

Tutorial de bot python

Debemos crear un ambiente a través del módulo `venv` con el siguiente comando:

```
python -m venv env

#LUEGO DEBE ACTIVAR EL AMBIENTE

source $PWD/env/bin/activate
```

Instalar libreria de telegram

La librería se llama `pytelegrambotapi` y se instala así: `pip install pytelegrambotapi`

Uso del token

Debemos importar el token por seguridad desde otro archivo python, ejemplo:

- Creamos un **config.py** y generamos una constante con el valor del token, esto es un dato string
- Creamos un archivo **main.py** importamos la constante generada

Config.py

```
VAR_TK = '6493247672:AAELFqWHbi2EbYKvrrRc6Wg-N_U8-9YaC4w'
```

main.py

```
import config
import telebot

bot = telebot.TeleBot(config.VAR_TK)
```

Con nuestro bot podemos programar cómo responder a los msj de los usuarios, qué hace el bot si envía un comando y el resto, por ej si envió una foto, texto, etc.

Crear comando

Con esto vamos a definir qué queremos que haga el bot con el usuario le envíe un comando, ej /start

1. Creamos un decorador `@bot.message_handler(commands=["comandos"])`
2. Creamos una función por ej `def cmd_start(message)`
3. Definimos qué queremos que haga el bot cuando el usuario envíe el comando (crear bloque de código), todo esto a través del objeto `bot` y el método `reply_to()`

Ejemplo

```
import config
import telebot

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help
#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Pruebas desde los command') #CITA EL MENSAJE ENVIADO
AL BOT Y DA UNA RESPUESTA, RECIBE 1 PARAM

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,
```

Nota: Es de aclarar que el método `reply_to` cita el mensaje.

Objeto message

Dentro del objeto **message** se almacena muchas interacciones del chat, por ej el id del chat, el texto que envía el usuario, etc.

Reaccionar a mensajes que no son comandos

Se utiliza el mismo decorador pero en el argumento va un parámetro distinto llamado

```
content_types=['text', 'photo']
```

1. Creamos un decorador `@bot.message_handler(content_types=['text'])`
2. Creamos una función por ej `def bot_msj_texto(message):`
3. Definimos qué queremos que haga el bot cuando el usuario envíe **TEXTO** (crear bloque de código), todo esto a través del objeto `bot` y el método `send_message()`
4. El método del punto 3 recibe como argumento **un chat id que es el identificador único que tiene cada chat en Telegram** el objeto **message** es quien recibe o alberga dicha información y más, partimos desde el objeto chat hasta llegar al atributo id ejemplo:

```
message.chat.id
```

Ejemplo

Vamos a condicionar según el texto que envíe el usuario, por ejemplo si envía un comando debes decir que ese comando no existe porque no se pasó en la función de comandos, aquí se utiliza el objeto **message** con el atributo **text** y con este a su vez podemos usar el método **startswith**

```
import config
import telebot
```

```

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help
#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list de comandos
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Bienvenid@, este BOT te dará noticias útiles sobre
el mundo tecnológico.') #CITA EL MENSAJE ENVIADO AL BOT Y DA UNA RESPUESTA,
RECIBE 1 PARAM

#REACCIONA A OTRAS COSAS QUE NO SON COMANDOS, EJ text, audio, document, photo,
sticker, video. etc.
@bot.message_handler(content_types=['text'])
def bot_msj_texto(message):
    """Gestiona los mensajes de texto recibidos"""

    # bot.send_message(message.chat.id, "Suscribete") #ENVIA UN MSJ SIN CITAR,
RECIBE 2 PARAM, EL OBJETO MESSAGE TIENE UN METODO
    if message.text.startswith('/'): #EN EL ATRIBUTO TEXT DEL OBJETO MESSAGE
SE ALMACENA LO QUE EL USUARIO ENVIA
        bot.send_message(message.chat.id, 'Comando no disponible') #NO CITA EL
MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID
    else:
        bot.send_message(message.chat.id, 'Todo bien') #CITA EL MENSAJE ENVIADO
RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y ATRIBUTO CHAT.ID

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,

```

Formatear texto

Puede pasar como parámetro a `send_message(var_mytexfull, parse_mode="html")` donde **var_mytexfull** es la variable con el contenido del texto formateado, es obligatorio pasar el **parse_mode** para que si se aplique el formato.

Editar texto

Cada mensaje tiene un identificador único, cuando nosotros enviamos un mensaje a través de `bot_send_message()` además de enviarlo nos devolverá un objeto que contiene muchos datos tales como el id del mensaje, podemos obtener el id del mensaje enviado por nosotros guardando su resultado en una variable.

Ejemplo

```

import config
import telebot
import time

```

```

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help
#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list de comandos
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Bienvenid@, este BOT te dará noticias útiles sobre
el mundo tecnológico.') #CITA EL MENSAJE ENVIADO AL BOT Y DA UNA RESPUESTA,
RECIBE 1 PARAM

@bot.message_handler(content_types=['text'])
def bot_mod_texto(message):
    """Modifica los mensajes de texto recibidos"""

    # bot.send_message(message.chat.id, "Suscribete") #ENVIA UN MSJ SIN CITAR,
RECIBE 2 PARAM, EL OBJETO MESSAGE TIENE UN METODO
    if message.text.startswith('/'): #EN EL ATRIBUTO TEXT DEL OBJETO MESSAGE
SE ALMACENA LO QUE EL USUARIO ENVIA
        x = bot.send_message(message.chat.id, 'Comando no disponible') #NO CITA
EL MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID
        time.sleep(3)
        bot.edit_message_text(f"<b>SU COMANDO {x.message_id} NO FUE ENCONTRADO
EN NUESTRA BASE DE DATOS</b>", message.chat.id, x.message_id,
parse_mode="html")
    else:
        bot.send_message(message.chat.id, "Su texto fue recibido") #NO CITA EL
MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,

```

Explicación

En este caso el método `edit_message_text` recibe como argumento lo siguiente

- Texto que se mostrará después de la modificación
- El chat id del mensaje
- El id del texto que se obtiene con el objeto de la variable x que creamos
- Opcional-mente si está enviando html debe parsear

```

bot.edit_message_text("<b>SU COMANDO NO FUE ENCONTRADO EN NUESTRA BASE DE
DATOS</b>", message.chat.id, x.message_id, parse_mode="html")

```

Eliminar texto enviado por el bot

Al igual que con editar usamos el método `delete_message` que recibe 2 parámetros, **el id del chat y el id del mensaje** este último lo obtiene desde el objeto **message**

```
import config
import telebot
import time

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help
#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list de comandos
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Bienvenid@, este BOT te dará noticias útiles sobre
el mundo tecnológico.') #CITA EL MENSAJE ENVIADO AL BOT Y DA UNA RESPUESTA,
RECIBE 1 PARAM

@bot.message_handler(content_types=['text'])
def bot_mod_texto(message):
    """Modifica los mensajes de texto recibidos"""

    # bot.send_message(message.chat.id, "Suscribete") #ENVIA UN MSJ SIN CITAR,
RECIBE 2 PARAM, EL OBJETO MESSAGE TIENE UN METODO
    if message.text.startswith('/'): #EN EL ATRIBUTO TEXT DEL OBJETO MESSAGE
SE ALMACENA LO QUE EL USUARIO ENVIA
        x = bot.send_message(message.chat.id, 'Comando no disponible') #NO CITA
EL MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID
        time.sleep(3)
        bot.delete_message(message.chat.id, x.message_id)

    else:
        x = bot.send_message(message.chat.id, "Su texto fue recibido") #NO CITA
EL MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,
```

Eliminar texto enviado por el usuario

Usamos el método `delete_message` que recibe 2 parámetros, **el id del chat y el id del mensaje del usuario**

```
import config
import telebot
```

```

import time

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help
#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list de comandos
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Bienvenid@, este BOT te dará noticias útiles sobre
el mundo tecnológico.') #CITA EL MENSAJE ENVIADO AL BOT Y DA UNA RESPUESTA,
RECIBE 1 PARAM

@bot.message_handler(content_types=['text'])
def bot_mod_texto(message):
    """Modifica los mensajes de texto recibidos"""

    # bot.send_message(message.chat.id, "Suscribete") #ENVIA UN MSJ SIN CITAR,
RECIBE 2 PARAM, EL OBJETO MESSAGE TIENE UN METODO
    if message.text.startswith('/'): #EN EL ATRIBUTO TEXT DEL OBJETO MESSAGE
SE ALMACENA LO QUE EL USUARIO ENVIA
        x = bot.send_message(message.chat.id, 'Comando no disponible') #NO CITA
EL MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID
        time.sleep(2)
        bot.delete_message(message.chat.id, message.message_id) #BORRA MSJ DEL
USUARIO
        bot.delete_message(message.chat.id, x.message_id) #BORRA MSJ DEL BOT

    else:
        x = bot.send_message(message.chat.id, "Su texto fue recibido") #NO CITA
EL MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,

```

Envíar fotos

Como hablamos anteriormente, en los parámetros del decorador se envía a qué queremos reaccionar, hemos estado reaccionando a texto, entonces cuando el usuario escriba algo le enviará una foto, usamos el método `send_photo` del objeto bot.

Ejemplo

```

import config
import telebot
import time

```

```

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help
#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list de comandos
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Bienvenid@, este BOT te dará noticias útiles sobre
el mundo tecnológico.') #CITA EL MENSAJE ENVIADO AL BOT Y DA UNA RESPUESTA,
RECIBE 1 PARAM

#REACCIONA A OTRAS COSAS QUE NO SON COMANDOS, EJ text, audio, document, photo,
sticker, video. etc.
@bot.message_handler(content_types=['text'])
def bot_msj_texto(message):
    """Gestiona los mensajes de texto recibidos"""

    # bot.send_message(message.chat.id, "Suscribete") #ENVIA UN MSJ SIN CITAR,
RECIBE 2 PARAM, EL OBJETO MESSAGE TIENE UN METODO
    if message.text.startswith('/'): #EN EL ATRIBUTO TEXT DEL OBJETO MESSAGE
SE ALMACENA LO QUE EL USUARIO ENVIA
        bot.send_message(message.chat.id, 'Comando no disponible') #NO CITA EL
MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID
    else:
        with open('./images/docker.png', 'rb') as photo:
            bot.send_photo(message.chat.id, photo, "DOCKER PRO")
        photo.close

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,

```

Explicación

Usamos `with open` para abrir el archivo y posteriormente pasarlo como argumento, el tercer argumento es el texto que el usuario verá reflejado.

Envíar archivo

Al igual que en el paso anterior de enviar fotos usaremos el método `send_document`

Ejemplo

```

import config
import telebot
import time

bot = telebot.TeleBot(config.VAR_TK)

#RESPONDE AL COMANDO /start /ayuda /help

```

```

#UTILIZAMOS UN DECORADOR - USAMOS EL METODO message_handler y como param
commands y le pasamos una list de comandos
@bot.message_handler(commands=['start', 'ayuda', 'help'])
def cmd_start(message):
    """Da la bienvenida"""
    bot.reply_to(message, 'Bienvenid@, este BOT te dará noticias útiles sobre
el mundo tecnológico.') #CITA EL MENSAJE ENVIADO AL BOT Y DA UNA RESPUESTA,
RECIBE 1 PARAM

#REACCIONA A OTRAS COSAS QUE NO SON COMANDOS, EJ text, audio, document, photo,
sticker, video. etc.
@bot.message_handler(content_types=['text'])
def bot_msj_texto(message):
    """Gestiona los mensajes de texto recibidos"""

    # bot.send_message(message.chat.id, "Suscribete") #ENVIA UN MSJ SIN CITAR,
RECIBE 2 PARAM, EL OBJETO MESSAGE TIENE UN METODO
    if message.text.startswith('/'): #EN EL ATRIBUTO TEXT DEL OBJETO MESSAGE
SE ALMACENA LO QUE EL USUARIO ENVIA
        bot.send_message(message.chat.id, 'Comando no disponible') #NO CITA EL
MENSAJE ENVIADO RECIBE 2 PARAM EL CHATID - SE OBTIENE DEL OBJETO MESSAGE Y
ATRIBUTO CHAT.ID
    else:
        with open('./docs/bot.pdf', 'rb') as mydoc:
            bot.send_document(message.chat.id, mydoc, caption="PDF bot python")
        mydoc.close

if __name__ == '__main__':
    print('Iniciando bot')
    bot.infinity_polling() #ESCUCHA O COMPRUEBA SI SE RECIBEN MSJ NUEVOS, TODO
SE DETIENE AQUI,

```