

Fuentes:

<https://atareao.es/tutorial/crea-tu-propio-bot-para-telegram/un-bot-de-telegram-con-php/>

Crear el bot

Buscamos @botfather en Telegram

Escribimos

```
/newbot
```

Nos contesta que ahora le tenemos que poner un nombre. El nombre puede ser largo y tener espacios:

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

Ponemos el nombre:

```
Bot de pruebas
```

Nos contesta:

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

El username no puede tener espacios y tiene que acabar con la palabra **bot**: Le ponemos:

```
jurjurbot
```

Nos dice que está creado y nos da el token del bot que debemos guardar.

```
Done! Congratulations on your new bot. You will find it at t.me/jurjurbot. 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:  
2144753637:AAHX3BhhasddifNTDbJSvhLLYp-dGvnTxmPtk  
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
```

Metemos usuario en un grupo. Escribimos /start en el grupo. y sacamos el nombre del grupo con el comando:

```
curl -X GET https://api.telegram.org/bot<token>/getUpdates|jq
```

Por ejemplo:

```
curl -X GET  
https://api.telegram.org/bot2345053637:AAHX3BhheADTFDTDbJSvhLLYp-dGvnTxmPtk/  
getUpdates|jq
```

Nos devuelve:

```
"update_id": 151787201,  
"message": {  
  "message_id": 11,  
  "from": {  
    "id": 123123123,  
    "is_bot": false,  
    "first_name": "Mi usuario telegram",  
    "username": "UsuarioTelegram",  
    "language_code": "en"  
  },  
  "chat": {  
    "id": -648234144,  
    "title": "Nombre del grupo",  
    "type": "group",  
    "all_members_are_administrators": true  
  },  
}
```

El ID del grupo es:

```
-648234144
```

Puedo enviar un mensaje con el comando:

```
curl -s -X POST https://api.telegram.org/bot<TOKEN>/sendMessage -d  
chat_id=<CHAT_ID> -d text="prueba"
```

Por ejemplo:

```
curl -s -X POST  
https://api.telegram.org/bot2143367637:AAHX3BhgThifNTDbJSvhLLYp-dGvnTxmPtk/s  
endMessage -d chat_id=-648234144 -d text="prueba"
```

Podemos hacer el script:

```
#!/bin/bash  
TOKEN="2143367637:AAHX3BhgThifNTDbJSvhLLYp-dGvnTxmPtk"  
CHAT_ID="-648234144"  
  
URL="https://api.telegram.org/bot$TOKEN/sendMessage"
```

```
curl -s -X POST $URL -d chat_id=$CHAT_ID -d text="$1"
```

Interactuar BOT

Podemos ver las actualizaciones del bot con la API getUpdates como hemos visto antes:

```
curl -X GET https://api.telegram.org/bot<token>/getUpdates | jq
```

Pero vamos a añadir un webhook para que que haga PUSH en vez de PULL, así cada vez que escribamos algo, lo leerá y no tenemos que estar constantemente leyendo los Updates.

El webhook tiene que apuntar a un fichero que esté en un servidor con https. En este caso usaremos php. Para añadir el webhook:

```
curl -X GET https://api.telegram.org/bot<TOKEN>/setWebhook?url=https://<MI WEB CON API>/<FICHERO>
```

Por ejemplo:

```
curl -X GET https://api.telegram.org/bot2115178200:AAElT1wG2T18bf8cZMo_X_LggLfAcIOaiHY/setWebhook?url=https://api.midominio.com/bot.php
```

Nos responderá esto:

```
{
  "ok": true,
  "result": true,
  "description": "Webhook was set"
}
```

Para comprobar que esté bien lanzamos este comando. Suele fallar por el certificado.

```
curl -X GET https://api.telegram.org/bot2115178200:AAElT1wG2T18ZMo_X_LggLfAcIOaiHY/getWebhookInfo
```

Si está bien, aunque el fichero esté vacío, luego lo creamos bien, dará esto:

```
{
  "ok": true,
  "result": {
    "url": "https://api.midominio.com/bot.php",
    "has_custom_certificate": false,
    "pending_update_count": 0,
    "max_connections": 40,
    "ip_address": "87.217.220.23"
  }
}
```

```
}
```

Creamos el fichero bot.php así. Cambiamos el TOKEN por el nuestro:

```
<?php
define('BOT_TOKEN', '2143053637:AAHX3BasdasdDbJSvhLlYp-dGvnTxmPtk');
define('API_URL', 'https://api.telegram.org/bot'.BOT_TOKEN.'/');

// read incoming info and grab the chatID
$content      = file_get_contents("php://input");
$update       = json_decode($content, true);
$chatID       = $update["message"]["chat"]["id"];
$message      = $update["message"]["text"];

// compose reply
$reply = "";
switch ($message) {
    case "/start":
        $reply = urlencode("Bienvenido al mi bot.\nEscribe /ayuda para ver los comandos posibles");
        break;
    case "/ayuda":
        $reply = urlencode("comandos:\n/consulta1 Lee el fichero fichero1.txt\n/consulta2 Lee el fichero fichero1.txt\n/ayuda Esta ayuda");
        break;
    case "/consulta1":
        $reply = "Fichero 1: ".trim(file_get_contents("https://api.midominio.com/fichero1.txt"), "\r\n");
        break;
    case "/consulta2":
        $reply = "Fichero 2: ".trim(file_get_contents("https://api.midominio.com/fichero2.txt"), "\r\n");
        break;
    default:
        $reply = "Comando no aceptado. Prueba con /ayuda";
}

// send reply
$sendto = API_URL."sendmessage?chat_id=".$chatID."&text=".$reply;
file_get_contents($sendto);

// Create a debug log.txt to check the response/repy from Telegram in JSON format.
// You can disable it by commenting checkJSON.
checkJSON($chatID,$update);
function checkJSON($chatID,$update){

    $myFile = "log.txt";
    $updateArray = print_r($update,TRUE);
    $fh = fopen($myFile, 'a') or die("can't open file");
    fwrite($fh, $chatID ."nn");
}
```

```
fwrite($fh, $updateArray."nn");  
fclose($fh);  
}
```

Cositas de PHP:

- Ponemos urlencode para poder poner \n y escribir un salto de línea
- Usamos trim para quitar una _ que aparece al leer una url como si fuera un salto de línea

```
trim(file_get_contents("https://api.midominio.com/fichero.txt"), "\r\n")
```

Menus con botfather

Desde el bot @botfather podemos añadir los menús para que aparezcan los comandos.

API en PHP

El webhook de telegram lanza peticiones a la API de PHP de este tipo con todas las acciones que pasan. Metiendo el bot @RawDataBot se pueden sacar de cada grupo. Va bien para saber el chatID por ejemplo:

```
{  
  "update_id": 816025253,  
  "message": {  
    "message_id": 974009,  
    "from": {  
      "id": 4982505,  
      "is_bot": false,  
      "first_name": "I Want to be Freak",  
      "username": "IWantToBeFreak"  
    },  
    "chat": {  
      "id": -699088581,  
      "title": "Jur123 group",  
      "type": "group",  
      "all_members_are_administrators": true  
    },  
    "date": 1638376574,  
    "new_chat_participant": {  
      "id": 211246197,  
      "is_bot": true,  
      "first_name": "Telegram Bot Raw",  
      "username": "RawDataBot"  
    },  
    "new_chat_member": {  
      "id": 211246197,
```

```

        "is_bot": true,
        "first_name": "Telegram Bot Raw",
        "username": "RawDataBot"
    },
    "new_chat_members": [
        {
            "id": 211246197,
            "is_bot": true,
            "first_name": "Telegram Bot Raw",
            "username": "RawDataBot"
        }
    ]
}

```

A nosotros solo nos interesa uno como este

```

{
  "message": {
    "chat": {
      "id": -699088581
    },
    "text": "/consulta2"
  }
}

```

Si lo lanzamos a la API de PHP haría el comando **/consulta2** El chat id tiene que ser el de la API

```
curl -X POST -d@peticion.json "https://api.midominio.com/bot.php"
```

ERRORES

El bot no da respuesta y en nginx da 499

```
curl -s -X GET
https://api.telegram.org/bot2143053637:AAHX3BhheQifNTDbJSvhLLYp-dGvnTxmPtk/g
etWebhookInfo |jq
```

Da error "Read timeout expired",

```

{
  "ok": true,
  "result": {
    "url": "https://apitelegam.lope99.info/bot.php",
    "has_custom_certificate": false,
    "pending_update_count": 37,
    "last_error_date": 1639677248,
    "last_error_message": "Read timeout expired",

```

```
"max_connections": 40,  
"ip_address": "87.217.220.23"  
}  
}
```

Era un problema de DNS pero para vaciar los mensajes pendientes he puesto en el fichero bot.php

```
$update = json_decode(file_get_contents('php://input'));  
if(isset($update->message) || isset($update->edited_message)) {  
    if(time() -  
    (($update->message->date)? : ($update->edited_message->date)) > 59) {  
        exit('Update Time Out !'); # print json update  
    }  
}
```

From:

<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:

<http://wiki.legido.com/doku.php?id=informatica:telegram:bot>



Last update: **2023/03/20 14:13**