



Bots para Telegram ó como automatizar hasta el infinito y más allá.

(sin morir en el intento)



atareao.es





Introducción



atareao.es





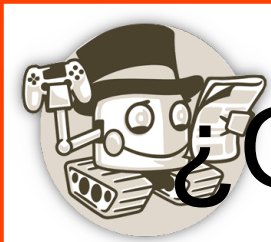
¿Que es un bot?

- Una aplicación informática, un script, una app dentro de otra app
- Una aplicación de terceros que funciona dentro de Telegram



atareao.es





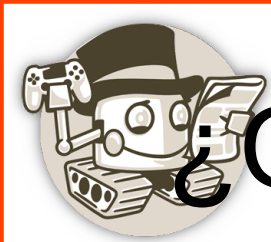
Que podemos hacer con un bot?

- Los usuarios pueden interactuar con los bots
 - Mediante mensajes
 - Comandos y botones
 - Con llamadas en línea



atareao.es





Que podemos hacer con un bot?

- Enviar notificaciones y noticias
- Integrar otros servicios (Gmail bot, Image bot, IMDB bot, Music bot, Wiki bot)
- Herramientas personalizadas (Markdown bot, Stiker bot, Vote bot, Like bot)
- Creación de juegos (GameBot)
- Servicios (HotOrBot)



atareao.es





¿Como funcionan los bots?

- Las peticiones que el usuario envía al bot son procesadas por los servidores de Telegram
- ¿Como nos enteramos de estas peticiones?
Dos opciones excluyentes:
 - Le preguntamos a Telegram: **getUpdates**
 - Telegram nos avisa: **Webhooks**



atareao.es





Crear un bot



atareao.es





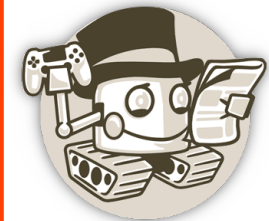
Crear un bot

- Para crear un **bot** utilizaremos un **bot**.
 - **BotFather**
- /newbot
 - name: Linux Center
 - username: linux_center_bot
- https://t.me/linux_center_bot
- Token:
716851483:AAGaXrbms19lhCbaqoR7tdCxCwvUsJtTf7c



atarea0.es





atareao.es



February 21

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

19:47

/newbot 19:47 ✓✓

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

19:48

Linux Center 19:48 ✓✓

linux_center_bot 19:48 ✓✓

Done! Congratulations on your new bot. You will find it at t.me/linux_center_bot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

716851483:AAGaXrbms19lhCbaqoR7tdCxwvUsJtTf7c

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>

19:48



Personarizar nuestro bot

- **/setname** para cambiar el nombre al bot
- **/setdescription** para cambiar la descripción de nuestro bot
- **/setabouttext** para cambiar la información que aparece en el acerca de
- **/setuserpic** para cambiar la imagen de perfil de nuestro bot



atareao.es





Como interactuar con el bot

- Llamadas a la API
 - Agnóstico al lenguaje de programación
- getUpdates
 - Python – python-telegram-bot (framework)
- Webhooks
 - Python
 - PHP



atareao.es





Insomnia



atareao.es





Insomnia

- Un cliente REST API sencillo y potente
- Permite crear llamadas HTTP requests
- Ver la respuesta de forma detallada
- Sobre todo: **ORGANIZACIÓN**



atareao.es





Insomnia

Insomnia

No Environment Cookies

Filter

Telegram

Inicio

GET getMe

GET getUpdates

Mensajes

POST sendMessageV1

POST sendMessageV2

POST sendMessageV3

POST sendMessageV4

POST sendMessageV5

Fotos

POST sendPhotoV1

POST sendPhotoV2

Document

Audio

POST sendAudio

Voice

POST sendVoice

POST

base_url token /sendMessage

Send

200 OK

TIME 97.9 ms

SIZE 464 B

JSON

Auth

Query

Header

Docs

1 {

2 "chat_id": "-311271447",

3 "text": "Esto en negrita y esto

4 en cursiva <a

5 href="https://www.atareao.es">atareao",

6 "parse_mode": "HTML",

7 "disable_web_page_preview": "False",

8 "disable_notification": "False",

9 "reply_to_message_id": 0

10 }

11 {

12 "ok": true,

13 "result": {

14 "message_id": 21,

15 "from": {

16 "id": 716851483,

17 "is_bot": true,

18 "first_name": "Linux Center",

19 "username": "linux_center_bot"

20 },

21 "chat": {

22 "id": -311271447,

23 "title": "Test",

24 "type": "group",

25 "all_members_are_administrators": true

26 },

27 "date": 1550782712,

28 "text": "Esto en negrita y esto en cursiva

29 atareao",

30 "entities": [

31 {

32 "offset": 8,

33 "length": 7,

34 "type": "bold"

35 },

36 {

37 "offset": 26,

38 "length": 7,

39 "type": "italic"

40 },

41 {

42 "offset": 34,

Beautify JSON

\$.store.books[*].author



atareao.es





Llamadas a la API



atareao.es





Llamadas a la API

- <https://core.telegram.org/bots/api#available-methods>
- getMe
 - <https://api.telegram.org/bot<token>/getMe>
- getUpdates
 - <https://api.telegram.org/bot<token>/getUpdates>
 - Nada mas crear no hay nada



atareao.es





Crear un nuevo grupo

- Creamos un nuevo grupo y añadimos a nuestro bot
 - getUpdates se ha actualizado
- Enviemos un mensaje
 - <https://api.telegram.org/bot<token>/sendMessage>
 - “chat_id”:-123456789” / grupospublicos: @nombre
 - el bot tiene que ser miembro del grupo o canal.
 - “text”:"Ejemplo”



atareao.es





Crear un nuevo grupo

- Creamos un nuevo grupo y añadimos a nuestro bot
 - getUpdates se ha actualizado
- Enviemos un mensaje
 - <https://api.telegram.org/bot<token>/sendMessage>
 - “chat_id”:-123456789” / grupospublicos: @nombre
 - el bot tiene que ser miembro del grupo o canal.
 - “text”:"Ejemplo”



atareao.es





Hagamos llamadas a la API

- sendMessage
 - Opciones de formateado
- sendPhoto
- sendAudio
- sendDocument
- SendVoice
- Otros



atareao.es





Hasta 40 llamadas a la API

- getMe
- sendMessage
- Formatting options
- forwardMessage
- sendPhoto
- sendAudio
- sendDocument
- sendVideo
- sendAnimation
- SendVoice
- sendVideoNote
- SendMediaGroup
- sendLocation
- editMessageLiveLocation
- stopMessageLiveLocation
- sendVenue
- sendContact
- sendChatAction
- getUserProfilePhotos
- getFile
- KickChatMember
- unbanChatMember
- restrictChatMember
- promoteChatMember
- ExportChatInviteLink
- setChatPhoto
- deleteChatPhoto
- setChatTitle
- setChatDescription
- pinChatMessage
- unpinChatMessage
- leaveChat
- getChat
- getChatAdministrators
- getChatMembersCount
- getChatMember
- setChatStickerSet
- deleteChatStickerSet
- answerCallbackQuery
- Inline mode methods

<https://core.telegram.org/bots/api#available-methods>



atareao.es





getUpdates

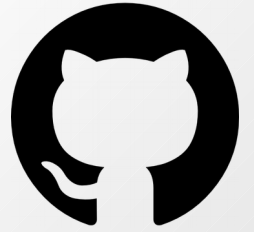


atarea0.es





getUpdates



<https://github.com/python-telegram-bot/python-telegram-bot/>



python-telegram-bot



atareao.es



<https://github.com/atareao/senderbot>





getUpdates

- Tres objetos clave
 - **Bot.** El bot 😊
 - **Updater.** Se encarga de interactuar con el bot
 - **Handler.** Se encarga de interactuar con el *humano*



atareao.es



<https://github.com/atareao/senderbot>





getUpdates

```
179 class SenderBot(Bot):
180     ... def __init__(self, token=None, channel_id=None):
181     ...     ... if token is None or channel_id is None:
182     ...     ...     configuration = Configuration()
183     ...     ...     token = configuration.get('token')
184     ...     ...     channel_id = configuration.get('channel_id')
185     ...     ... Bot.__init__(self, token)
186     ...     ... self.channel_id = channel_id
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
188     def send_message(self, text):
189         super(SenderBot, self).send_message(chat_id=self.channel_id,
190                                             text=text,
191                                             parse_mode=ParseMode.HTML)
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
193 ... def send_image(self, file_image):
194 ...     print(file_image)
195 ...     if os.path.exists(file_image):
196 ...         super(SenderBot, self).sendPhoto(chat_id=self.channel_id,
197 ...         photo=open(file_image, 'rb'))
198 ...     elif file_image.startswith('https') or file_image.startswith('http'):
199 ...         print(1)
200 ...         super(SenderBot, self).sendPhoto(chat_id=self.channel_id,
201 ...         photo=file_image)
```

atareao.es



<https://github.com/atareao/senderbot>

getUpdates

```
203     def send_audio(self, file_audio):
204         if os.path.exists(file_audio):
205             extension = os.path.splitext(file_audio)[1][1:].strip().lower()
206             if extension == 'ogg':
207                 super(SenderBot, self).sendVoice(chat_id=self.channel_id,
208                                                    voice=open(file_audio, 'rb'))
209             elif extension == 'mp3':
210                 super(SenderBot, self).sendAudio(chat_id=self.channel_id,
211                                                    audio=open(file_audio, 'rb'))
212
213     def send_document(self, file_document):
214         if os.path.exists(file_document):
215             super(SenderBot, self).sendDocument(chat_id=self.channel_id,
216                                                  document=open(file_document,
217                                                             'rb'),
218                                                  timeout=999)
```



atareao.es

<https://github.com/atareao/senderbot>



getUpdates

```
220     ... def get_images(self):
221     ...     updates = super(SenderBot, self).get_updates()
222     ...     for index, update in enumerate(updates):
223     ...         if update.message.photo:
224     ...             file_id = update.message.photo[-1].file_id
225     ...             afile = self.get_file(file_id)
226     ...             afile.download()
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

(1)

```
updater = Updater(token=token)
```

(2)

```
def start(bot, update):  
    bot.send_message(chat_id=update.message.chat_id,  
                     text="¿Si?, mi amo!")
```

(3)

```
start_handler = CommandHandler('start', start)  
updater.dispatcher.add_handler(start_handler)
```

(4)

```
updater.start_polling()  
updater.idle()
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

- **Handlers**
 - **CommandHandler.** Comandos
 - **MessageHandler.** Mensajes
 - **ConversationHandler.** Conversaciones
 - **CallbackQueryHandler.** Respuestas



atareao.es



<https://github.com/atareao/senderbot>



getUpdates



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
def caps(bot, update, args):  
    text_caps = ' '.join(args).upper()  
    bot.send_message(chat_id=update.message.chat_id, text=text_caps)
```

```
caps_handler = CommandHandler('caps', caps, pass_args=True)  
dispatcher.add_handler(caps_handler)
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
293 .....def temp(bot, update):  
294 .....    data = open('/sys/class/thermal/thermal_zone0/temp', 'r')  
295 .....    temperatura = data.read()  
296 .....    data.close()  
297 .....    temperatura = temperatura[:-1]  
298 .....    temperatura = str(float(temperatura) / 1000.0) + ' °C'  
299 .....    bot.send_message(chat_id=update.message.chat_id, text=temperatura)
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
301 ..... def samba(bot, update):
302 .....         salida = run_command('systemctl status smbd')
303 .....         if salida is not None:
304 .....             if salida.find('running') > -1:
305 .....                 bot.send_message(chat_id=update.message.chat_id,
306 .....                                     text="Samba está funcionando!")
307 .....                 return
308 .....             elif salida.find('dead') > -1:
309 .....                 bot.send_message(chat_id=update.message.chat_id,
310 .....                                     text="Samba está parado!")
311 .....                 return
312 .....             bot.send_message(chat_id=update.message.chat_id,
313 .....                                 text="Algo no va bien!!! Samba está parado!")
```



atarea.es



<https://github.com/atareao/senderbot>

getUpdates

```
645 .....def take_photo(bot, update):
646 .....    if exists_camera is True:
647 .....        rasp_photo = BytesIO()
648 .....        rasp_photo.name = 'image.png'
649 .....        try:
650 .....            camera = PiCamera()
651 .....            camera.start_preview()
652 .....            # Camera warm-up time
653 .....            sleep(2)
654 .....            camera.capture(rasp_photo, format='png',
655 .....                           use_video_port=True)
656 .....        except Exception as e:
657 .....            print(e)
658 .....        finally:
659 .....            camera.close()
660 .....            rasp_photo.seek(0)
661 .....            bot.send_photo(chat_id=update.message.chat_id,
662 .....                           photo=rasp_photo)
663 .....    else:
664 .....        bot.send_message(chat_id=update.message.chat_id,
665 .....                           text="0 no hay cámara o no la puedo usar!!!")
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

MessageHandler

- Parámetros
 - Filters
 - Filters.text
 - Filters.audio
 - Filters.photo
 - Filters.document
 - Filters.command
 - callback



atareao.es



<https://github.com/atareao/senderbot>





getUpdates

MessageHandler

```
def echo(bot, update):  
    bot.send_message(chat_id=update.message.chat_id,  
                     text=update.message.text)
```

```
echo_handler = MessageHandler(Filters.text, echo)  
dispatcher.add_handler(echo_handler)
```



atareao.es



<https://github.com/atareao/senderbot>





getUpdates

```
def grises(bot, update):  
    photo_file = bot.get_file(update.message.photo[-1].file_id)  
    response = requests.get(photo_file['file_path'])  
    img = Image.open(BytesIO(response.content))  
    img = img.convert('LA')  
    bio = BytesIO()  
    bio.name = 'image.png'  
    img.save(bio, 'PNG')  
    bio.seek(0)  
    bot.send_photo(chat_id=update.message.chat_id,  
                  photo=bio)
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

MessageHandler

```
def tobase64(bot, update):  
    photo_file = bot.get_file(update.message.document.file_id)  
    response = requests.get(photo_file['file_path'])  
    encoded_string = base64.b64encode(response.content).decode()  
    bot.send_message(chat_id=update.message.chat_id,  
                     text=encoded_string)
```



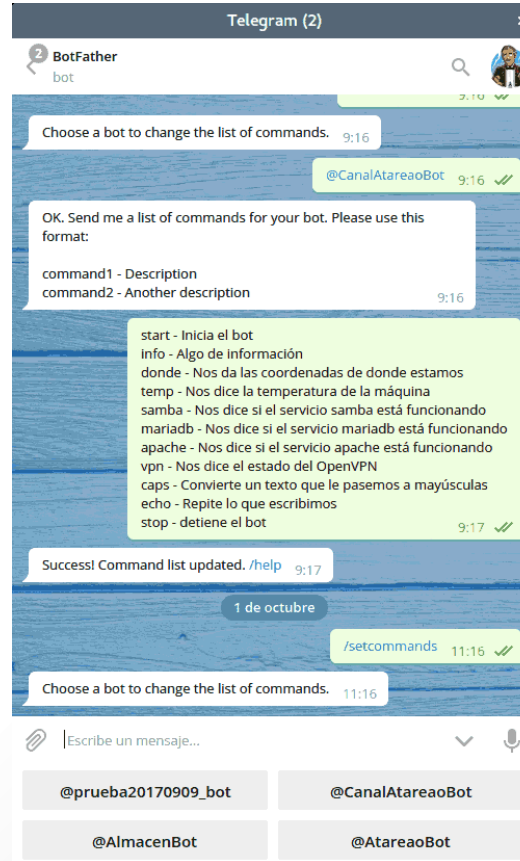
atareao.es



<https://github.com/atareao/senderbot>



getUpdates



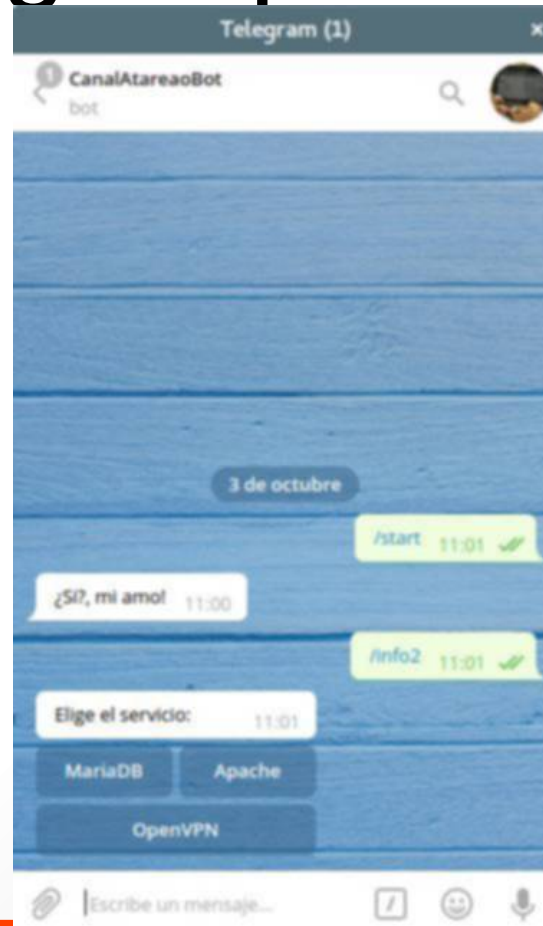
<https://github.com/atareao/senderbot>





getUpdates

CallbackQueryHandler



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
def info2(bot, update):  
    keyboard = [[InlineKeyboardButton("MariaDB", callback_data='1')  
                 InlineKeyboardButton("Apache", callback_data='2')]  
                [InlineKeyboardButton("OpenVPN", callback_data='3')]  
    reply_markup = InlineKeyboardMarkup(keyboard)  
    update.message.reply_text('Elige el servicio:',  
                              reply_markup=reply_markup)
```

```
info2_handler = CommandHandler('info2', info2)  
dispatcher.add_handler(info2_handler)
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

```
def button(bot, update):  
    query = update.callback_query  
    logger.info('TELEGRAM query data: "%s" type: "%s"' % (  
        str(query.data), str(type(query.data))))  
    if query.data == '1':  
        salida = run_command('systemctl status mariadb')  
        if salida is not None:  
            if salida.find('running') > -1:  
                msg = 'MariaDB está funcionando!'  
            elif salida.find('dead') > -1:  
                msg = 'MariaDB está parado!'  
            else:  
                msg = 'Algo no va bien! MariaDB está parado!'  
        else:  
            msg = 'Algo no va bien! MariaDB está parado!'  
        bot.edit_message_text(text=msg,  
                               chat_id=query.message.chat_id,  
                               message_id=query.message.message_id)
```

```
dispatcher.add_handler(CallbackQueryHandler(button))
```



atareao.es



<https://github.com/atareao/senderbot>



getUpdates

CallbackQueryHandler

```
dispatcher.add_handler(CallbackQueryHandler(button))
```



atareao.es



<https://github.com/atareao/senderbot>





Webhooks



atareao.es





Webhooks

- Necesitas un dominio (o ddns)
- Puerto abierto
 - Telegram solo trabaja con 443, 80, 88, 8443
- Cifrado SSL/TTLS (puede ser autofirmado, también Let's Encrypt)



atareao.es



<https://github.com/atareao/senderbot>





Webhooks

- Métodos para trabajar con tu webhook
 - setWebhook
 - deleteWebhook
 - getWebhookInfo



atareao.es



<https://github.com/atareao/senderbot>





Webhooks

Insomnia - getWebhookInfo

Insomnia GET base_uri token /getWebhookInfo Send 200 OK TIME 255 ms SIZE 87 B

No Environment Cookies Filter

Telegram

- Webhooks
 - GET setWebhook
 - GET deleteWebhook
 - GET getWebhookInfo
- Inicio
 - GET getMe
 - GET getUpdates
- Mensajes
 - POST sendMessageV1

Body Auth Query Header Docs

URL PREVIEW

https://api.telegram.org/bot716851483:AAGaXrbms19lhCbaqrR7tdCxQwvUsJtTf7c/getWebhookInfo

New name New value

Import from URL

Preview Header 9 Cookie Timeline

```
1 {
2   "ok": true,
3   "result": {
4     "url": "",
5     "has_custom_certificate": false,
6     "pending_update_count": 5
7   }
8 }
```

\$.store.books[*].author



atareao.es



<https://github.com/atareao/senderbot>





Python



atarea0.es





Webhooks - Python

```
2 import requests
3
4 class Telegram():
5     def __init__(self, token):
6         self.token = token
7
8     def send_simple_message(self, chat_id, message):
9         url = 'https://api.telegram.org/bot{0}/{1}'.format(self.token, 'sendMessage')
10        payload = {'chat_id': chat_id,
11                  'text': message,
12                  'parse_mode': 'HTML'}
13        r = requests.post(url, data=payload)
14
15    def kick_member(self, chat_id, user_id):
16        url = 'https://api.telegram.org/bot{0}/{1}'.format(self.token, 'kickChatMember')
17        payload = {'chat_id': chat_id,
18                  'user_id': user_id}
19        r = requests.post(url, data=payload)
```



atareao.es





Webhooks - Python

```
1 from flask import Flask, request, abort
2 from dobot.telegramapi import Telegram
3 from dobot.database import Database
4 app = Flask(__name__)
5
6 @app.route('/c8f4c132-e5fb-4d0d-b4ce-d7dd501ecd85', methods=['POST', 'GET'])
7 def webhook():
```



atareao.es





Webhooks - Python

```
7 def webhook():
8     ... if request.method == 'POST':
9     ...     update = request.json
10    ...     print(update)
11    ...     if 'message' in update and 'new_chat_members' in update['message']:
12    ...         database = Database()
13    ...         for member in update['message']['new_chat_members']:
14    ...             exists = database.exists_user(member['id'])
15    ...             print('exists', exists)
16    ...             telegram = Telegram(app.token)
17    ...             if not exists:
18    ...                 if member['is_bot'] is True:
19    ...                     telegram.kick_member(update['message']['chat']['id'], member['id'])
20    ...                     msg = 'Expulsado {0}. No se admiten otros bots que no sea yo mismo.'
21    ...                     telegram.send_simple_message(update['message']['chat']['id'], msg)
22    ...                 else:
23    ...                     database.insert_user(member)
24    ...                     # msg = 'Hola <strong>{0}</strong>, bienvenido a ¡GNU/Linux Valencia
25    ...                     msg = 'Hola <strong>{0}</strong>, bienvenido a ¡Ubuntizados!'.format
26    ...                     msg += 'Por favor, <strong>revisa las normas</strong> que se encuent
27    ...                     # msg += 'No te olvides visitar nuestra web <a href="https://gnulinu
28    ...                     msg += 'No te olvides visitar la web <a href="https://www.atareao.es
29    ...                     telegram.send_simple_message(update['message']['chat']['id'], msg)
30    ...                 else:
31    ...                     msg = 'Hola de nuevo <strong>{0}</strong>'.format(member['first_name'])
32    ...                     telegram.send_simple_message(update['message']['chat']['id'], msg)
```



atareao.es





Webhooks - Python

```
33 ..... elif 'message' in update and 'from' in update['message']:  
34 .....     database = Database()  
35 .....     exists = database.exists_user(update['message']['from']['id'])  
36 .....     if not exists:  
37 .....         database.insert_user(update['message']['from'])  
38 ..... elif 'edited_message' in update and 'from' in update['edited_message']:  
39 .....     database = Database()  
40 .....     exists = database.exists_user(update['edited_message']['from']['id'])  
41 .....     if not exists:  
42 .....         database.insert_user(update['edited_message']['from'])
```



atareao.es





PHP



atarea0.es





Webhooks - PHP

2 funciones clave para las llamadas a la API

```
18 function http_get($url)
19 {
20     ...$ans = null;
21     ...$ch = curl_init();
22     ...curl_setopt($ch, CURLOPT_URL, $url);
23     ...try
24     ...{
25         ...//Disable SSL verification
26         ...curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
27         ...//Will return the response, if false it print the response
28         ...curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
29         ...$ans = json_decode(curl_exec($ch));
30         ...if($ans->ok !== TRUE)
31         ...{
32             ...$ans = null;
33         ...}
34     ...}
35     ...catch(Exception $e)
36     ...{
37         ...echo "Error: ", $e->getMessage(), "\n";
38     ...}
39     ...curl_close($ch);
40     ...return json_encode($ans);
41 }
```



atareao.es





Webhooks - PHP

```
43 function http_post($url, $json)
44 {
45     $ans = null;
46     $ch = curl_init();
47     curl_setopt($ch, CURLOPT_URL, $url);
48     try
49     {
50         $data_string = json_encode($json);
51         //Disable SSL verification
52         curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
53         curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
54         curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
55         //Will return the response, if false it print the response
56         curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
57         curl_setopt($ch, CURLOPT_HTTPHEADER, array(
58             'Content-Type: application/json',
59             'Content-Length: ' . strlen($data_string)
60         ));
61         $ans = json_decode(curl_exec($ch));
62         if($ans->ok != TRUE)
63         {
64             $ans = null;
65         }
66     }
67     catch(Exception $e)
68     {
69         echo "Error: ", $e->getMessage(), "\n";
70     }
71     curl_close($ch);
72     return $ans;
73 }
```



atareao.es





Webhooks - PHP

El web hook
(integrado con WordPress)

```
2  if(function_exists('add_action'))
3  {
4      ...function my_awesome_func( $data ) {
5          ...global $URL;
6          ...return [ 'url' => $URL,
7                      ...'ok' => TRUE ];
8      }
9
10     ...add_action( 'rest_api_init', function() {
11         ...register_rest_route( 'b1abc5d8-49ag-66ee-vd78-4dfsrgvadfss/v1', '/tb/', array(
12             ...'methods' => 'GET',
13             ...'callback' => 'my_awesome_func',
14             ...));
15     ...});
16 }
```



atareao.es



Webhooks - PHP

```
75 function getMe($url)
76 {
77     return http_get($url.'/getMe');
78 }
79
80 function sendMessage($url, $chat_id, $text)
81 {
82     $json = ['chat_id' => $chat_id,
83             'text' => $text,
84             'parse_mode' => 'HTML'];
85     return http_post($url.'/sendMessage', $json);
86 }
87
88 function sendPhoto($url, $chat_id, $image_url, $caption)
89 {
90     $json = ['chat_id' => $chat_id,
91             'photo' => $image_url,
92             'caption' => $caption,
93             'parse_mode' => 'HTML'];
94     return http_post($url.'/sendPhoto', $json);
95 }
96
97 function getLastChatId($url)
98 {
99     return http_get($url.'/getUpdates');
100 }
```





Recursos



atareao.es





Recursos



- <https://www.atareao.es/tutorial/crea-tu-propio-bot-para-telegram/>
- <https://github.com/atareao/senderbot>
- <https://github.com/python-telegram-bot/python-telegram-bot/>



atareao.es



<https://github.com/atareao/senderbot>





Agradecimientos



atareao.es





Agradecimientos

SLIMBOOK



atareao.es

