**Documentation**

**Building an R Bot (with “PhorusBot” as a case study)**

1. **Introduction**

PhorusBot is an analytics bot built with R using Telegram API Token. The bot was built to share my projects, contact, social media handle, links to dataset, business news, analytics etc. with users and allow users to have access to other bots such as Youtube Bot, Wikipedia Search Bot, Classical Music Bot etc.

1. **How PhorusBot was Built**

PhorusBot was built by generating API Access Token from Telegram chat bot called BotFather (@BotFather). The following steps should be taken to build a Bot with R;

**Step 1: Generate Access Token**

To generate access token connect with @BotFather (<https://telegram.me/botfather>) and follow the steps described here (<https://core.telegram.org/bots#6-botfather>).

**Step 2: Install and Load the telegram.bot package**

# installing telegram.bot package

install.packages("telegram.bot")

# loading or importing telegram.bot package

library(telegram.bot)

**Step 3: Create an object of Your Bot**

You create an object of your bot, in this case object of PhorusBot, using the code below and replace the “TOKEN” with Telegram Bot’s API Access Token generated from step 1 above. The token should look like this; 1234567890:AAEjAF9yJycaPofoLjKupZFeOQjJd9ers4J

bot <- Bot(token = "TOKEN")

**Step 4: Creating the updater object**

The Updater class continuously fetches new updates from Telegram and passes them on to the Dispatcher class. If you create an Updater object, it will create a Dispatcher.

Create the updater object using the “Updater” function by replacing the “TOKEN” with Telegram Bot’s API Access Token generated from step 1 above, in the code below. The token should look like this 1234567890:AAEjAF9yJycaPofoLjKupZFeOQjJd9ers4J

updater <- Updater(token = "TOKEN")

**Step 5: Create Function(s)**

The next step is to create function(s) that should process a specific type of update that you desired.

In the case of PhorusBot, I created the functions below;

start <- function(bot, update){

bot$sendMessage(chat\_id = update$message$chat\_id,text = print("Hello! Welcome to Phorus Analytics"))

}

email <- function(bot, update){

bot$sendMessage(chat\_id = update$message$chat\_id,text = print("mailphorus@gmail.com"))

}

The goal is to have these functions called every time the Bot (PhorusBot) receives a Telegram message that contains the /start command (for start function) and /email command (for email function). To accomplish that, you can use a CommandHandler (one of the provided Handler sub-classes) and register it in the updaters’s dispatcher (which is done with the + operator):

Note: You can create as many functions as possible, learn more on how to do this using this link (<https://github.com/ebeneditos/telegram.bot/wiki/Introduction-to-the-API>).

**Step 6: Create Handler update the updater**

start\_handler <- **CommandHandler**("start", start)

email\_handler <- CommandHandler("email", email)

updater <- updater + start\_handler + email\_handler

**Step 7: Starting the Bot**

To start the bot run the code below;

updater$**start\_polling**()

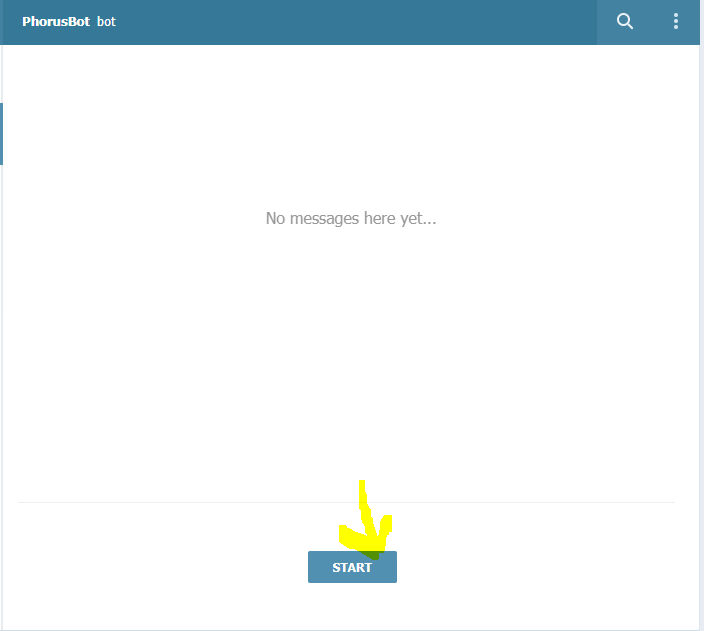
Start a chat with your bot and issue the /start command to the Bot (PhorusBot in my own case), it will reply you. To be able to chat with the Bot you search for the name of your Bot (e.g. PhorusBot) on Telegram application.

1. **How to Interact with PhorusBot**

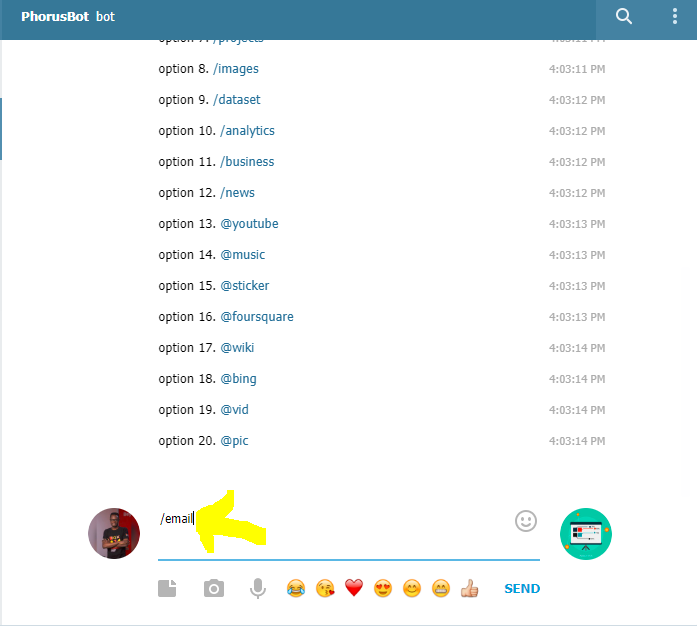
To interact with the PhorusBot or the Bot you created. You either have Telegram application installed or you access web version of Telegram using this URL **(**<https://web.telegram.org/>**).**

**Then, follow the steps below;**

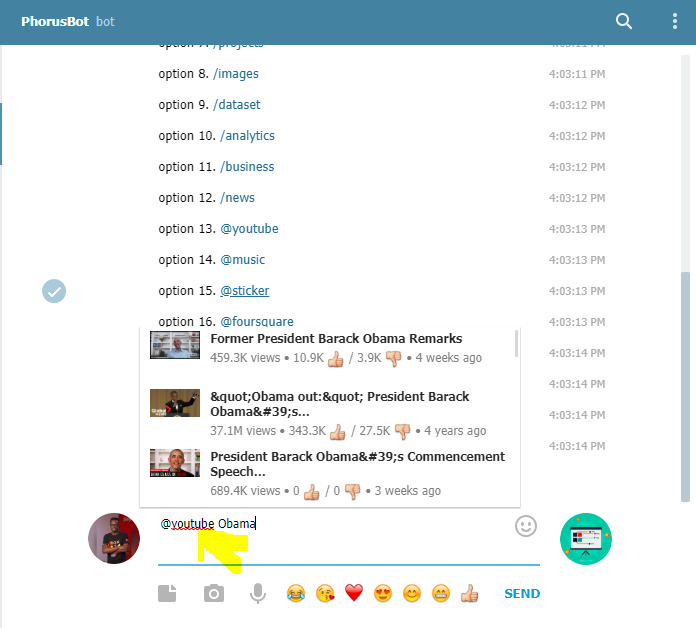
1. Search for the Bot’s name (“PhorusBot” in my own case)
2. Click on the Bot (e.g. PhorusBot)
3. Click “Start”



1. Enter the command of information you want as reply from the Bot (e.g.”/email”)



1. To access other Bots like Youtube Bot, enter “@youtube” and type the title of what you wish to search for (e.g. Obama)



1. You can as well search for other Bots such as @wiki, @sticker, @vid, @pic, @music etc.
2. **Usefulness of Bots**

* **Create custom tools**: A bot may provide you with advert, alerts, weather forecasts, translations, formatting or other services e.g. PhorusBot **(**<https://t.me/>JhonsphorusBot**), StickerBot (**<https://t.me/sticker>**),** VoteBot **(**<https://t.me/vote>**),** LikeBot **(**<https://t.me/like>**) etc.**
* **Get customized notifications and news:** A bot can act as a smart newspaper, sending you relevant content as soon as it's published.
* **Build social services.** A bot could connect people looking for conversation partners based on common interests or proximity.
* **Integrate with other services:** A bot can enrich chats with content from external services.

References:

Ernest Benedito - <https://cran.r-project.org/web/packages/telegram.bot/vignettes/telegrambot-building-a-bot.html>

<https://core.telegram.org/bots>

<https://core.telegram.org/bots#6-botfather>