



[R] TG手動交易狗-建置說明

R telegram bot 建起來相當簡單，造著cran-r附的說明，很快就做出一個多功能的Bot了，本篇同時介紹製作Bot功能和建置步驟，在後面會簡單介紹Bot code內容，就是讓讀者幫我抓個bug。不過裡面接API的東西有點麻煩，如果R的版本沒辦法下載tidyquant，可能就無法繪製K線，不過就少一個功能這樣，也還好。

使用工具與服務

大致上把R的開發環境安裝OK就好了

1. [R 4.0.3](#) + [Rstudio 1.3.1093](#)
2. [Miniconda](#) ← 這個不用先載
3. [R套件下載工具 pacman](#)
4. [R Python串接介面 reticulate](#)
5. [R tidyverse](#)
6. [R telegram.bot](#)
7. [telegram](#)

第一步、創建新的Bot

telegram bot 很好創建，就和telegram官方製作的Bot father說說就有了，之後管理也是在這邊弄，反正就先創一隻telegram帳號，再加BotFather為好友，就可以輕鬆的創建自己的Bot了。



<https://t.me/BotFather>



BotFather 

@BotFather

BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

SEND MESSAGE

- 加BotFather為好友後，他就會跟你說你可以用甚麼指令和他互動。

```
I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual (https://core.telegram.org/)

You can control me by sending these commands:

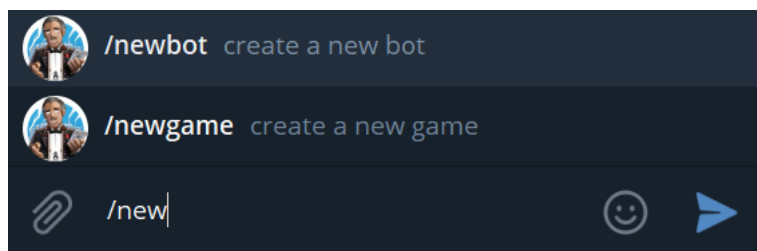
/newbot - create a new bot
/mybots - edit your bots [beta]

Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot

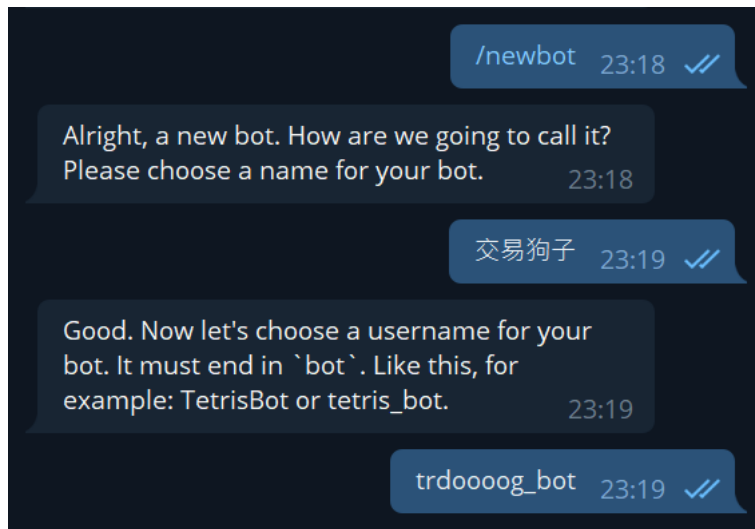
Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode (https://core.telegram.org/bots/inline)
/setinlinegeo - toggle inline location requests (https://core.telegram.org/bots/inline#location-based-results)
/setinlinefeedback - change inline feedback (https://core.telegram.org/bots/inline#collecting-feedback) settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode (https://core.telegram.org/bots#privacy-mode) in groups

Games
/mygames - edit your games (https://core.telegram.org/bots/games) [beta]
/newgame - create a new game (https://core.telegram.org/bots/games)
/listgames - get a list of your games
/editgame - edit a game
/deletgame - delete an existing game
```

- 輸入 `/newbot` 來創建一隻新的bot



- 接著輸入Bot名字和username(類似網址但要以bot結尾)接下來就會Bot拔拔就會吐API給你了



Done! Congratulations on your new bot. You will find it at t.me/trdooooog_bot. You can now add a description, about section and

Use this token to access the HTTP API:
這一行就是你的API
Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: <https://core.telegram.org/bots/api>

t.me/username 為你bot的位置，待會要先和他互動拿到client ID，他是用來加朋友的連結。

在 上面我標示那行 為API，可以access [telegram API](#)，請勿外洩。

第二步、R Bot getID

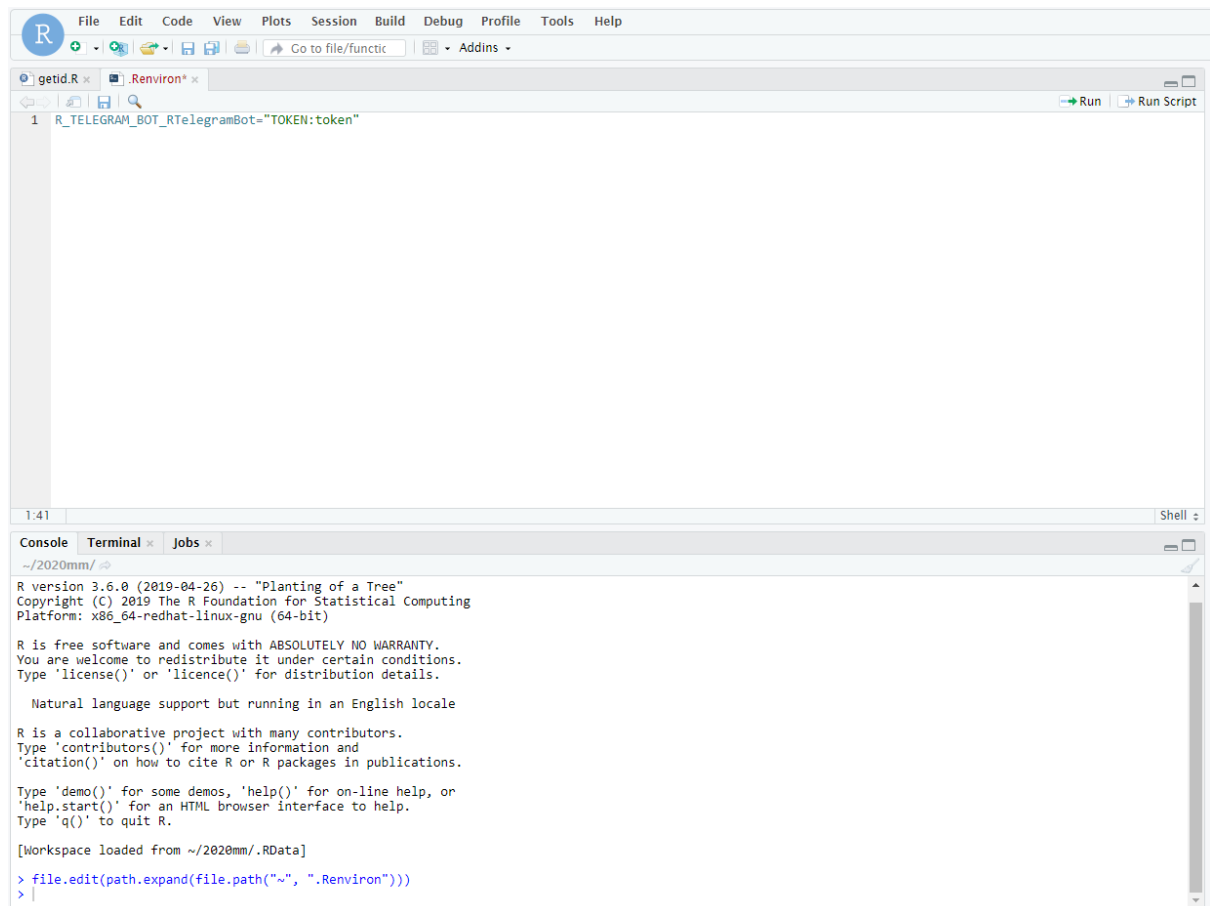
如果你用rstudio-server的話，把**tgbot.zip** upload到home目錄底下，如果用普通電腦，放自己喜歡的地方就好，不過待會要改工作位置。然後開啟 **getid.R** 然後執行，這步會下載需要的package和運用bot告訴你你的client id，不過也要先設定自己的API。

- 輸入

```
file.edit(path.expand(file.path("~", ".Renvirom")))
```

- 在 **.Renvirom** 裡面輸入

```
R_TELEGRAM_BOT_RTelegramBot="你的token"
```



```
1 R_TELEGRAM_BOT_RTelegramBot="TOKEN:token"
```

```
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (c) 2019 The R Foundation for Statistical Computing
Platform: x86_64-redhat-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

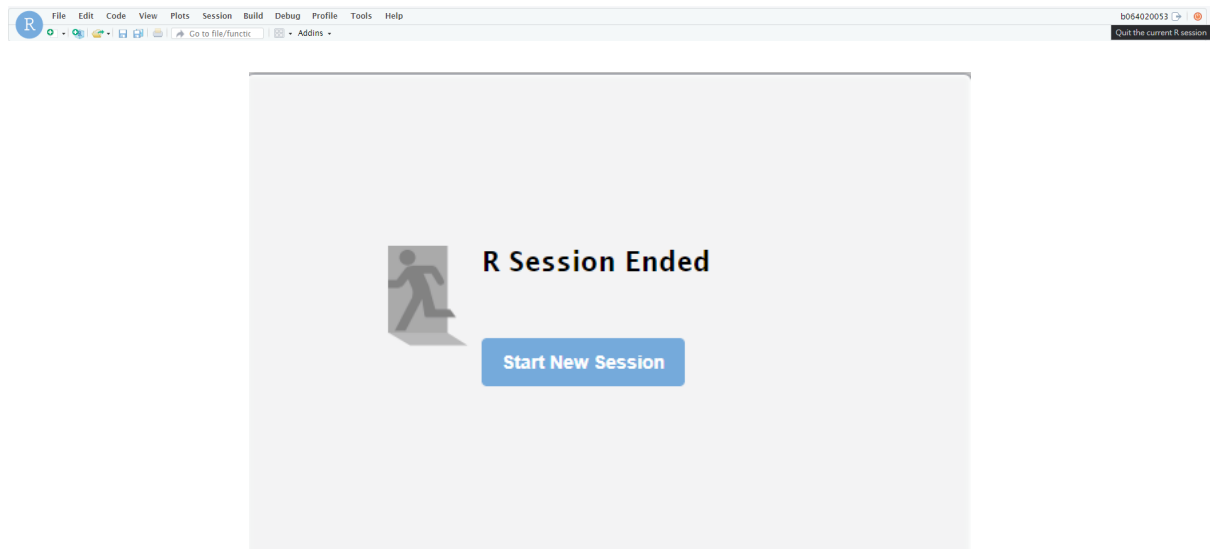
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/2020mm/.RData]
> file.edit(path.expand(file.path("~", ".Renviron")))
>
```



[必要]然後儲存後重新開啟Rstudio或Rsession



接著就可以執行 `getid.R`，第一次下載pkg可能要一陣子

```
setwd("~/")
rm(list=ls(all=T)) # 清空資料
if(!require(pacman))install.packages("pacman")
pacman::p_load(telegram.bot, reticulate, tidyverse, tidyquant) # 第一次可能跑一陣子

# file.edit(path.expand(file.path("~", ".Renviron")))
```

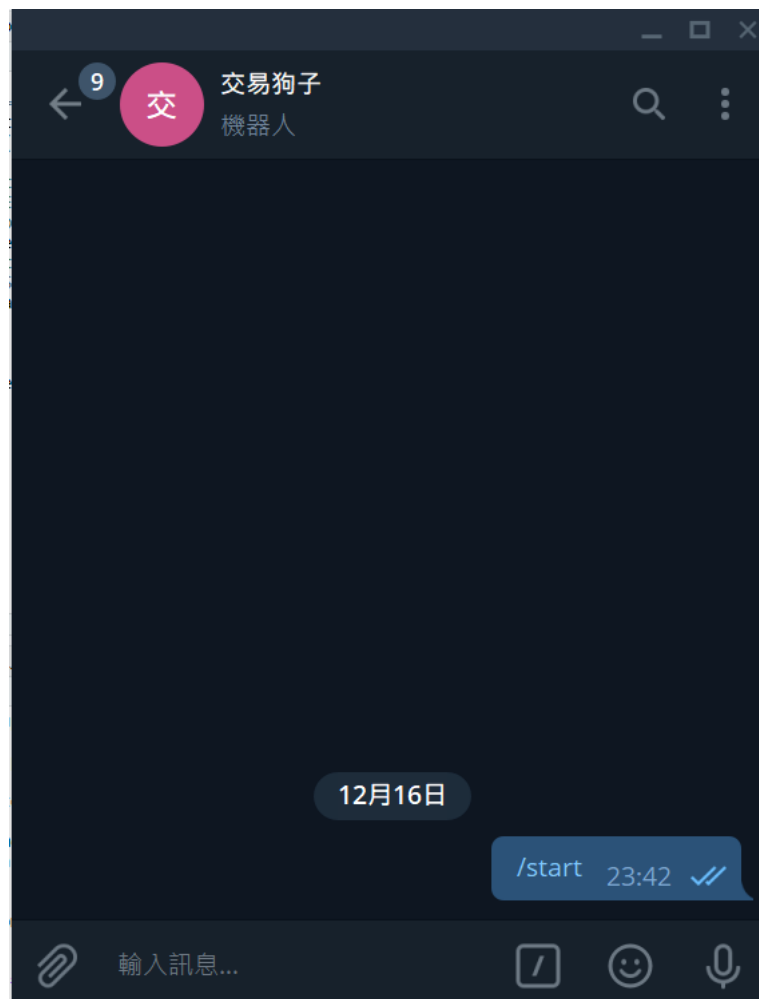
```
# example. R_TELEGRAM_BOT_RTelegramBot=TOKEN
# Initialize bot
bot <- Bot(token = bot_token("RTelegramBot"))
# print(bot$getMe())
updates <- bot$getUpdates()
chat_id <- updates[[1L]]$from_chat_id()
print(chat_id)

# Send message
bot$sendMessage(chat_id, text = paste("*A !*your id:", chat_id))
```



注意一下，`reticulate` 會問你要不要下載 `miniconda`，如果不下載的話要設定還要多打自己conda的location，我的建議是按y直接下載下去。

在執行完 `bot <- Bot(token = bot_token("RTelegramBot"))` 這行後，就可以加自己的bot好友然後和他說 `/start` 了，然後再執行剩下幾行，他就會把你的chatID直接傳給你，也同時會顯示再r Console裡。



```
> chat_id <- updates[[1L]]$from_chat_id()
> print(chat_id)
[1] [REDACTED]
> # Send message
> bot$sendMessage(chat_id, text = paste("*A !*your id:", chat_id))
```



如果遇到這行 `Error in updates[[1L]] : subscript out of bounds` 就再傳一次訊息再update一次，應該就可以解決了

第三步、開啟tgc.R

- 接著把你的 `MAXAPI` 和 `client ID` 填進去 `.Renviron`

```
file.edit(path.expand(file.path("~", ".Renviroon")))
```

- 在 `.Renviroon` 裡面輸入，再重啟一次 **Rstudio** 或 **rsession**



這邊學我用 `TG_WEAK`、`weak_MAX_API_KEY`、`weak_MAX_API_SECRET` 可以直接跳到第五步，把他run起來就OK了

```
TG_ME="clientid"
MAX_API_KEY="key"
MAX_API_SECRET="secret"
```

```
1 R_TELEGRAM_BOT_RTelegramBot="token:token"
2 TG_ME="clientid"
3 MAX_API_KEY="key"
4 MAX_API_SECRET="secret"
5
```

tg.R前25行code

```
setwd("~/")
rm(list=ls(all=T)) # 清空資料
if(!require(pacman)) install.packages("pacman")
pacman::p_load(tidyverse, tidyquant, reticulate, telegram.bot)
max <- import("max.client")
client <- max$client("", "")
wclient <- max$client(Sys.getenv("weak_MAX_API_KEY"), Sys.getenv("weak_MAX_API_SECRET"))
sclient <- max$client(Sys.getenv("shu_MAX_API_KEY"), Sys.getenv("shu_MAX_API_SECRET"))
lclient <- max$client(Sys.getenv("lin_MAX_API_KEY"), Sys.getenv("lin_MAX_API_SECRET"))
#
updater <- Updater(token = bot_token("RTelegramBot"))
# useful function
r2df <- function(l) {
  if(length(l)!=0){
    df <- data.frame(matrix(unlist(l), nrow=length(l), byrow=T),
                          stringsAsFactors=FALSE);
    name <- names(l[[1]]); colnames(df) <- name[1:ncol(df)];
    return(df)} }
chkclient <- function(cid) {
  if(cid==Sys.getenv("TG_WEAK")){ return (wclient)
  }else if(cid==Sys.getenv("TG_SHU")){ return (sclient)
  }else if(cid==Sys.getenv("TG_LIN")){ return (lclient)
  }else {return (client)}
}
```



主要要修改的地方在第7-9行和第20-25的chkclient函式

- 修改 `wclient` 那行改成

```
wclient <- max$client(Sys.getenv("MAX_API_KEY"), Sys.getenv("MAX_API_SECRET"))
```

- 如果有很多人的話，可以造著你在 `.Renviroon` 輸入的名稱修改 把21行改成你設定的，格式一樣是 `Sys.getenv("你設定的")`

```
# 該行為偵測誰是TG主人來傳切換API
if(cid==Sys.getenv("TG_ME")){ return (wclient)
```

第四步、filter掉其他使用者

這不就是讓其他使用者使用你的bot不會有任何的回應，不會讓其他怪怪的人操作你的Bot

- `tg.c.R` 353行後的code

```
# tg setting
filter_user <- as.BaseFilter(function(message)
  message$from_user %in% c(Sys.getenv("TG_WEAK"), Sys.getenv("TG_SHU"), Sys.getenv("TG_LIN")))
updater <- updater + CallbackQueryHandler(answer_cb)
updater <- updater +
  CommandHandler("buy", buy, pass_args = TRUE, filter_user) +
  CommandHandler("ubuy", ubuy, pass_args = TRUE, filter_user) +
  CommandHandler("sell", sell, pass_args = TRUE, filter_user) +
  CommandHandler("getk", getk, pass_args = TRUE, filter_user) +
  CommandHandler("cvol", cvol, pass_args = TRUE, filter_user) +
  CommandHandler("cpr", cpr, pass_args = TRUE, filter_user) +
  CommandHandler("can1", can1, pass_args = TRUE, filter_user) +
  CommandHandler("canid", canid, pass_args = TRUE, filter_user) +
  CommandHandler("profit", profit, pass_args = FALSE, filter_user) +
  CommandHandler("pricex", pricex, pass_args = FALSE, filter_user) +
  CommandHandler("this", this, pass_args = TRUE, filter_user) +
  CommandHandler("ohis", ohis, pass_args = FALSE, filter_user) +
  CommandHandler("help", help, pass_args = FALSE, filter_user) +
  CommandHandler("start", help, pass_args = FALSE, filter_user)
# Create error callback
error_callback <- function(bot, error) {
  warning(simpleWarning(conditionMessage(error), call = "Updates polling"))
}
# Register it to the updater's dispatcher
updater$dispatcher$add_error_handler(error_callback)
updater$start_polling()
bot <- Bot(token = bot_token("RTelegramBot"))
updates <- bot$getUpdates()
bot$sendMessage(Sys.getenv("TG_WEAK"), text = "**懶狗通知*壞掉了要重開")
```

- 主要就是355行filter user改掉，把 `TG_WEAK` 改成 `TG_ME`

```
message$from_user %in% c(Sys.getenv("TG_ME"), Sys.getenv("TG_SHU"), Sys.getenv("TG_LIN")))
```

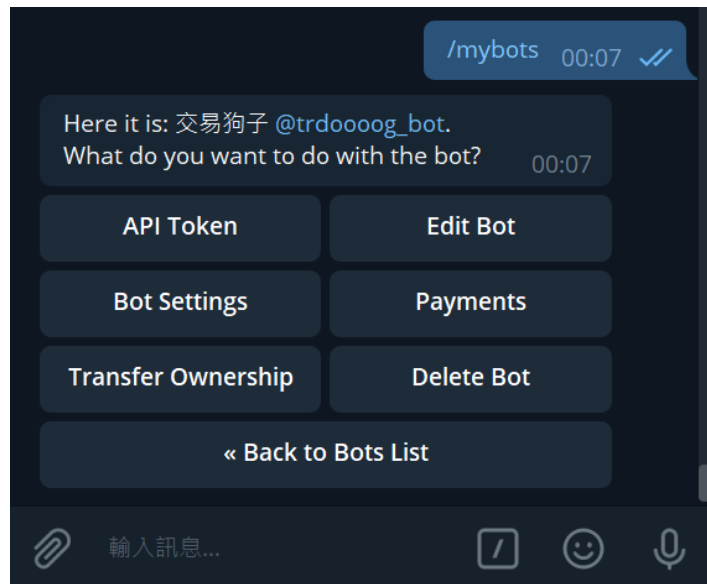
- 然後最後一行也需要修改，邏輯同上，這行非常重要，如果有突發狀況，他會告訴你掛掉了，你就要去重新開啟

```
bot$sendMessage(Sys.getenv("TG_ME"), text = "**懶狗通知*壞掉了要重開")
```

然後全選 run起來，你的bot就架好了 😊

第五步、TG設定快速command捷徑

跟@BotFather說 `/mybots` 點擊自己的bot，按**Edit Bot**再按**Edit Commands** 輸入



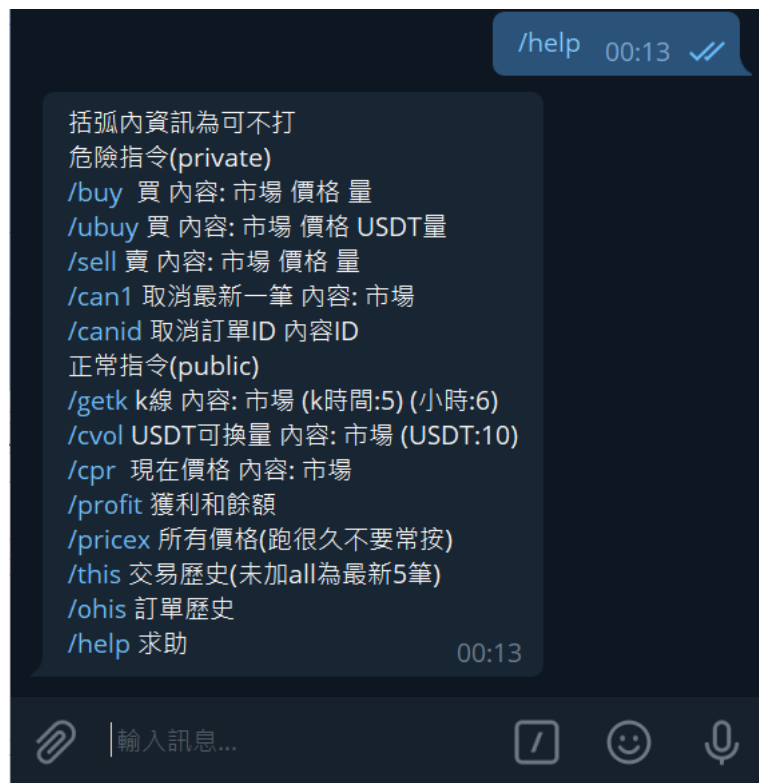
目前不包含help總共有12種功能，可買賣以及看價格

```
buy-買 內容: 市場 價格 量
ubuy-買 內容: 市場 價格 USDT量
sell-賣 內容: 市場 價格 量
getk-k線 內容: 市場 (k時間:5) (小時:6)
cvol-USDT可換量 內容: 市場 (USDT:10)
cpr-現在價格 內容: 市場
can1-取消最新一筆 內容: 市場
canid-取消訂單ID 內容ID
profit-獲利和餘額
pricex-所有價格(跑很久不要常按)
this-交易歷史(未加all為最新5筆)
ohis-訂單歷史
help-求助
```

這樣你輸入 `/` 他就會提醒你輸入甚麼



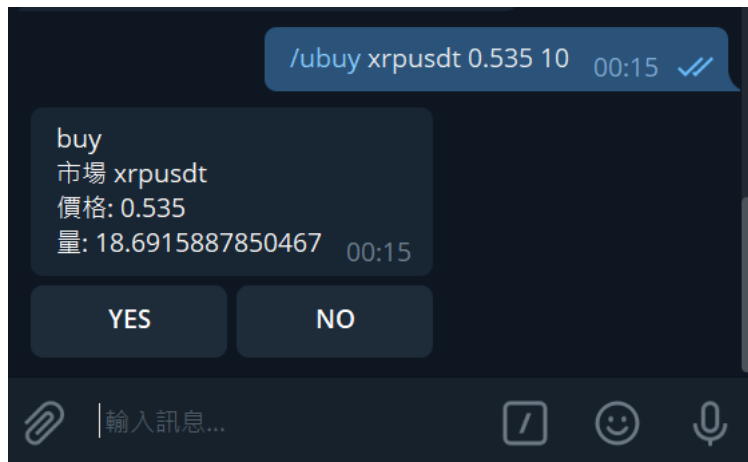
範例 `/help` 可獲得所有指令



範例 `/getk` 可輸入市場來繪製K線，可自行輸入更長指令，範例: `/getk xrpusdt 15 9`



範例 `/ubuy` 可輸入市場、價、USDT量來購買。



💡 注意Level 1帳號不能掛新台幣，裡面所有市場都是xxxusdt

🌟其他說明

如果伺服器掛掉，需要跑 `not_recover_README.R`

```
# 緊急復原檔
# 如果壞掉的話-> 重跑一次 tgc.R (ctrl + A -> ctrl + enter)
# 但是如果請求還是一直跳error和跳出Debug畫面
# 先右上角退出 Rsession 再跑這個檔案去更新掉最後的通知
# 再重跑一次tgc.R
rm(list=ls(all=T)) # 清空資料
if(!require(telegram.bot))install.packages("telegram.bot")

bot <- Bot(token = bot_token("RTelegramBot"))
updates <- bot$getUpdates()
updater <- Updater(token = bot_token("RTelegramBot"))
updater$start_polling()
# bot$sendMessage(Sys.getenv("TG_WEAK"), text = "**懶狗通知*壞掉了幹")
```

該內容很容易理解，就是把整個bot沒有任何function的再跑過一遍，來達到修復的效果