Tutorial pyTelegramBotAPI

#0 – Introducción a los bots de Telegram

¿Qué es Telegram?

- Aplicación de mensajería instantánea polivalente
- Servidores centralizados
- Disponible para dispositivos móviles, ordenadores y web, sin restricciones absurdas al conectarnos desde diferentes dispositivos.
- Creación e identificación de cuentas: número de teléfono móvil
- Números de teléfono ocultos a personas desconocidas
- Grupos y canales
- Servicios añadidos: bots, reproducción de música y vídeo, envío de archivos, almacenamiento en la nube, pagos, juegos…

github.com/enforcerzhukov

¿Qué son los bots de Telegram?

- Se identifican como usuarios normales, con quienes se puede hablar e interactuar
- Según cómo estén programados, responderán a ciertos mensajes o comandos brindando información o realizando ciertas acciones
- Permiten multitud de posibilidades: consultar información, recibir notificaciones, realizar tareas obteniendo resultados...

Limitaciones de los bots

- No pueden iniciar conversaciones: sólo pueden enviar mensajes a usuarios que los hayan inicializado o contactado con ellos.
- Pueden ser bloqueados por los usuarios.
- No pueden interactuar con otros bots.
- Reciben todos los mensajes que les envían sus usuarios. En grupos también, si tienen deshabilitada la privacidad de grupos.
- No tienen acceso a datos sensibles del usuario ni dispositivo, como número de teléfono o localización, salvo que el bot lo pida y el cliente se lo proporcione.
- Su nombre de usuario (@username) siempre termina en “bot” (salvo los bots oficiales).

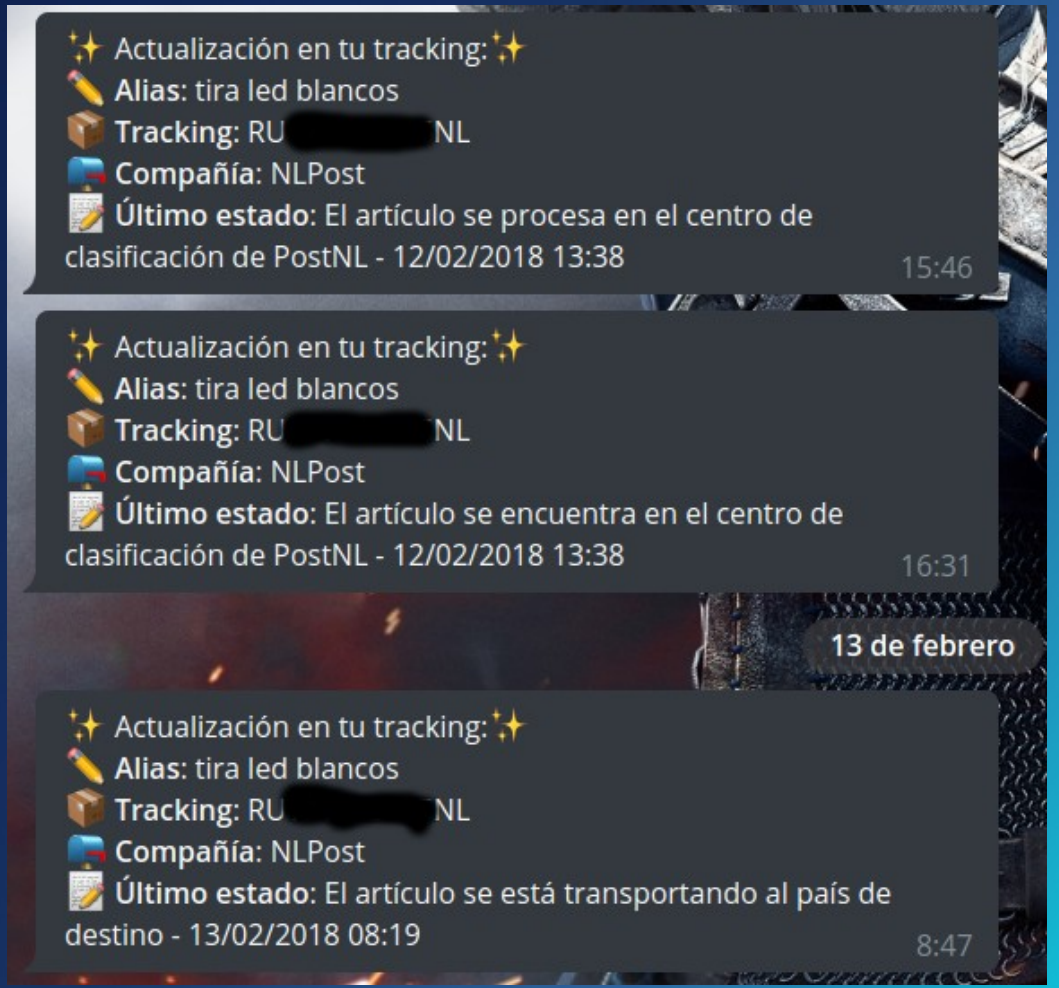
github.com/enforcerzhukov

Ejemplos de bots

@MiTrackingBot

seguimiento de trackings de envíos, recibiendo las actualizaciones de los mismos

Compatible con más de 100 empresas de transporte a nivel mundial



github.com/enforcerzhukov

Ejemplos de bots

@vigobusbot

@madbusbot

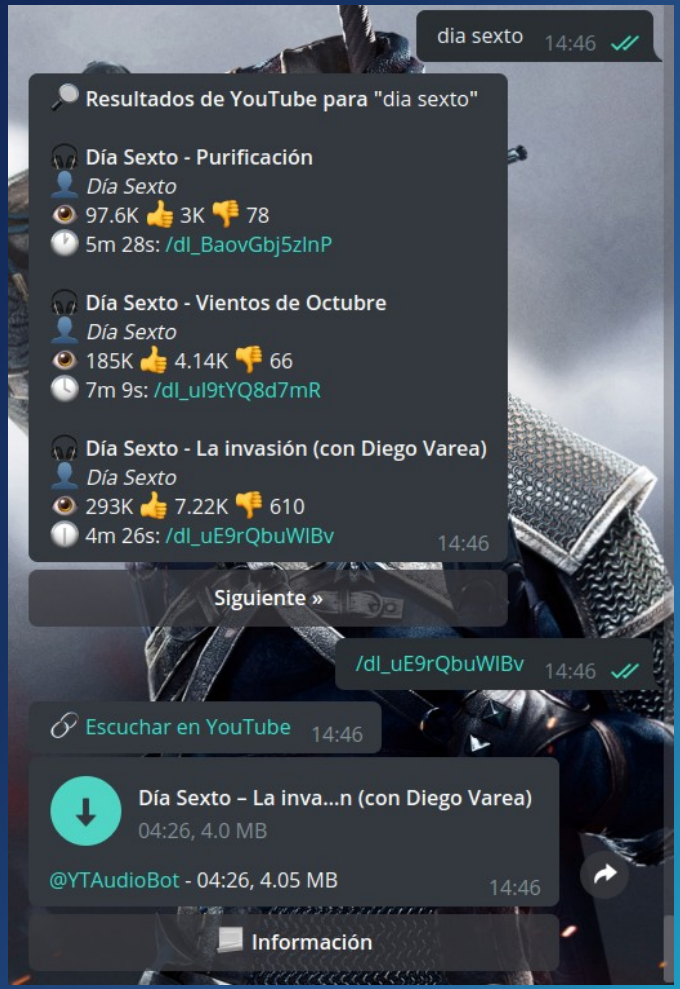
Bots para obtener estimaciones de llegada de autobuses a paradas

github.com/enforcerzhukov

Ejemplos de bots

@YTAudioBot

Descarga vídeos de YouTube como audio mp3



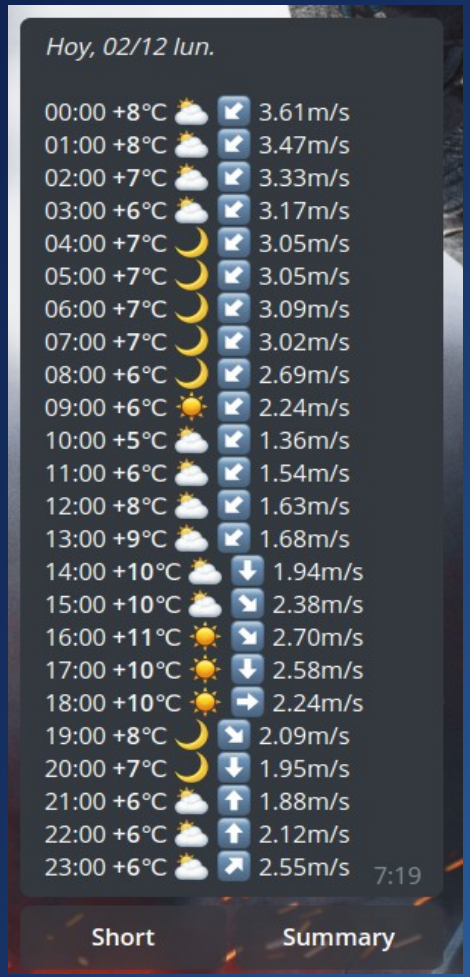
github.com/enforcerzhukov

Ejemplos de bots

@weatherman_bot

Consulta de previsiones meteorológicas

Envía automáticamente la previsión del día



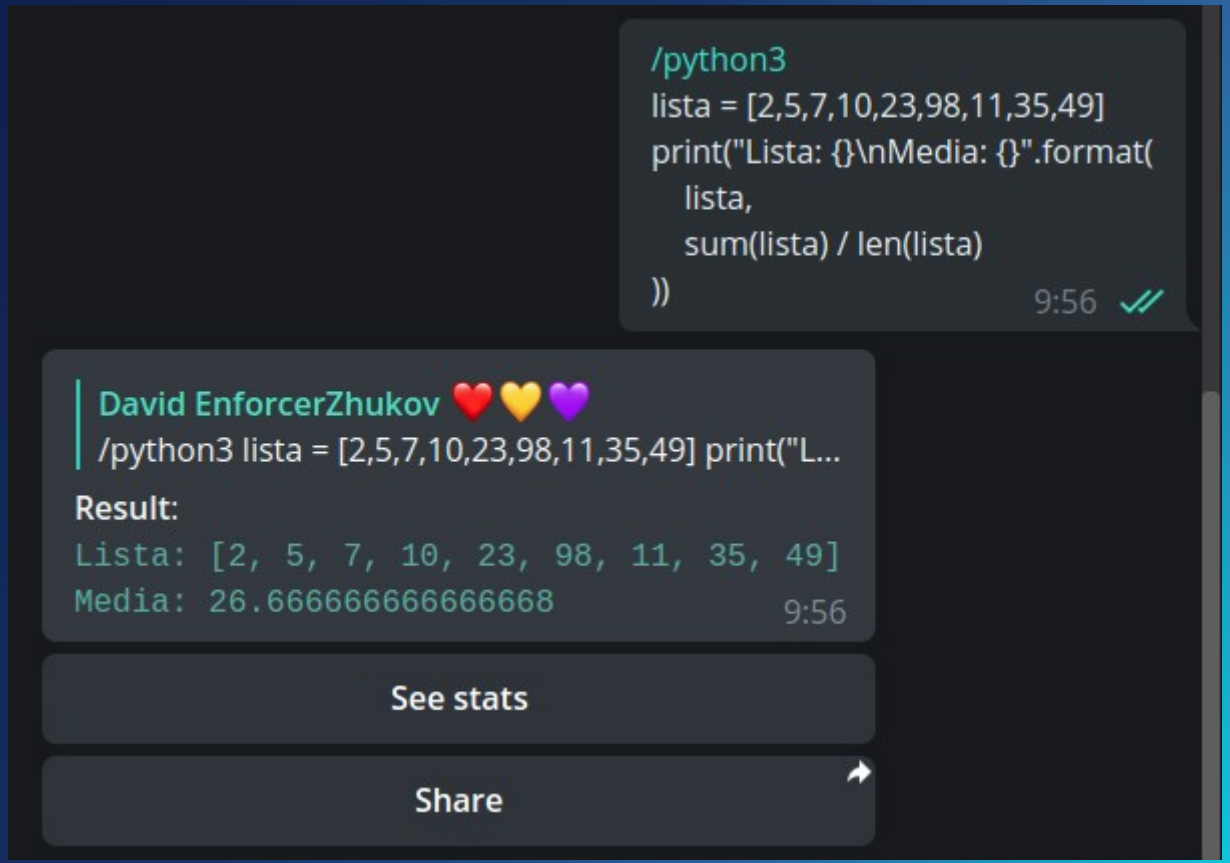
github.com/enforcerzhukov

Ejemplos de bots

@rextester_bot

Ejecuta código y muestra el resultado.

Compatible con Python, Java, Javascript, Ruby, C, C#, C++, MySQL, Scala…



github.com/enforcerzhukov

Interacción cliente-bot

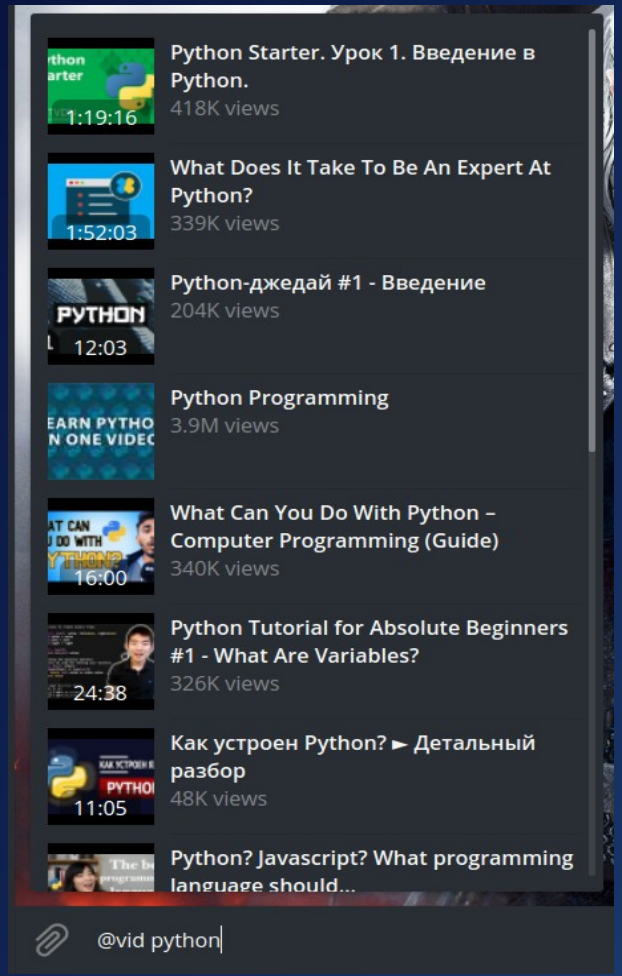
- **Comandos:** empiezan con una barra / seguido del comando, y pueden incluir texto a continuación. Por ejemplo: */help* , */buscar tutoriales de python*
  - En grupos pueden requerir indicar el username del bot (*/help@mibot*)
  - Limitaciones de los comandos: sin espacios, máx. 32 caracteres, sólo letras, números y guiones bajos _
  - Siempre que un usuario inicializa un bot, envía el comando */start*
- **Texto directo:** el bot recibe cualquier mensaje o contenido (incluyendo archivos y multimedia). En grupos depende de la configuración de privacidad del bot (por defecto sólo pueden ver mensajes con sus comandos o menciones al bot).
- **Modo Inline:** se pueden lanzar consultas al bot sin hablar directamente con él ni enviar comandos. Especialmente útil para realizar búsquedas.

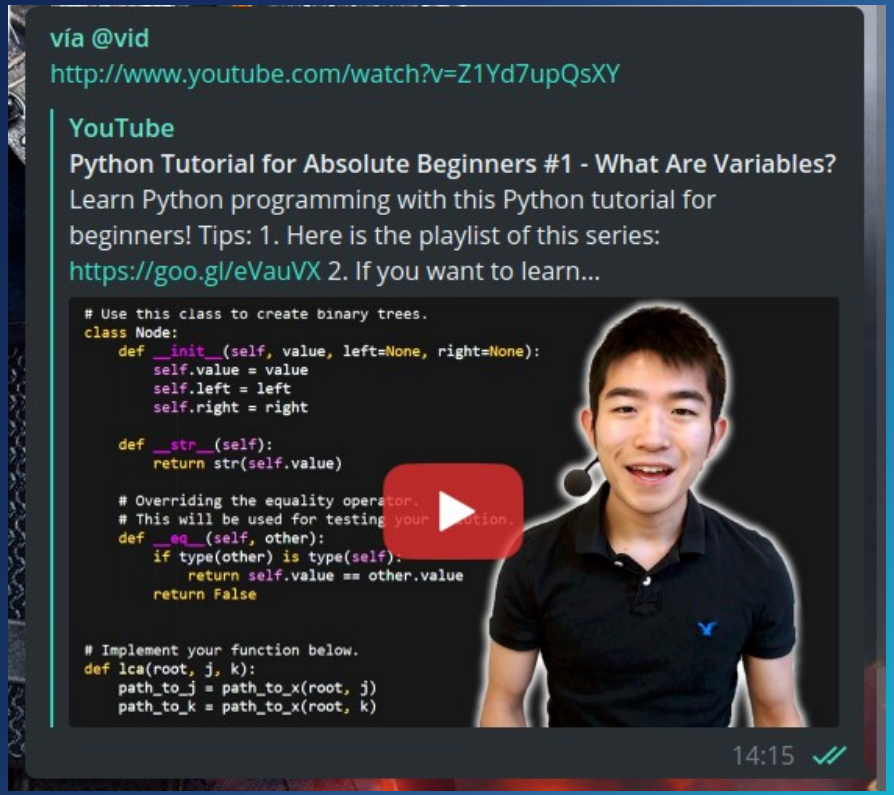
github.com/enforcerzhukov

Ejemplos de bots Inline

@vid

búsqueda de vídeos en YouTube



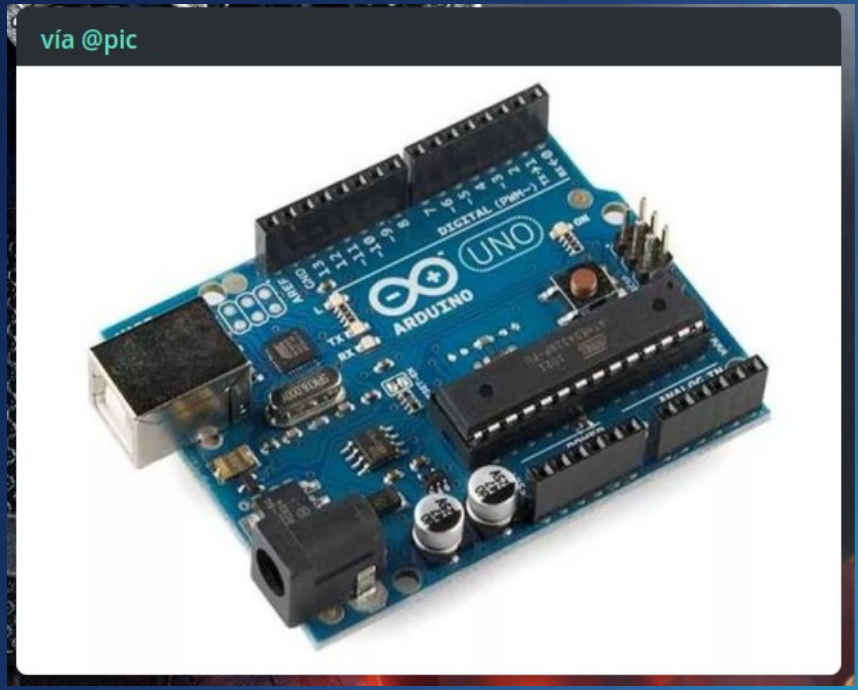


github.com/enforcerzhukov

Ejemplos de bots Inline

@pic

búsqueda de imágenes vía Yandex



github.com/enforcerzhukov

Características del modo Inline

- Empleo: mencionar al bot (@nombrebot), espacio y término de búsqueda.
- Accesible desde cualquier grupo o chat privado, incluso sin que el bot esté presente.
- El bot no tiene acceso a los mensajes del grupo o conversación donde se le llama.
- Se genera un listado con resultados (texto, imágenes, enlaces, multimedia...).
- Al seleccionar un resultado se envía un mensaje al grupo (este mensaje se envía con el cliente que realizó la búsqueda, indicando además “vía @nombrebot”).

github.com/enforcerzhukov

Funcionamiento interno de un bot

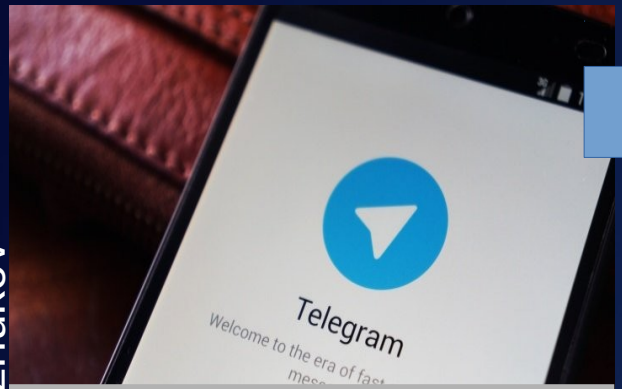
Partes involucradas:

- **Cliente:** interactúa con el bot, enviándole peticiones
- **Servidor del bot:** procesa las peticiones del cliente, actuando en consecuencia (enviando respuestas, contenido...)
- **Servidores de Telegram:** intermediario entre cliente y bot

github.com/enforcerzhukov

Funcionamiento interno de un bot

Un usuario interactúa con un bot enviando un comando



Cliente envía comando /help

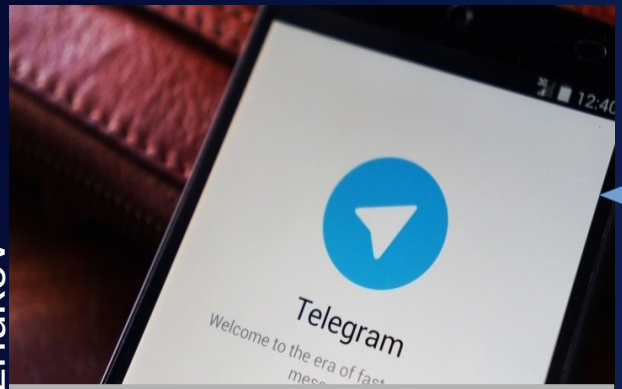
Servidor del bot recibe notificación: comando /help enviado por cliente

Telegram recibe el mensaje
Lo notifica al servidor del bot

github.com/enforcerzhukov

Funcionamiento interno de un bot

El bot responde al comando /help enviando un mensaje con ayuda



Cliente recibe el mensaje

Telegram recibe el mensaje del bot
Se lo envía a cliente

Servidor del bot reacciona ante /help:
debe enviarle a cliente un mensaje de texto

github.com/enforcerzhukov

Obtención de actualizaciones por el bot

Los mensajes de Cliente → Bot deben llegar al programa de control del bot de alguna forma. Existen dos métodos para obtener actualizaciones de clientes:

- **Long Polling** (recomendada para empezar y pequeños proyectos)
- **Webhook** (recomendada para grandes proyectos finales)

github.com/enforcerzhukov

Obtención de actualizaciones por el bot

Long Polling

Cada cierto tiempo se le pregunta a Telegram por las actualizaciones.

Telegram nos responde con las nuevas actualizaciones.

Ventajas:

- Fácil de usar
- No requiere crear servidor (el programa es “cliente” hacia Telegram)

Desventajas:

- Poco eficiente

github.com/enforcerzhukov

Obtención de actualizaciones por el bot

Webhook

Creamos un servidor web al que Telegram enviará las novedades únicamente cuando las haya.

Ventajas:

- Más eficiente y rápido que el long polling

Desventajas:

- Requiere crear un servidor para que Telegram tenga acceso, con todos los requisitos que esto conlleva

github.com/enforcerzhukov

API de Telegram para bots

- Introducción para el desarrollo: https://core.telegram.org/bots
- Documentación: **https://core.telegram.org/bots/api**
- Se basa en el envío de requests a la API de Telegram: `https://api.telegram.org/bot<token>/<METHOD>`
- Sólo funciona con HTTPS

Librerías para crear bots de Telegram

- Implementan las llamadas a la propia API de Telegram en varios lenguajes de programación
- Facilitan el desarrollo de un bot, evitándonos el proceso de generar los requests y procesar las respuestas devueltas.

Ejemplos:

- Python: pyTelegramBotAPI, python-telegram-bot
- Java: TelegramBots, java-telegram-bot-api
- C#: Telegram.Bot
- C++: tgbot-cpp, telegram-bot-api
- NodeJS: node-telegram-bot-api, telegraf

github.com/enforcerzhukov

Registrar un bot

Todo bot debe registrarse en Telegram para que aparezca en la aplicación y cualquier cliente pueda utilizarlo.

Esto se realiza desde **@BotFather**, un bot oficial que nos permite registrar bots y modificarlos.

I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual.

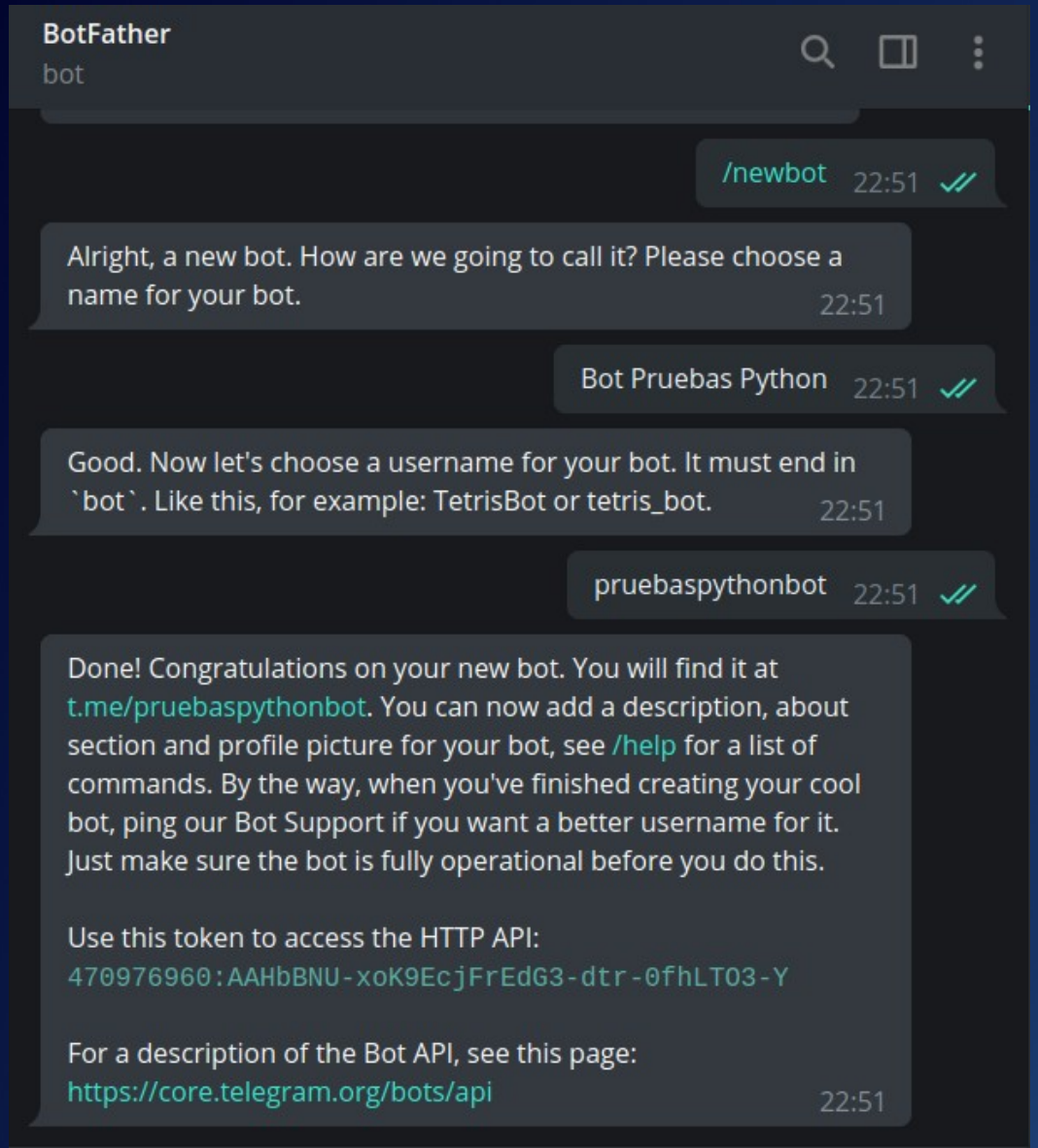
You can control me by sending these commands:

/newbot - create a new bot
/mybots - edit your bots **[beta]**
/mygames - edit your games **[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
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

github.com/enforcerzhukov



Para registrar un bot se piden siempre dos parámetros obligatorios:

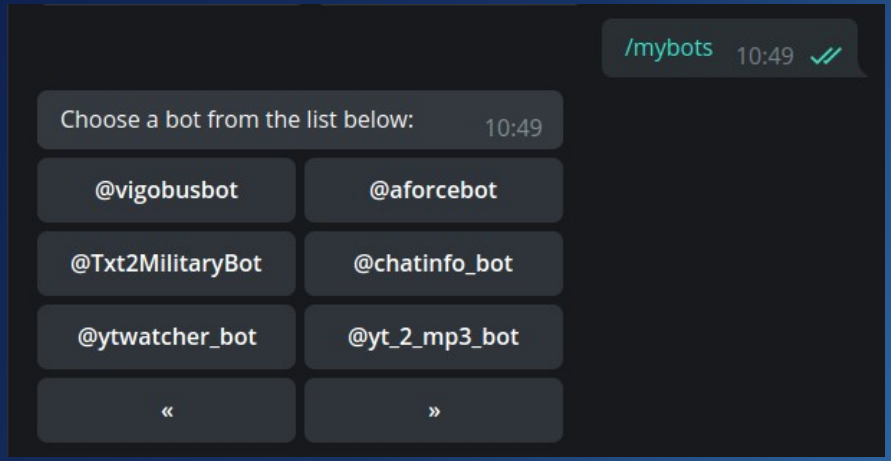
- Name – nombre completo
- Username (@nombrebot) – debe terminar siempre en *bot*

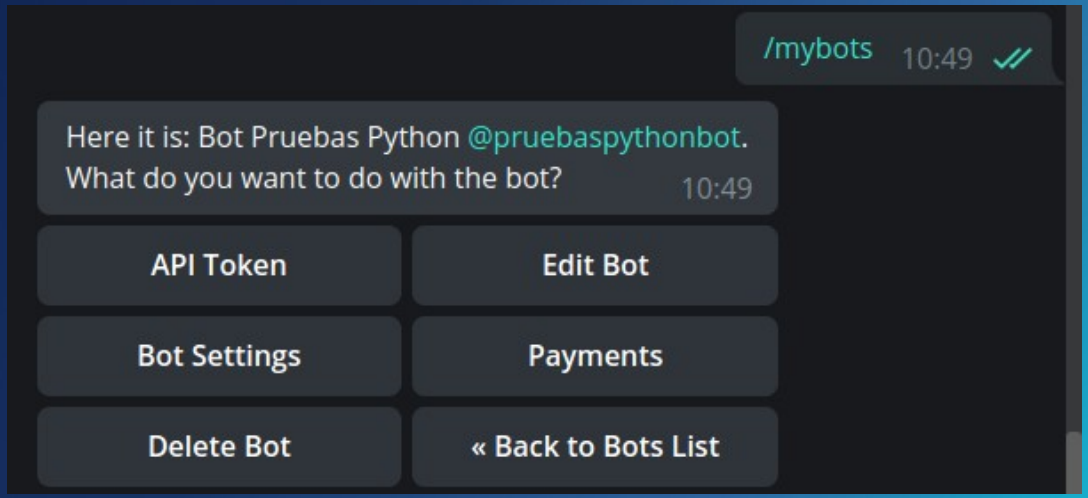
github.com/enforcerzhukov

Configurando el bot

Desde /mybots podemos ver todos los bots que hayamos registrado.

Pulsando sobre uno de ellos accedemos a todas sus configuraciones.





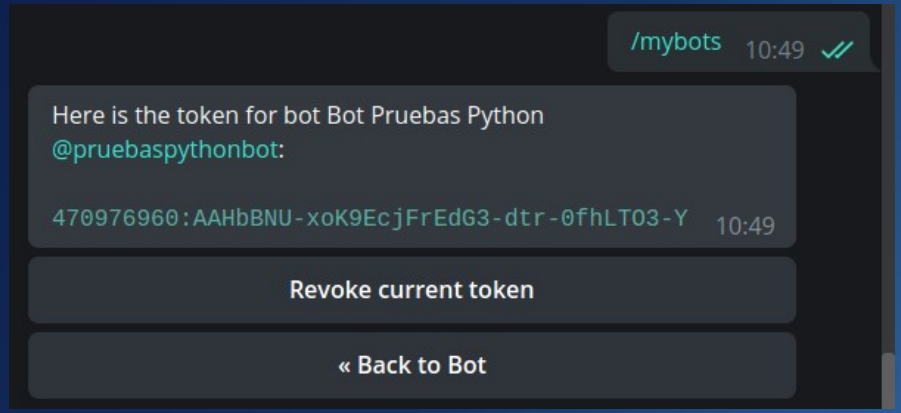
github.com/enforcerzhukov

La TOKEN

Es un código que identifica y autentica al bot cuando usamos la API de Telegram.

Desde la sección “API Token” de BotFather podemos ver la TOKEN actual.

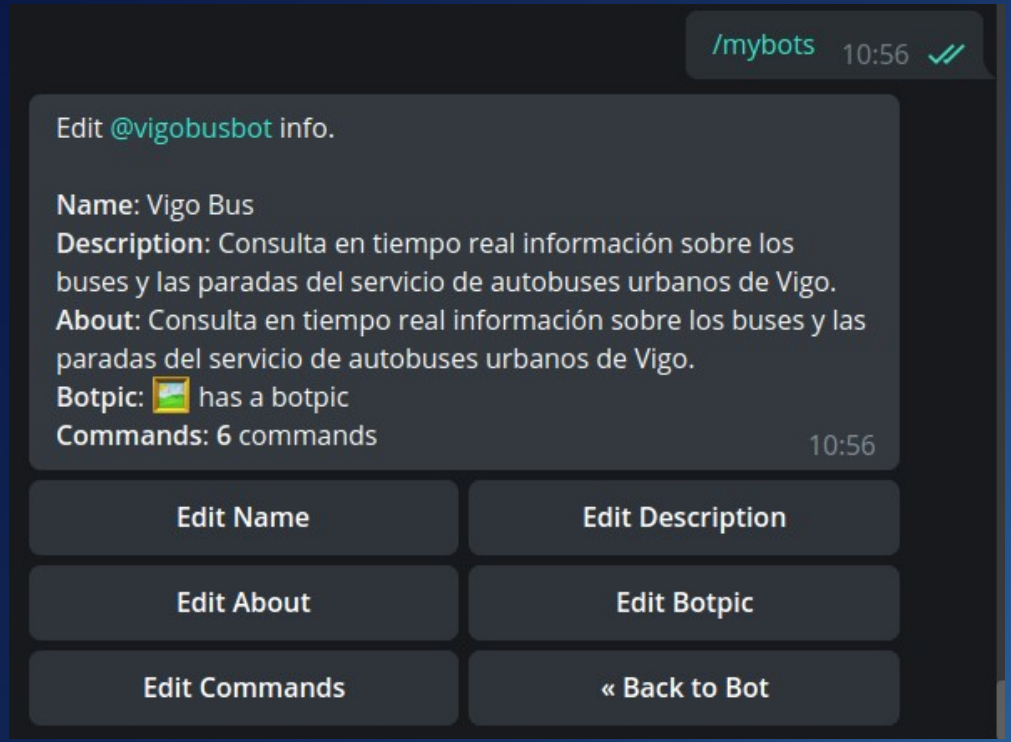
También es posible generar otra (“Revoke current token”), lo cual inhabilitará la anterior.



github.com/enforcerzhukov

Personalización

Desde “Edit bot” accedemos a varias opciones para personalizar el apartado estético del bot:



github.com/enforcerzhukov

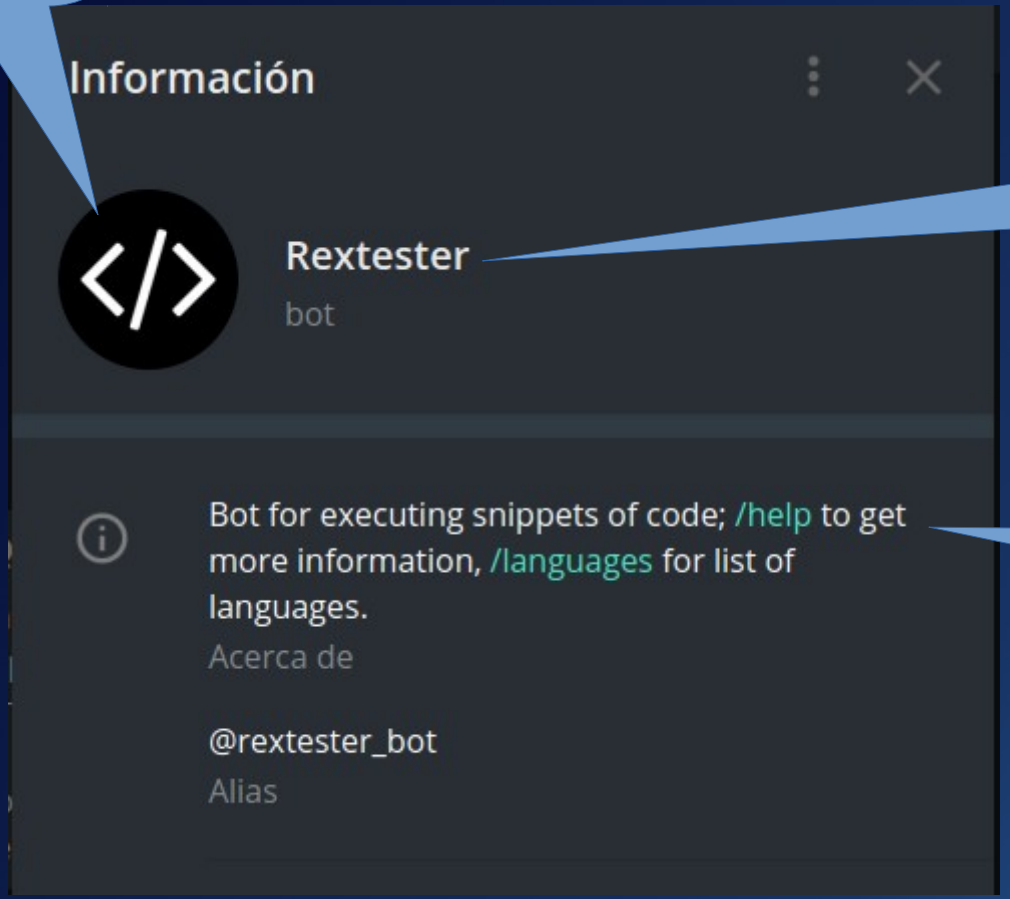
Personalización

Desde “Edit bot” accedemos a varias opciones para personalizar el apartado estético del bot:

- Edit Name: Nombre público (no @username)
- Edit Description: Descripción antes de inicializar el bot
- Edit About: Descripción en el perfil del bot
- Edit Botpic: Imagen de perfil (avatar) del bot
- Edit Commands: Registrar listado de comandos del bot (sólo estético, no funcional)

github.com/enforcerzhukov

Personalización



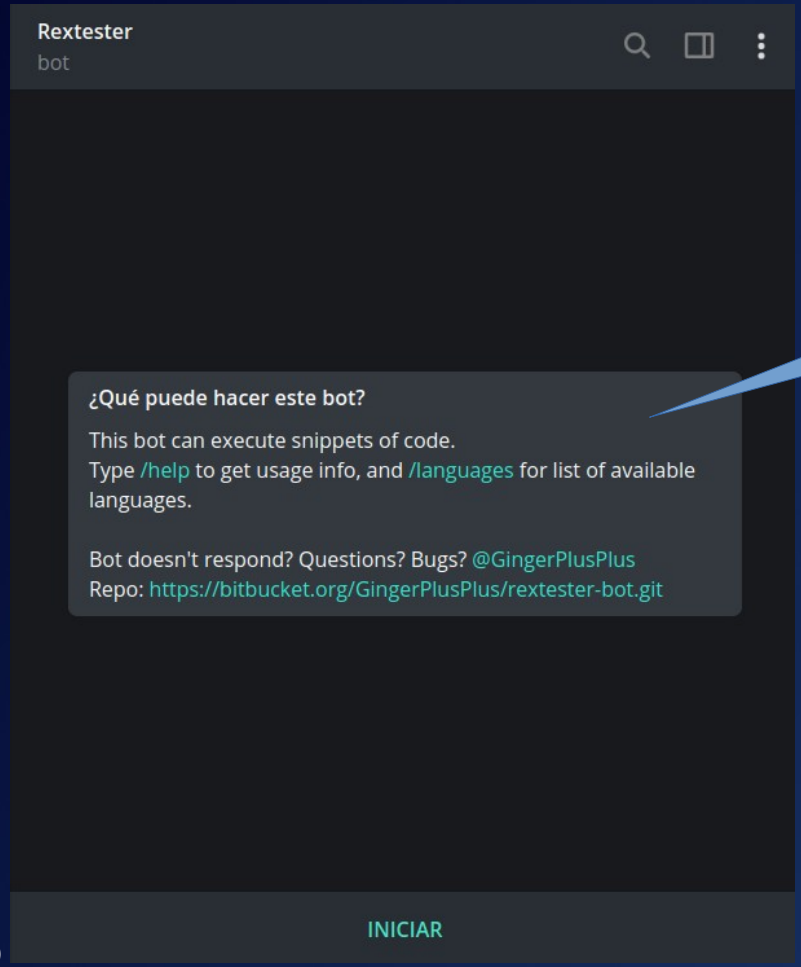
Botpic

Name

About

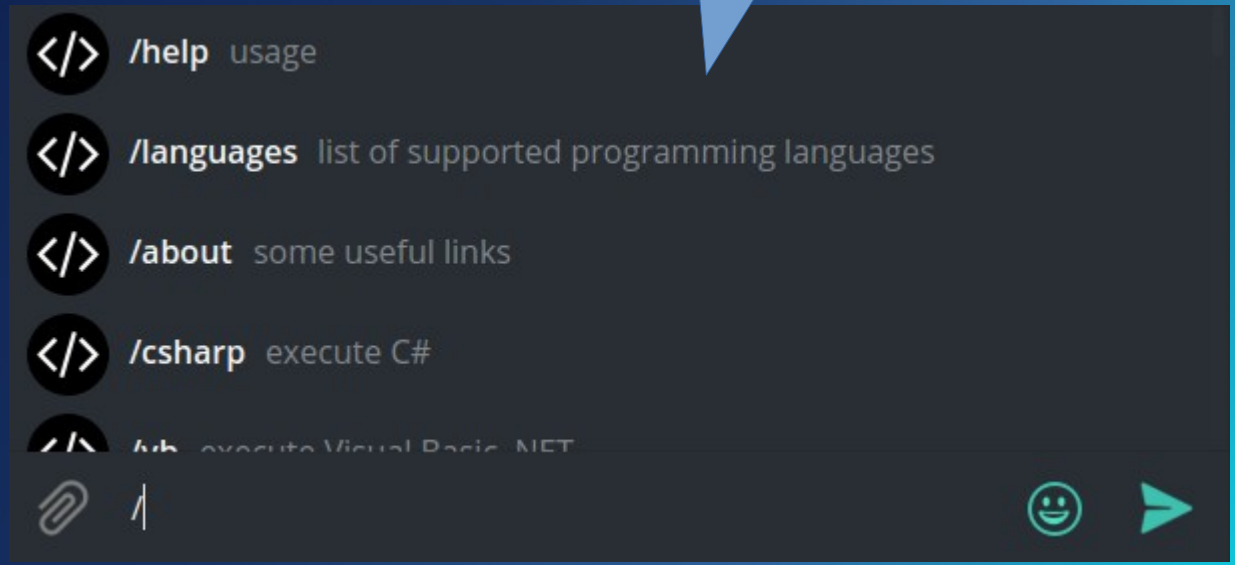
github.com/enforcerzhukov

Personalización



Description

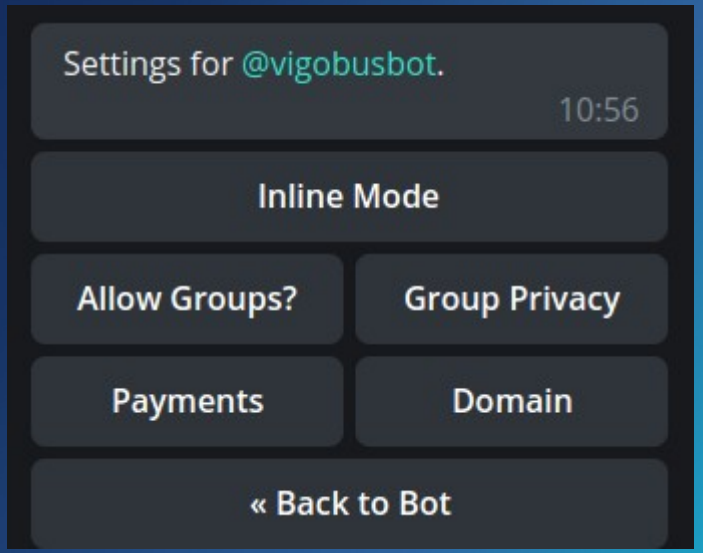
Commands



github.com/enforcerzhukov

Configuraciones del bot

- Inline Mode: habilitar y configurar el modo Inline
- Allow groups?: bloquear el bot en grupos
- Group privacy: obtener acceso a todos los mensajes
- Payments: opciones de pago con el bot
- Domain: configurar el login en páginas web mediante Telegram



github.com/enforcerzhukov