else will result in "off".
time	The time of day to schedule Google Calendar sync.
length	The number of days to sync from the Google Calendar

RSVP Settings

Keyword	Arguments
rsvp	Use "on" to enable rsvp, "off" to disable rsvp
rsvp add	First argument is the group name, second argument should be an emoji.
rsvp remove	The name of the group to remove.

# Guild Command Options

Keyword	Arguments
prefix	The prefix used to trigger bot commands.
control	The designated channel to listen for restricted commands.
restrict	The name of the command to restrict. Do not include the command prefix.

Keyword	Arguments
unrestrict	The name of the command to set as unrestricted.

# Custom Announcement Messages

Every schedule channel is configured with it's own message to announce to a specified channel when an event begins or ends. With the `config` command, a custom message can be configured.

Saber will announce the message string verbatum unless a **%** character is encountered. When a **%** character is encountered, Saber will read the next character(s) and substitute the token with whatever value is associated that character combination.

Advanced substition strings may also be used to further customize the announcement message. An advanced substitution string follows the form **${..}**. Advanced subsitution strings are a work in progress and are subject to change.

If you are having trouble understanding how this behaves, experiment with different message formats using the `config` and `test` commands.

Basic Substitutions

%-Token	Text Substitution
%t	Title of event announced
%c[n]	The [n]th comment of the event
%f	All comments on the event. Each comment appears on separate line.
%a	"begins [in x minutes]" or "ends [in x minutes]"
%b	"begins" or "ends"
%x	"in [x] minutes"
%i	The ID of the event
%s	The start time of the event
%e	The end time of the event
%d	The day of the month (number) the event starts
%D	The day of the week (word) the event starts
%m	The month (number) the event starts

%-Token	Text Substitution
%M	The month (word) the event starts
%y	The year the event starts
%u	The URL used by the title of the event
%v	The image URL of the event
%w	The thumbnail URL of the event
%n	A new line

Usage Examples

- `@here Event: %t`
- `@everyone %t %c1`
- `@everyone %f`
- `@here %t %a`
- `@here %t %b`
- `@here %x %t %b`
- `@here **%t** [%i] %a`
- `@here %t %b on the %dth`
- `@here %t %b on %D`
- `@here %t %b on %m/%d`
- `@here %t %b on %dth of %M`
- `@here %t %b on %y-%m-%d`
- `@here %t %b : %u`
- `@here %t %b : %v`
- `@here %t %b : %w`

Advanced Substitutions

%-String	Text Substitution
%{rsvp *Name*}	Inserts the RSVP user count for the *Name* category
%{[..]s}	If the event is starting, substitute in the text inside the brackets
%{[..]e}	If the event is ending, substitute in the text inside the brackets
%{[..]c*n*[..]}	If the nth comment exists, substitute in the text inside the brackets
%{[..]m[..]}	If the announcement is a reminder, substitute in the text inside the brackets
%{[..]u[..]}	If the event has a title URL, substitute in the text inside the brackets
%{[..]v[..]}	If the event has an image, substitute in the text inside the brackets

%-String	Text Substitution
%{[..]w[..]}	If the event has an thumbnail, substitute in the text inside the brackets

Usage Examples

- `@here %t %a: %{rsvp Yes} %{rsvp No}`
- `@here %t %{[has started!]s}`
- `@here %{[In ]m[ minutes]} %t ${[closes]e}`
- `@everyone %t %{c1}`
- `@here %t %{[in ]h[ hours!]}`
- `@here %t %b %{[Please visit ]u}`
- `@here %t %b %{[Image: ]v}`
- `@here %t %b %{[Thumbnail: ]w}`

# Self-Hosting Saber Bot

1. Saber-bot has been developed using the Java 8 programming language. To run this application, you will need to download and install the Java 8 runtime environment for your system.

2. Saber-bot uses MongoDB to store the persistent event and schedule data. Install the latest version of MongoDB on your system. I suggest you setup MongoDB to run on system startup as well.

3. With the prequisite software installed, you should now be able to launch the bot application. Download the latest version of the bot from here, or compile from the bot from source using Maven.

4. Running the .jar file for the first time should cause the application to generate a fresh configuration file and close. Add your discord bot token and your discord user ID to the configuration file and restart the bot.

5. To have Google Calendar integration, you need to generate new credentials on the Google Developer Console. Generate a new 'service account key' and download the key as a JSON file type. Modify the configuration file's `google_service_key` settings to indicate the location of your service key file.

6. Some notes on the `saber.toml` configuration file:
   - If you have configured you're MongoDB application to a non-default state the `mongodb` setting may need to be adjusted.

- The `web_token` setting is used to share bot metrics with bots.discord.pw, and is completely optional.
- Sharding is typically unnecessary for self-hosted applications. If sharding is disabled by default with the `shard_total` set to 0. However, if you decide to enable sharding the `shard_total` configures the total number of shards used by all instances of the bot and most always be equal to or greater than the number of shards in the `shards` list. The `shards` list should contain the shard IDs for each shard the local application should manage.
- The `rsvp_yes`, `rsvp_no`, and `rsvp_clear` are unicode emoticons. If your server does not have the required packages installed, the emoticons may not be processed correctly. If so, use the escaped unicode IDs (`\u2705`, `\u274c` and `\u2754`) instead.

Documentation last updated: 2017-10-10