Telegram Bot

Author: Nick Lee

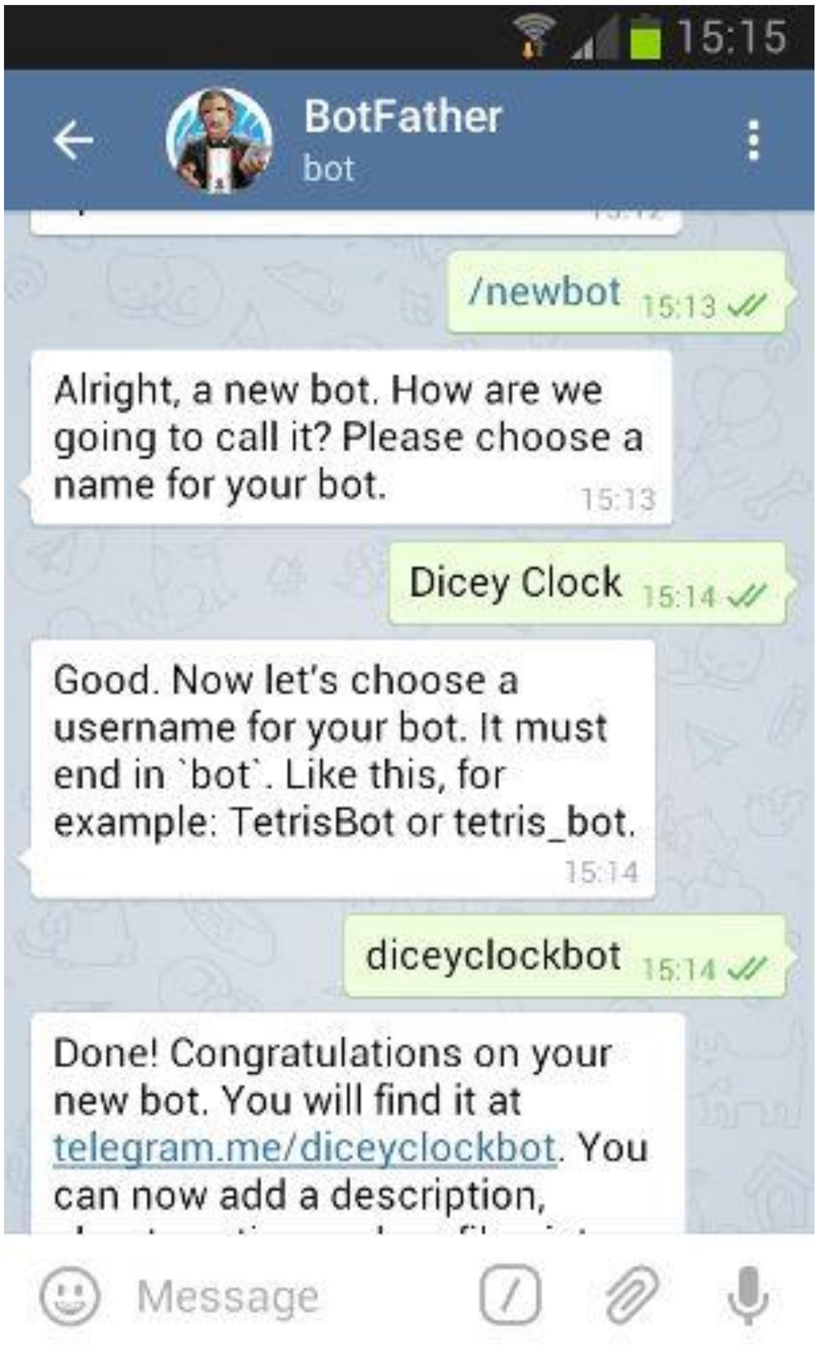
Obtain a token

Open Telegram on the phone, search for a user called **BotFather**. As the name implies, he is the Father of All Bots.

Text him **/newbot**. He will then ask a couple of questions. Follow his guidance and you will get a **token** eventually, which looks something like this:

```
123456789:ABCdefGhIJKlmNoPQRsTUVwxyZ
```

That token identifies the bot account you are going to put on the Pi.



/newbot 15:13

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 15:13

Dicey Clock 15:14

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 15:14

diceyclockbot 15:14

Done! Congratulations on your new bot. You will find it at telegram.me/diceyclockbot. You can now add a description,

Telegram Web Version

We have a problem. How are we going to copy the looooonnnng token from the phone to the Pi? By hand?

Fortunately, you can access Telegram on the web, from a PC.

On your PC, open a browser, go to:

https://web.telegram.org/

It will ask for your phone number, then send you an SMS message containing a code. Enter the code, and you will be led to an interface very similar to your Telegram app.

Find the conversation with BotFather. You should see the token right there. Now, you can easily copy-and-paste the token from the browser window to the Pi, whenever you want, as often as you want.

Test the token

Enter the Pi. Install a Python package that helps you talk to Telegram.

```
$ sudo pip3 install telepot
```

```
$ python3
```

```
>>> import telepot
>>> bot = telepot.Bot('COPY TOKEN FROM BROWSER')
>>> bot.getMe()
```

You should see the bot's info.

```
>>> bot.getUpdates()
[]
```

It is an empty list because nobody has sent it any messages yet.

Receive messages

On the phone, search for your bot, and send it a text message.

Then, try this:

```
>>> from pprint import pprint
>>> response = bot.getUpdates()
>>> pprint(response)
[{'message': {'chat': {'first_name': 'Nick',
                       'id': 00000000,
                       'type': 'private'},
              'date': 1504581971,
              'from': {'first_name': 'Nick',
                       'id': 00000000,
                       'is_bot': False,
                       'language_code':  'en-US'},
              'message_id': 25252,
              'text': 'Hello'},
  'update_id': 100000000}]
```

You should see something roughly similar to the above. Note the **chat** field and **from** field. That is YOU.

Also note the **update_id**. It will be useful very soon.

Receive messages, again

On the phone, send the bot another message.

```
>>> response = bot.getUpdates()
>>> pprint(response)
```

Do you see two messages? How come the first message is still there?

It is because you have to acknowledge to the server that you have received a message. If you don't acknowledge, the server will keep giving the same old messages over and over, because it thinks you have not gotten them successfully.

How to acknowledge? Here is what the **update_id** is for. To say "I have received all update_id below 100000001", you do:

```
>>> bot.getUpdates(offset=100000001)
```

and the server will no longer bother you with update_id smaller than 100000001. Next update_id you get will be 100000001.

Your update_id will NOT be 100000000 and 100000001. Do not copy me blindly.

An easier way to receive messages

In a program, it is troublesome to keep checking messages and managing **offset**. Telepot can take care of that for you, and notify you whenever new messages arrive.

```
>>> from telepot.loop import MessageLoop
>>> from pprint import pprint
>>> def handle(msg):
...     pprint(msg)
...
>>> MessageLoop(bot, handle).run_as_thread()
```

Send some messages to the bot. Look at them arriving on the screen.

Send messages

```
>>> bot.sendMessage(999999999, 'Good evening!')
```

Use your own ID instead of 999999999. Do not copy me blindly.

Do you receive the message on the phone?

Can you make a bot that echo what you say? That is, it sends you back anything you send it?

echobot.py

```
import sys
import time
import telepot
from telepot.loop import MessageLoop

def handle(msg):
    chat_id = msg['chat']['id'] ❶
    text = msg['text']

    bot.sendMessage(chat_id, text) ❷

TOKEN = sys.argv[1] ❸

bot = telepot.Bot(TOKEN)
MessageLoop(bot, handle).run_as_thread()
print('Listening ...')

while 1: ❹
    time.sleep(1)
```

❶ Extract message info

❷ Reply sender with the original text

❸ Get token as a command-line argument. It is not a good habit to embed token in source code, because token is supposed to be kept secret.

❹ Keep the program running. Otherwise, we would lose the bot and its message notification.

Dicey Clock

Let's make a more capable bot. Imagine a bot that responds to two commands:

- **/roll** - reply with a random integer between 1 and 6, like rolling a dice.
- **/time** - reply with the current time, like a clock.

I call this a Dicey Clock.

The commands start with a slash (/). We will see why in a moment.

How would you program?

diceyclock.py

```
import sys
import time
import random
import datetime
import telepot
from telepot.loop import MessageLoop

def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']

    if command == '/roll':
        n = random.randint(1,6) ❶
        bot.sendMessage(chat_id, n)
    elif command == '/time':
        d = str(datetime.datetime.now()) ❷
        bot.sendMessage(chat_id, d)

TOKEN = sys.argv[1]

bot = telepot.Bot(TOKEN)
MessageLoop(bot, handle).run_as_thread()
print('Listening ...')

while 1:
    time.sleep(1)
```

❶ Get a random integer between 1 and 6

❷ Get the current time and turn it into a textual representation

/setcommands

According to Telegram's convention, commands for bots start with a slash (/). You may configure the bot to present a list of commands, so you don't have to type them every time.

Go back to the conversation with BotFather.

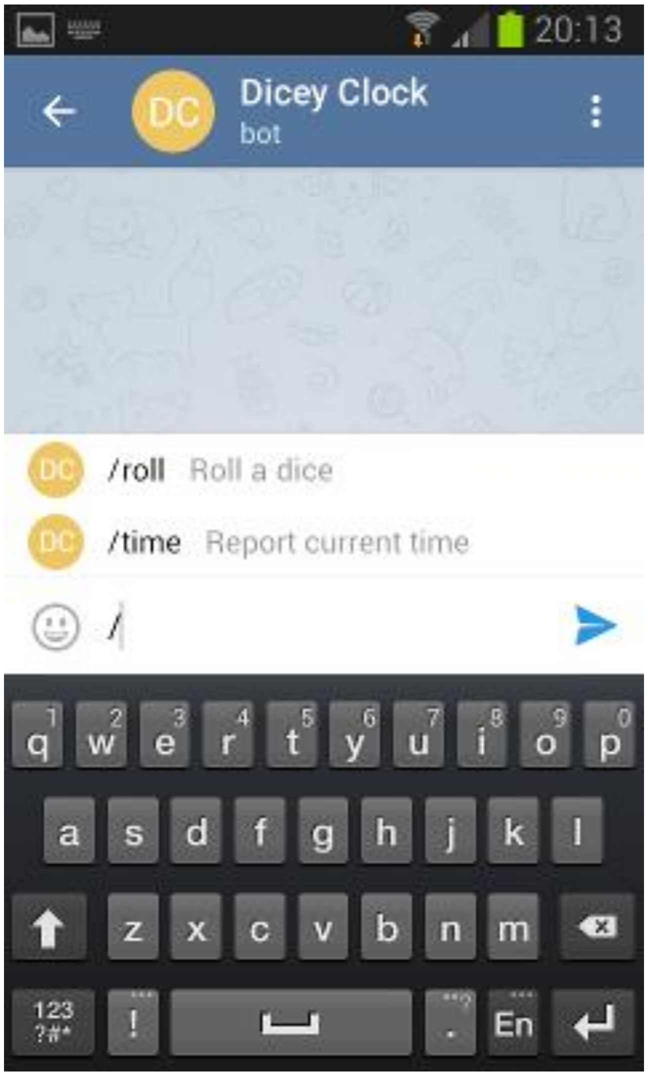
Text him **/setcommands**. Follow his instructions, and somewhere along the way, he should ask you for a list of command descriptions. In response, send:

roll - Roll a dice
time - Report current time

Start each line with **lowercase**, and **no slash**.

After /setcommands is complete, go back to your Dicey Clock bot. Type a slash, do you see the command list as shown on the opposite page?

It probably takes a few minutes for Telegram to update. If you don't want to wait, just exit, kill (force stop), and re-launch the Telegram app. You should see the command list right away.



/roll Roll a dice

/time Report current time

/