

---

# **telegram-reminder-bot**

***Release 0.1.0***

**Yoann Piétri**

**Jan 03, 2022**



**CONTENTS:**

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Getting the software . . . . .	3
1.2	Database . . . . .	5
1.3	Configuration . . . . .	5
1.4	Init database . . . . .	6
1.5	Starting and stop the bot . . . . .	7
1.6	Full example . . . . .	7
<b>2</b>	<b>Usage</b>	<b>11</b>
2.1	Start command . . . . .	11
2.2	Help command . . . . .	11
2.3	Add job and normal behavior . . . . .	12
2.4	Show jobs . . . . .	13
2.5	Pause job . . . . .	13
2.6	Resume job . . . . .	14
2.7	Delete a job . . . . .	15
<b>3</b>	<b>Bot</b>	<b>17</b>
<b>4</b>	<b>Models</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



1



## INSTALLATION

### 1.1 Getting the software

There are several ways of getting the `telegram-reminder-bot` software, the simpler being using `pip`.

#### 1.1.1 With `pip`

You can install the package using the command

```
pip install telegram-reminder-bot
```

#### 1.1.2 With a `git clone`

You clone the repository from github. This will allow easier updates when new code is released :

```
git clone https://github.com/nanoy42/telegram-reminder-bot
```

You can then install the dependencies using `poetry` or `pip` :

##### Using `poetry`

```
poetry install
```

or if you only want the dependencies and not the `telegram-reminder-bot` package to be installed :

```
poetry install --no-root
```

---

**Note:** If you want to use the daemonized version, you will need extra dependencies : `poetry install -E daemons` or `poetry install -E daemons --no-root`

---

You can install dev dependencies using

```
poetry install --dev
```

## Using pip

You can install the dependencies using the following command :

```
pip install -r requirements.txt
```

You can also install the dev dependencies using :

```
pip install -r dev-requirements.txt
```

## 1.1.3 From source

You can download the latest release here : <https://github.com/nanoy42/telegram-reminder-bot/releases>

You can then install the dependencies using poetry or pip :

## Using poetry

```
poetry install
```

or if you only want the dependencies and not the telegram-reminder-bot package to be installed :

```
poetry install --no-root
```

---

**Note:** If you want to use the daemonized version, you will need extra dependencies : `poetry install -E daemons` or `poetry install -E daemons --no-root`

---

You can install dev dependencies using

```
poetry install --dev
```

## Using pip

You can install the dependencies using the following command :

```
pip install -r requirements.txt
```

You can also install the dev dependencies using :

```
pip install -r dev-requirements.txt
```



## 1.2 Database

You will need a working database to use the bot. Any database that is compatible with `sqlalchemy` (see <https://docs.sqlalchemy.org/en/14/dialects/index.html> for the list of supported databases).

The bot was explicitly tested with `sqlite`, it would probably work with other databases but it was not tested.

## 1.3 Configuration

The configuration file is the file holding the configuration for the telegram bot and the database.

You can create the default configuration file by calling:

```
telegram-reminder-bot init_config -c configuration_path
```

The default configuration file will be created at the configuration path.

The default configuration file looks like

```
; telegram-reminder-bot Copyright (c) 2021-2022 Yoann Piétri
;
; This software is released under the MIT License.
; https://opensource.org/licenses/MIT
;
; This is the configuration file for telegram-reminder-bot
; The configuration file is separated in two sections : telegram and db
; The first holds the token and the list of allowed users
; The second holds the engine configuration

[telegram]
; Token of the telegram Bot in the format 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11
; Default is 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11
token = 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11

; List of allowed users, separated by commas. If 0 is included
; all users are considered to be allowed.
; An empty string means no user is allowed
; Default is an empty string
allowed_users =

[db]
; Configuration of the engine. See https://docs.sqlalchemy.org/en/14/core/engines.html
; WARNING : relative path are not supported yet for daemonized mode.
; Default is sqlite:///var/telegram-reminder-bot/db.db
engine = sqlite:///var/telegram-reminder-bot/db.db

[logs]
; Path for the logs
; If no path is given, no logs are written
; Default is /var/log/telegram-reminder-bot.log
path = /var/log/telegram-reminder-bot.log
```

### 1.3.1 telegram

#### token

The token of the telegram bot, in the format 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11 (without bot at the beginning). Default is 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11.

#### allowed\_users

The list of allowed users and chats to use the bot. This as to be the list of IDs of users and chats separated by commas. If the list is empty, no user or chat is allowed to user the bot. If the list contains a `0`, then all users and chats are allowed to use the bot. Default is an empty string.

---

**Note:** This is maybe the moment to speak of the behavior of this bot inside a group. Please keep in mind that this was **not** tested and this is only a prediction of what will happen.

The reminder will be associated to the chat id, and not a user id, if of course the chat id is in the list of allowed users in the configuration. Hence, everyone in the chat will be able to make modifications on the jobs.

I would not recommend the usage of this bot, as it is, in a chat. Hence, in the documentation I will refer as a user using the bot.

---

### 1.3.2 db

#### engine

The database configuration. See <https://docs.sqlalchemy.org/en/14/dialects/index.html> for documentation. Default is `sqlite:///var/telegram-reminder-bot/db.db`.

**Warning:** Relative paths are not supported yet when using the daemonized mode. Please use absolute paths.

### 1.3.3 logs

#### path

Path for the log file. If none is given, no logs are written. Default to `/var/log/telegram-reminder-bot.log`

## 1.4 Init database

Before using the bot, the database must be initialized.

There is a command line instruction in the `telegram-reminder-bot` script to initialize the database (`init_db`) :

```
telegram-reminder-bot init_db -c config.ini
```

**Warning:** This code is released as alpha version and therefore the structure of the databse is likely to evolve in the future and no migration infrastructure was planned in this version. While we will try to provide ways of migrating, we cannot ensure for now that the migration of the data will be simple.

## 1.5 Starting and stop the bot

The bot may be started using the `entrypoint.py` script. If the package was installed using `pip` the command `telegram-reminder-bot` is equivalent.

This command may be used as

```
usage: telegram-reminder-bot [-h] [-c CONFIGURATION_FILE] {start,stop,restart,debug,init_
↳db}

Telegram reminder bot v0.1.0

positional arguments:
{start,stop,restart,debug,init_db,init_config}

options:
-h, --help            show this help message and exit
-c CONFIGURATION_FILE, --configuration-file CONFIGURATION_FILE
                        Path of the configuration file. Default to /etc/telegram-
↳reminder-bot/config.ini
```

The possible actions are

- `start` : start the bot, in daemon mode (requires the `daemons` package).
- `stop` : stop the bot, in daemon mode (requires the `daemons` package).
- `restart` : restart the bot, in daemon mode (requires the `daemons` package).
- `debug` : start the bot directly with debug level for logs
- `init_db` : initialize the database.
- `init_config` : copy the default configuration file to the location given by `-c` or `--configuration-file` (`/etc/telegram-reminder-bot/config.ini` by default).

The path of the configuration file can be given using the `-c` of `--configuration-file` option. The default value for the option is `/etc/telegram-reminder-bot/config.ini`.

The documentation of the command can be accessed using `--help`.

## 1.6 Full example

This a full example of a quickstart to run the bot :

```
$ pip install telegram-reminder-bot
[...]
$ telegram-reminder-bot -h
usage: telegram-reminder-bot [-h] [-c CONFIGURATION_FILE] {start,stop,restart,debug,init_
↳db,init_config}

Telegram reminder bot v0.1.0

positional arguments:
{start,stop,restart,debug,init_db,init_config}
```

(continues on next page)

(continued from previous page)

```

options:
-h, --help            show this help message and exit
-c CONFIGURATION_FILE, --configuration-file CONFIGURATION_FILE
                        Path of the configuration file. Default to /etc/telegram-
↳ reminder-bot/config.ini

$ telegram-reminder-bot init_config -c config.ini
[OK] The default configuration file was copied to config.ini.

$ ls
config.ini

$ cat config.ini
; telegram-reminder-bot Copyright (c) 2021-2022 Yoann Piétri
;
; This software is released under the MIT License.
; https://opensource.org/licenses/MIT
;
; This is the configuration file for telegram-reminder-bot
; The configuration file is separated in two sections : telegram and db
; The first holds the token and the list of allowed users
; The second holds the engine configuration

[telegram]
; Token of the telegram Bot in the format 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11
; Default is 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11
token = 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11

; List of allowed users, separated by commas. If 0 is included
; all users are considered to be allowed.
; An empty string means no user is allowed
; Default is an empty string
allowed_users =

[db]
; Configuration of the engine. See https://docs.sqlalchemy.org/en/14/core/engines.html
; WARNING : relative path are not supported yet for daemonized mode.
; Default is sqlite:///var/telegram-reminder-bot/db.db
engine = sqlite:///var/telegram-reminder-bot/db.db

[logs]
; Path for the logs
; If no path is given, no logs are written
; Default is /var/log/telegram-reminder-bot.log
path = /var/log/telegram-reminder-bot.log

$ vim config.ini
[...]

$ cat config.ini
; telegram-reminder-bot Copyright (c) 2021-2022 Yoann Piétri
;

```

(continues on next page)

(continued from previous page)

```

; This software is released under the MIT License.
; https://opensource.org/licenses/MIT
;
; This is the configuration file for telegram-reminder-bot
; The configuration file is separated in two sections : telegram and db
; The first holds the token and the list of allowed users
; The second holds the engine configuration

[telegram]
; Token of the telegram Bot in the format 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11
; Default is 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11
token = *****

; List of allowed users, separated by commas. If @ is included
; all users are considered to be allowed.
; An empty string means no user is allowed
; Default is an empty string
allowed_users = *****

[db]
; Configuration of the engine. See https://docs.sqlalchemy.org/en/14/core/engines.html
; WARNING : relative path are not supported yet for daemonized mode.
; Default is sqlite:///var/telegram-reminder-bot/db.db
engine = sqlite:///home/nanoy/Projets/test-telegram-reminder-bot/db.db

[logs]
; Path for the logs
; If no path is given, no logs are written
; Default is /var/log/telegram-reminder-bot.log
path = telegram-reminder-bot.log

$ telegram-reminder-bot init_db -c config.ini
[OK] The database was successfully initialized

$ ls
config.ini  db.db

$ telegram-reminder-bot start -c config.ini

```



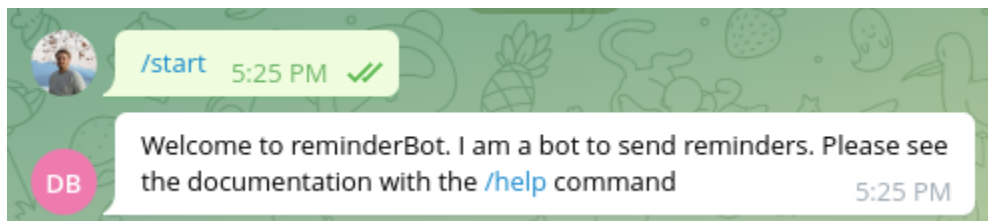
## USAGE

This page will show examples of interaction with a telegram\_reminder\_bot.

This assumes that the bot has been set up and the user is in the allowed users in the configuration,

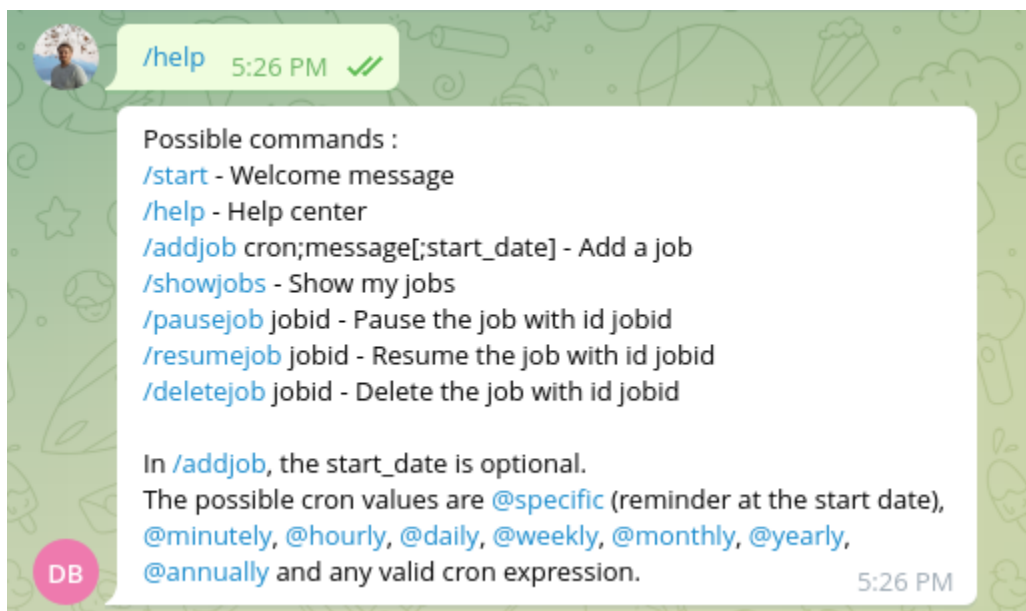
### 2.1 Start command

The `/start` command does not take any argument. The displayed message depends on if the user is authorized or not.



### 2.2 Help command

The `/help` command does not take any argument. The displayed message depends on if the user is authorized or not.

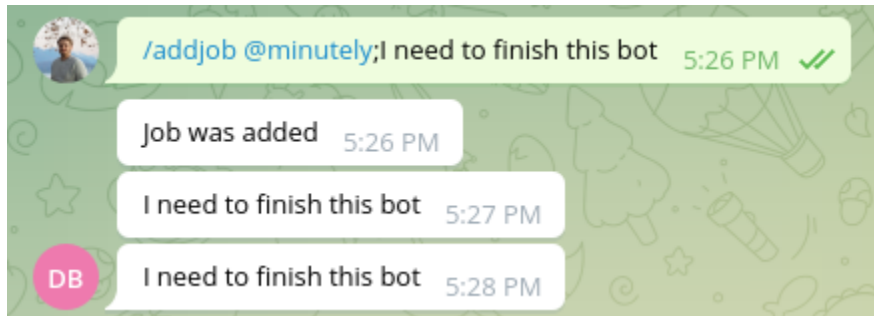


## 2.3 Add job and normal behavior

The `/addjob` commands take up to three arguments :

- the cron expression or `@specific`.
- the message
- an optional start date

in the format : `/addjob cron;message[;start_date]`.



### 2.3.1 The cron expression

This can be any valid cron expression (like `*/5 * * * *`) or shortcuts :

Shortcut	Cron value
@minutely	* * * * *
@hourly	0 * * * *
@midnight	0 0 * * *
@daily	0 0 * * *
@weekly	0 0 * * 0
@monthly	0 0 1 * *
@yearly	0 0 1 1 *
@annually	0 0 1 1 *

The `@specific` keyword does not represent a cron expression, but a single reminder that will happen at the start date.

---

**Note:** When using `@specific`, be sure to set the start date, otherwise, you will have a specific remind in the minute following the creation and that's it.

---

---

**Note:** A job with `@specific` will be deleted after the message is sent, as it is the only message to send.

---



### 2.3.2 The message

The message can be composed of any character that is not a semi-colon (;).

### 2.3.3 The start date

The start date is optional.

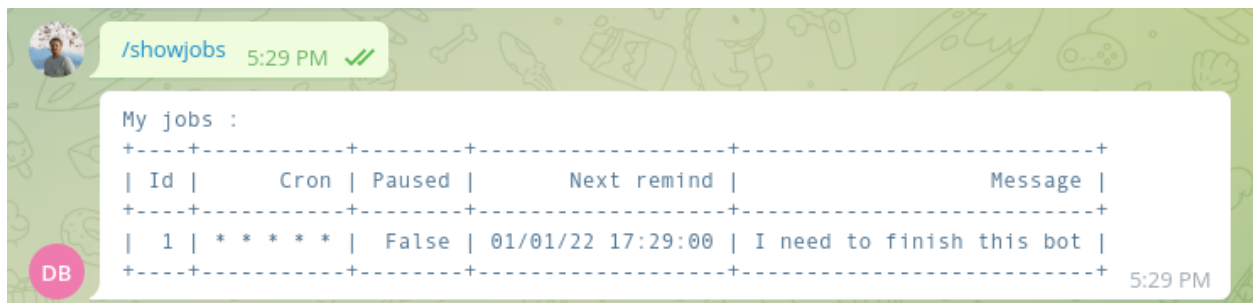
The start date must be given in the format %d/%m/%y %H:%M:%S.

If no start date is given, then the default is now.

**Warning:** If you don't give a start date, don't pull the final semi-colon (;). For instance, /addjob @minutely; message; would fail to parse.

## 2.4 Show jobs

The /showjobs command will display an array with all the jobs associated to the user making the request.



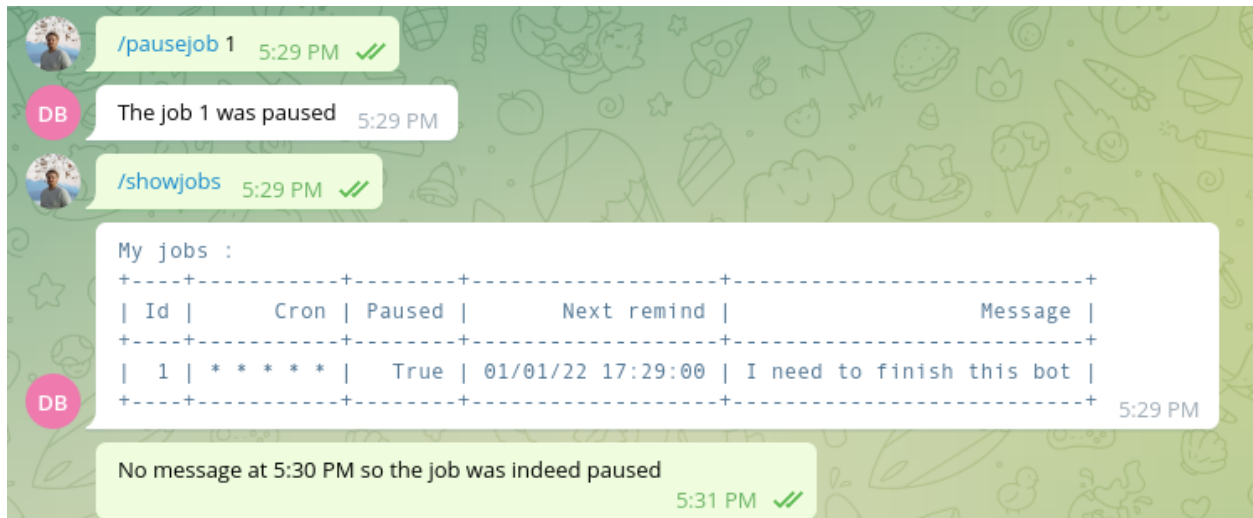
Please note that will the value of “next remind” will indicate that the reminder should happen at 00 seconds, there is no reliable way to tell when the reminder will actually happen (it will happen between :00 and :59 seconds).

## 2.5 Pause job

The /pausejob command takes jobid as parameter : /pausejob jobid. The command will only work if

- jobid is an int,
- that is the id of a Reminder,
- and the user associated to the reminder is the user sending the message.

No message will be sent for this job until it is resumed.

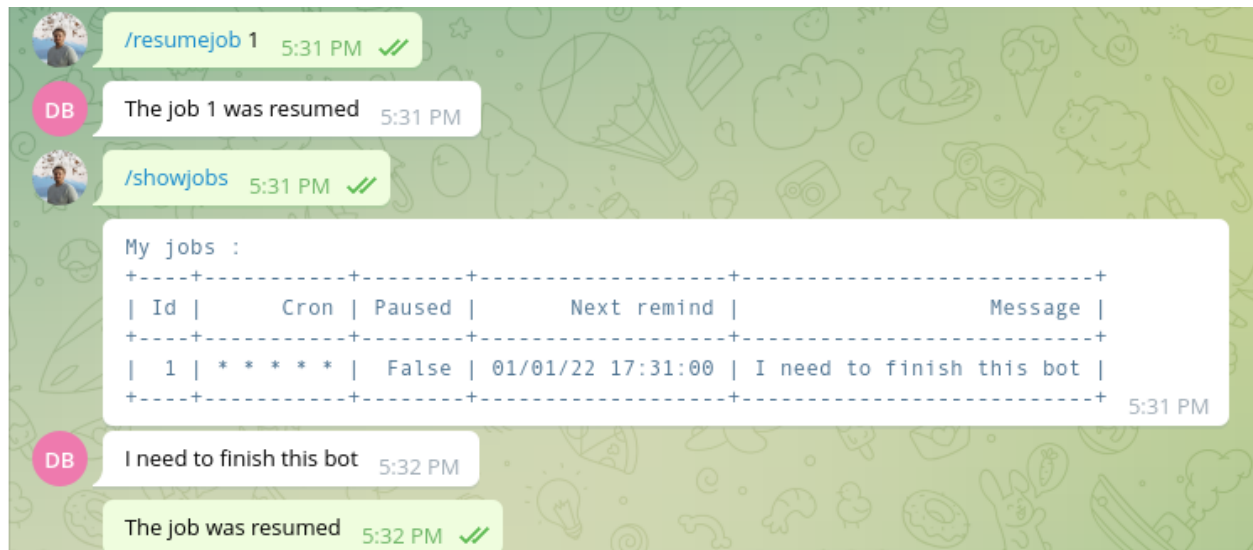


**Note:** The command will work on a reminder that is paused, but will have no effect.

## 2.6 Resume job

The `/resumejob` command takes `jobid` as parameter : `/resumejob jobid`. The command will only work if

- `jobid` is an int,
- that is the id of a Reminder,
- and the user associated to the reminder is the user sending the message.

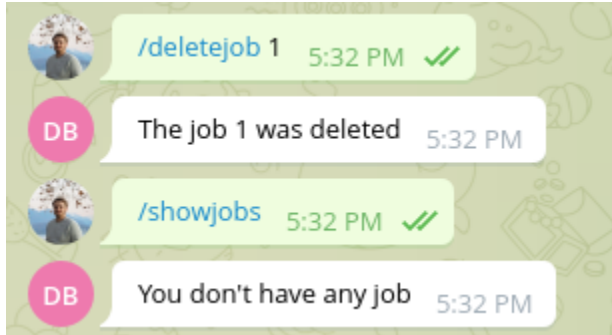


**Note:** The command will work on a reminder that is not paused, but will have no effect.

## 2.7 Delete a job

The `/deletejob` command takes `jobid` as parameter : `/deletejob jobid`. The command will only work if

- `jobid` is an int,
- that is the id of a Reminder,
- and the user associated to the reminder is the user sending the message.



**Warning:** There is no way of reverting a `deletejob` command.



---

CHAPTER  
**THREE**

---

**BOT**



**MODELS**





## INDEX

### A

`allowed_users`, 6

### E

`engine`, 6

### P

`path`, 6

### T

`token`, 6