
Framebot Documentation

Versão 0.1.0

Giancarlo and Ródgger

06 jun, 2018

1	Começando	3
1.1	Instalação	3
1.2	Configuração	4
1.3	Utilização	4
1.4	Exemplos	4
2	Tutorial	5
2.1	Estrutura básica	5
2.2	Variáveis	6
2.3	Pegando atualizações	6
2.4	Respondendo as mensagens	6
2.5	Resetando updates	7
2.6	Limpando estruturas	7
3	Objetos	9
3.1	Objetos exclusivos	9
3.2	Referência completa	9
4	Funções	19
4.1	Referencia completa	19

Framebot é um framework escrito C com o objetivo de ajudar programadores a criarem chat bots para o [Telegram](#).

Framebot é uma biblioteca feita em C para facilitar o uso da BOT API do telegram.

1.1 Instalação

Para instalar a biblioteca você deve primeiro clonar o nosso repositório:

```
git clone https://github.com/giancarlopro/framebot
```

Agora você vai precisar ter o *cmake* instalado, e algumas libs:

```
sudo apt install cmake openssl curl jansson
```

Após isso entre na pasta onde a biblioteca foi clonada, por padrão vai ser *framebot*:

```
cd framebot
```

Agora crie uma pasta para compilar a biblioteca e rode o *cmake* para criar o *makefile*:

```
mkdir build  
cd build  
cmake ..`
```

Agora utilize o *makefile* para compilar e instalar a biblioteca:

```
make  
make install
```

1.2 Configuração

Para utilizar a biblioteca em seus projetos, você deve informar ao compilador explicitamente as bibliotecas que serão compiladas juntas.

- Se estiver compilando pelo terminal:

```
gcc projeto.c -o out -lcurl -ljansson -lframebot
```

- Se estiver utilizando o cmake basta informar no seu CMakeLists.txt, em target_link_libraries:

```
target_link_libraries(seuexecutavel framebot)
```

1.3 Utilização

Para utilizar a biblioteca em seus projetos, você deve incluir o arquivo *framebot.h*.

```
#include <framebot/framebot.h>
```

1.4 Exemplos

Você encontra exemplos de uso da biblioteca na pasta *examples* em nosso repositório no [GitHub](#).

Neste pequeno tutorial, vamos criar um echo bot. Ele basicamente vai reenviar as mensagens que são recebidas como resposta.

O código está disponível em [GitHub: Echo Bot](#).

2.1 Estrutura básica

Para este projeto, nós vamos receber o *token* do bot na chama do binário, ficando assim:

```
./bot TOKEN
```

Para isso vamos criar nossa estrutura em c da seguinte forma:

```
#include <stdio.h>
#include <framebot/framebot.h>

int main (int argc, char **argv) {
    if (argc < 2) {
        printf("Usage: ./echo BOT_TOKEN\n");
        exit(-1);
    }

    framebot_init(); // Inicializamos

    return 0;
}
```

Aqui nós criamos uma condição caso o *token* não seja fornecido, e inicializamos a biblioteca com a função `framebot_init()`.

2.2 Variáveis

Agora precisaremos de algumas variáveis para manipular a biblioteca. Serão elas:

```
long int last_offset = 0;
Bot *echo;
Framebot *frame_update;
Update *message, *temp;
```

Onde: `last_offset` vai ser utilizado para podermos resetar a lista de updates mais tarde

`echo` é o nosso objeto do tipo `Bot`, essencial para manipular a biblioteca

`frame_update` que vai ser inicializado pela função `get_updates()` com as últimas mensagens

`message` que vai conter nosso array de mensagens

Agora vamos inicializar nosso objeto `Bot` para podermos usar a API do Telegram. Para isso utilizamos a função `framebot()`, caso aconteça algum erro, ela retornará `NULL`. Vamos passar para a função nosso token que está localizado em `argv[1]`.

```
echo = framebot(argv[1]);
```

2.3 Pegando atualizações

Agora já podemos pegar as atualizações direto da API.

```
frame_update = get_updates(echo, NULL, last_offset, 100, 0, NULL);
message = frame_update->update_message;
temp = message;
```

Na primeira linha é retornado um objeto do tipo `Framebot`, que contém referências para as atualizações de diferentes tipos. Na segunda linha nós armazenamos na variável `message` os updates do tipo `Message`. Na última linha nós passamos a referência do primeiro `Update` para a variável `temp`, que nós vamos utilizar para iterar os `Updates`.

2.4 Respondendo as mensagens

Com as atualizações da API em mãos, agora vamos responder as mensagens recebidas pelo bot. Primeiro vamos iterar nossos `Updates`, e responder com a mesma mensagem recebida.

```
while (temp) {
    if (temp->message) {
        if (send_message_chat (echo, temp->message->from->id, temp->message->text, NULL,
↪ 0, 0, temp->message->message_id, NULL)) {
            printf("Sendado: \"%s\" to %s!\n", temp->message->text, temp->message->from->
↪ username);
        }
    }

    last_offset = temp->update_id + 1; // Aqui armazenamos o offset do ultimo update,
↪ para podermos resetar depois
    temp = temp->next;
}
```

Aqui temos um `while` para iterar nossos `Updates`. Primeiro verificamos se recebemos de fato alguma mensagem, depois tentamos enviar nossa resposta, caso consigamos é escrito na tela uma mensagem informando o que foi enviado e para quem. Depois atualizamos nosso `last_offset` para podermos resetar a lista de updates.

2.5 Resetando updates

Resetar a lista de updates é importante para que não recebamos novamente as mesmas atualizações, dessa forma só respondemos uma única vez. Para resetar, nós fazemos mais uma chamada à `get_updates()` e passamos o nosso `last_offset`. Dessa forma, na próxima chamada à `get_updates()` não iremos receber os mesmos `Updates`.

```
get_updates(echo, frame_update, last_offset, 100, 0, NULL);
```

2.6 Limpando estruturas

No final nós liberamos a memória alocada com a função `framebot_free()`.

```
framebot_free(frame_update);
```

Aqui finalizamos nosso tutorial.

Código final disponível em [GitHub: Echo Bot](#).

Nossa biblioteca implementa todos os objetos da API do telegram, com os mesmos nomes, e a princípio com os mesmos membros. Alguns tipos podem possuir um ponteiro `next` apontando para o próximo objeto.

3.1 Objetos exclusivos

Objetos que existem somente na biblioteca, e não possuem representação na API do Telegram.

Bot

Objeto que vai ser utilizado para referenciar seu bot nas funções da API.

char ***token**
Seu *token* do Telegram.

User ***user**
Objeto retornado da chamada à `get_me()`

3.2 Referência completa

User

```
long int id
bool is_bot
char *first_name
char *last_name
char *username
char *language_code
struct _user *next
```

Bot

char ***token**

User ***user**

ChatPhoto

char ***small_file_id**

char ***big_file_id**

Chat

long int **id**

char ***type**

char ***title**

char ***username**

char ***first_name**

char ***last_name**

bool **all_members_are_administrators**

ChatPhoto ***photo**

char ***description**

char ***invite_link**

struct _message ***pinned_message**

char ***sticker_set_name**

bool **can_set_sticker_set**

MessageEntity

char ***type**

long int **offset**

long int **length**

char ***url**

User ***user**

struct _message_entity ***next**

Audio

char ***file_id**

long int **duration**

char ***performer**

char ***title**

char ***mime_type**

long int **file_size**

PhotoSize

char ***file_id**

int **width**

int **height**

long int **file_size**

struct _photo_size ***next**

Game

char ***file_id**

PhotoSize ***thumb**

char ***file_name**

char ***mime_type**

long int **file_size**

char ***title**

char ***description**

char ***text**

Animation ***animation**

Sticker

char ***file_id**

int **width**

int **height**

PhotoSize ***thumb**

char ***emoji**

long int **file_size**

Video

char ***file_id**

int **width**

int **height**

long int **duration**

PhotoSize ***thumb**

char ***mime_type**

long int **file_size**

VideoNote

char ***file_id**
long **length**
long **duration**
PhotoSize ***thumb**
long **file_size**

Voice

char ***file_id**
long int **duration**
char ***mime_type**
long int **file_size**

Contact

char ***phone_number**
char ***first_name**
char ***last_name**
long int **user_id**

Location

float **latitude**
float **longitude**

Venue

Location ***location**
char ***title**
char ***address**
char ***foursquare_id**

Invoice

char ***title**
char ***description**
char ***start_parameter**
char ***currency**
long **total_amount**

ShippingAddress

char ***country_code**
char ***state**


```
char *city
char *street_line1
char *street_line2
char *post_code
```

OrderInfo

```
char *name
char *phone_number
char *email
ShippingAddress *shipping_address
```

SuccessfulPayment

```
char *currency
long total_amount
char *invoice_payload
char *shipping_option_id
OrderInfo *order_info
char *telegram_payment_charge_id
char *provider_payment_charge_id
```

ShippingQuery

```
char *id
User *from
char *invoice_payload
ShippingAddress *shipping_address
```

PreCheckoutQuery

```
char *id
User *from
char *currency
long total_amount
char *invoice_payload
char *shipping_option_id
OrderInfo *order_info
```

Message

```
long int message_id
User *from
```

long int **date**
Chat ***chat**
User ***forward_from**
Chat ***forward_from_chat**
long int **forward_from_message_id**
char ***forward_signature**
long int **forward_date**
struct _message ***reply_to_message**
long int **edit_date**
char ***media_group_id**
char ***author_signature**
char ***text**
MessageEntity ***caption_entities**
Audio ***audio**
Document ***document**
Game ***game**
Sticker ***sticker**
Video ***video**
Voice ***voice**
VideoNote ***video_note**
char ***caption**
Contact ***contact**
Location ***location**
Venue ***venue**
User ***new_chat_members**
User ***left_chat_member**
char ***new_chat_title**
bool **delete_chat_photo**
bool **group_chat_created**
bool **supergroup_chat_created**
bool **channel_chat_created**
long int **migrate_to_chat_id**
long int **migrate_from_chat_id**
struct _message ***pinned_message**
Invoice ***invoice**
SuccessfulPayment ***successful_payment**

char *connected_website

InlineQuery

char *id

User *from

Location *location

char *query

char *offset

ChosenInlineResult

char *result_id

User *from

Location *location

char *inline_message_id

char *query

CallbackQuery

char *id

User *from

Message *message

char *inline_message_id

char *chat_instance

char *data

char *game_short_name

Framebot

long int update_id

Message *message

Message *edited_message

Message *channel_post

Message *edited_channel_post

InlineQuery *inline_query

ChosenInlineResult *chosen_inline_result

CallbackQuery *callback_query

ShippingQuery *shipping_query

PreCheckoutQuery *pre_checkout_query

struct_update *next

long int **update_id**
UPMessage ***message**
UPMessage ***edited_message**
UPMessage ***channel_post**
UPMessage ***edited_channel_post**
UPIResult ***chosen_inline_result**
UPPCQuery ***pre_checkout_query**

ChatMember

User ***user**
char ***status**
long int **until_date**
bool **can_be_edited**
bool **can_change_info**
bool **can_post_messages**
bool **can_edit_messages**
bool **can_delete_messages**
bool **can_invite_users**
bool **can_restrict_members**
bool **can_pin_messages**
bool **can_promote_members**
bool **can_send_messages**
bool **can_send_media_messages**
bool **can_send_other_messages**
bool **can_add_web_page_previews**
struct_chat_member ***next**

File

char ***file_id**
long **file_size**
char ***file_path**

UserProfilePhotos

long **total_count**

CallbackGame

long int **user_id**

```
long int score
bool force
bool disable_edit_message
long int chat_id
long int message_id
char *inline_message_id
```

KeyboardButton

```
char *text
bool request_contact
bool request_location
struct _keyboard_button *next
```

InlineKeyboardButton

```
char *text
char *url
char *callback_data
char *switch_inline_query
char *switch_inline_query_current_chat
CallbackGame *callback_game
bool pay
```

InlineKeyboardMarkup

```
int type
```

ReplyKeyboardMarkup

```
int type
bool resize_keyboard
bool one_time_keyboard
bool selective
```

ReplyKeyboardRemove

```
int type
bool remove_keyboard
bool selective
```

ForceReply

```
int type
```

bool **force_reply**

bool **selective**

Keyboard

int **type**

InlineKeyboardButton ***inline_keyboard_markup**

ReplyKeyboardMarkup ***reply_keyboard_markup**

ReplyKeyboardRemove ***reply_keyboard_remove**

ForceReply ***_force_reply**

Aqui você encontra as funções implementadas pela biblioteca.

Todas as funções disponíveis na API do Telegram, estão disponíveis na biblioteca usando o formato lower case separando os nomes com underscore.

4.1 Referencia completa

User ***user** (long int *id*, bool *is_bot*, const char **first_name*, const char **last_name*, const char **username*, const char **language_code*)

void **user_add** (*User* **origin*, *User* **next*)

void **user_free** (*User* **usr*)

void **chat_free** (*Chat* **cht*)

void **message_entity_free** (*MessageEntity* **msgett*)

void **message_entity_add** (*MessageEntity* **dest*, *MessageEntity* **src*)

size_t **message_entity_len** (*MessageEntity* **message_entity*)

MessageEntity ***message_entity_get** (*MessageEntity* **message_entity*, int *index*)

Audio ***audio** (const char **file_id*, long int *duration*, const char **performer*, const char **title*, const char **mime_type*, long int *file_size*)

void **audio_free** (*Audio* **audio*)

void **photo_size_free** (*PhotoSize* **photoSize*)

void **photo_size_add** (*PhotoSize* **root*, *PhotoSize* **newps*)

PhotoSize ***photo_size_get** (*PhotoSize* **root*, int *i*)

size_t **photo_size_len** (*PhotoSize* **ophoto_sise*)

Document ***document** (const char **file_id*, *PhotoSize* **thumb*, const char **file_name*, const char **mime_type*, long int *file_size*)

`void document_free` (`Document *document`)

`Animation *animation` (`const char *file_id`, `PhotoSize *thumb`, `const char *file_name`, `const char *mime_type`, `long int file_size`)

`void animation_free` (`Animation *animation`)

`Game *game` (`const char *title`, `const char *description`, `PhotoSize *photo`, `const char *text`, `MessageEntity *text_entities`, `Animation *animation`)

`void game_free` (`Game *game`)

`Sticker *sticker` (`const char *file_id`, `int width`, `int height`, `PhotoSize *thumb`, `const char *emoji`, `long int file_size`)

`void sticker_free` (`Sticker *_sticker`)

`Video *video` (`const char *file_id`, `int width`, `int height`, `long int duration`, `PhotoSize *thumb`, `const char *mime_type`, `long int file_size`)

`void video_free` (`Video *_video`)

`Voice *voice` (`const char *file_id`, `long int duration`, `const char *mime_type`, `long int file_size`)

`void voice_free` (`Voice *_voice`)

`Contact *contact` (`const char *phone_number`, `const char *first_name`, `const char *last_name`, `long int user_id`)

`void contact_free` (`Contact *_contact`)

`Location *location` (`float latitude`, `float longitude`)

`void location_free` (`Location *_location`)

`Venue *venue` (`Location *location`, `const char *title`, `const char *address`, `const char *foursquare_id`)

`void venue_free` (`Venue *_venue`)

`void message_free` (`Message *message`)

`Bot *bot` (`const char *token`, `User *user`)

`void bot_free` (`Bot *bot`)

`ChosenInlineResult *chosen_inline_result` (`const char *result_id`, `User *from`, `Location *location`, `const char *inline_message_id`, `const char *query`)

`void chosen_inline_result_free` (`ChosenInlineResult *cir`)

`Update *update` (`long int update_id`, `Message *message`, `Message *edited_message`, `Message *channel_post`, `Message *edited_channel_post`, `InlineQuery *inline_query`, `ChosenInlineResult *chosen_inline_result`, `CallbackQuery *callback_query`, `ShippingQuery *shipping_query`, `PreCheckoutQuery *pre_checkout_query`)

`void update_free` (`Update *ouupdate`)

`void update_add` (`Update *dest`, `Update *src`)

`Update *update_get` (`Update *u`, `int index`)

`size_t update_len` (`Update *u`)


```

ChatMember *chat_member (User *user, const char *status, long int until_date,
                          bool can_be_edited, bool can_change_info, bool can_post_messages,
                          bool can_edit_messages, bool can_delete_messages, bool can_invite_users,
                          bool can_restrict_members, bool can_pin_messages,
                          bool can_promote_members, bool can_send_messages,
                          bool can_send_media_messages, bool can_send_other_messages,
                          bool can_add_web_page_previews)

void chat_member_free (ChatMember *chatMember)

void chat_member_add (ChatMember *dest, ChatMember *src)

ChatMember *chat_member_get (ChatMember *chatMember, int index)

size_t chat_member_len (ChatMember *chatMember)

InlineQuery *inline_query (const char *id, User *user, Location *location, const char *query, const
                           char *offset)

void inline_query_free (InlineQuery *inline_query)

CallbackQuery *callback_query (const char *id, User *user, Message *message, const char *in-
                               line_message_id, const char *chat_instance, const char *data, const
                               char *game_short_name)

void callback_query_free (CallbackQuery *callback_query)

VideoNote *video_note (const char *file_id, long length, long duration, PhotoSize *photo_size,
                       long file_size)

void video_note_free (VideoNote *video_note)

Invoice *invoice (const char *title, const char *description, const char *start_parameter, const char *cur-
                 rency, long total_amount)

void invoice_free (Invoice *invoice)

ShippingQuery *shipping_query (const char *id, User *from, const char *invoice_payload, ShippingAd-
                              dress *shipping_address)

void shipping_query_free (ShippingQuery *shipping_query)

ShippingAddress *shipping_address (const char *country_code, const char *state, const char *city, const
                                   char *street_line1, const char *street_line2, const char *post_code)

void shipping_address_free (ShippingAddress *shipping_address)

OrderInfo *order_info (const char *name, const char *phone_number, const char *email, ShippingAd-
                      dress *shipping_address)

void order_info_free (OrderInfo *order_info)

void pre_checkout_query_free (PreCheckoutQuery *pcq)

SuccessfulPayment *successful_payment (const char *currency, long total_amount,
                                       const char *invoice_payload, const char *ship-
                                       ping_option_id, OrderInfo *oorder_info, const
                                       char *telegram_payment_charge_id, const char *provi-
                                       der_payment_charge_id)

void successful_payment_free (SuccessfulPayment *spayment)

File *file (const char *file_id, long int file_size, const char *file_path)

void file_free (File *ofile)

UserProfilePhotos *user_profile_photos (long int total_count, PhotoSize ** photo_size)

```

```
void user_profile_photos_free (UserProfilePhotos *oupp)
ChatPhoto *chat_photo (const char *small_file_id, const char *big_file_id)
void chat_photo_free (ChatPhoto *ochat_photo)
void error (long int error_code, const char *description)
void error_free ()
Error *get_error ()
void framebot_add (Framebot *framebot, Update *update)
void framebot_free (Framebot *framebot)
void framebot_init ()
Bot * framebot (const char *token)
refjson *generic_method_call (const char *token, char *formats, ...)
Error * show_error ()
User * get_me (const char *token)
Framebot *get_updates (Bot *bot, Framebot *framebot, long int offset, long int limit, long int timeout,
char *allowed_updates)
Message * send_message (Bot *bot, char * chat_id, char *text, char * parse_mode, bool disable_web_page_preview, long int reply_to_message_id, char * reply_markup)
Message * send_message_chat (Bot *bot, long int chat_id, char *text, char * parse_mode, bool disable_web_page_preview, long int reply_to_message_id, char * reply_markup)
Message * forward_message_from (Bot * bot, long int chat_id, char * from_chat_id, long int message_id)
Message * forward_message_from_chat (Bot * bot, char * chat_id, long int from_chat_id, long int message_id)
Message * forward_message (Bot * bot, char * chat_id, char * from_chat_id, long int message_id)
Message * forward_message_chat (Bot * bot, long int chat_id, long int from_chat_id, long int message_id)
Message * send_photo (Bot * bot, char * chat_id, char * filename, char * caption, long int reply_to_message_id, char * reply_markup)
Message * send_photo_chat (Bot * bot, long int chat_id, char * filename, char * caption, long int reply_to_message_id, char * reply_markup)
Message * send_audio (Bot *bot, char * chat_id, char * filename, char * caption, long int duration, char * performer, char * title, long int reply_to_message_id, char * reply_markup)
Message * send_audio_chat (Bot * bot, long int chat_id, char * filename, char * caption, long int duration, char * performer, char * title, long int reply_to_message_id, char * reply_markup)
Message * send_document (Bot * bot, char * chat_id, char * filename, char * caption, long int reply_to_message_id, char * reply_markup)
Message * send_document_chat (Bot * bot, long int chat_id, char * filename, char * caption, long int reply_to_message_id, char * reply_markup)
Message * send_video (Bot * bot, char * chat_id, char * video, long int duration, long int width, long int height, char * caption, long int reply_to_message_id, char * reply_markup)
```

Message * **send_video_chat** (*Bot* * *bot*, long int *chat_id*, char * *video*, long int *duration*, long int *width*, long int *height*, char * *caption*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_voice** (*Bot* * *bot*, char * *chat_id*, char * *filename*, char * *caption*, long int *duration*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_voice_chat** (*Bot* * *bot*, long int *chat_id*, char * *filename*, char * *caption*, long int *duration*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_video_note** (*Bot* * *bot*, char * *chat_id*, char * *filename*, long int *duration*, long int *length*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_video_note_chat** (*Bot* * *bot*, long int *chat_id*, char * *filename*, long int *duration*, long int *length*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_location** (*Bot* * *bot*, char * *chat_id*, float *latitude*, float *longitude*, long int *live_period*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_location_chat** (*Bot* * *bot*, long int *chat_id*, float *latitude*, float *longitude*, long int *live_period*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **edit_message_live_location** (*Bot* * *bot*, char * *chat_id*, long int *message_id*, char * *inline_message_id*, float *latitude*, float *longitude*, char * *reply_markup*)

Message * **edit_message_live_location_chat** (*Bot* * *bot*, long int *chat_id*, long int *message_id*, char * *inline_message_id*, float *latitude*, float *longitude*, char * *reply_markup*)

Message * **stop_message_live_location** (*Bot* * *bot*, char * *chat_id*, long int *message_id*, char * *inline_message_id*, char * *reply_markup*)

Message * **stop_message_live_location_chat** (*Bot* * *bot*, long int *chat_id*, long int *message_id*, char * *inline_message_id*, char * *reply_markup*)

Message * **send_venue** (*Bot* * *bot*, char * *chat_id*, float *latitude*, float *longitude*, char * *title*, char * *address*, char * *foursquare_id*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_venue_chat** (*Bot* * *bot*, long int *chat_id*, float *latitude*, float *longitude*, char * *title*, char * *address*, char * *foursquare_id*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_contact** (*Bot* * *bot*, char * *chat_id*, char * *phone_number*, char * *first_name*, char * *last_name*, long int *reply_to_message_id*, char * *reply_markup*)

Message * **send_contact_chat** (*Bot* * *bot*, long int *chat_id*, char * *phone_number*, char * *first_name*, char * *last_name*, long int *reply_to_message_id*, char * *reply_markup*)

int **send_chat_action** (*Bot* * *bot*, char * *chat_id*, char * *action*)

int **send_chat_action_chat** (*Bot* * *bot*, long int *chat_id*, char * *action*)

UserProfilePhotos * **get_user_profile_photos** (*Bot* * *bot*, char * *dir*, char * *user_id*, long *offset*, long *limit*)

UserProfilePhotos * **get_user_profile_photos_chat** (*Bot* * *bot*, char * *dir*, long *user_id*, long *offset*, long *limit*)

char * **get_file** (*Bot* * *bot*, char * *dir*, const char * *file_id*)

bool **kick_chat_member** (*Bot* * *bot*, char * *chat_id*, long int *user_id*, char * *until_date*)

bool **kick_chat_member_chat** (*Bot* * *bot*, long int *chat_id*, long int *user_id*, char * *until_date*)

bool **unban_chat_member** (*Bot* * *bot*, char * *chat_id*, long int *user_id*)

bool **unban_chat_member_chat** (*Bot* * *bot*, long int *chat_id*, long int *user_id*)

```

bool restrict_chat_member (Bot *bot, char *chat_id, long int user_id, long int until_date,
                          bool can_send_messages, bool can_send_media_messages,
                          bool can_send_other_messages, bool can_add_web_page_previews)

bool restrict_chat_member_chat (Bot *bot, long int chat_id, long int user_id,
                               long int until_date, bool can_send_messages,
                               bool can_send_media_messages, bool can_send_other_messages,
                               bool can_add_web_page_previews)

bool promote_chat_member (Bot *bot, char *chat_id, long int user_id, bool can_change_info,
                          bool can_post_messages, bool can_edit_messages,
                          bool can_delete_messages, bool can_invite_users,
                          bool can_restrict_members, bool can_pin_messages,
                          bool can_promote_members)

bool promote_chat_member_chat (Bot *bot, long int chat_id, long int user_id, bool can_change_info,
                               bool can_post_messages, bool can_edit_messages,
                               bool can_delete_messages, bool can_invite_users,
                               bool can_restrict_members, bool can_pin_messages,
                               bool can_promote_members)

char *export_chat_invite_link (Bot *bot, char *chat_id)

char *export_chat_invite_link_chat (Bot *bot, long int chat_id)

int set_chat_photo (Bot *bot, char *chat_id, char *filename)

int set_chat_photo_chat (Bot *bot, long int chat_id, char *filename)

int delete_chat_photo (Bot *bot, char *chat_id)

int delete_chat_photo_chat (Bot *bot, long int chat_id)

int set_chat_title (Bot *bot, char *chat_id, char *title)

int set_chat_title_chat (Bot *bot, long int chat_id, char *title)

bool set_chat_description (Bot *bot, char *chat_id, char *description)

bool set_chat_description_chat (Bot *bot, long int chat_id, char *description)

bool pin_chat_message (Bot *bot, char *chat_id, long int message_id, bool disable_notification)

bool pin_chat_message_chat (Bot *bot, long int chat_id, long int message_id, bool disable_notification)

bool unpin_chat_message (Bot *bot, char *chat_id)

bool unpin_chat_message_chat (Bot *bot, long int chat_id)

bool leave_chat (Bot *bot, char *chat_id)

bool leave_chat_chat (Bot *bot, long int chat_id)

Chat *get_chat (Bot *bot, char *chat_id)

Chat *get_chat_chat (Bot *bot, long int chat_id)

ChatMember *get_chat_administrators (Bot *bot, char *chat_id)

ChatMember *get_chat_administrators_chat (Bot *bot, long int chat_id)

int get_chat_members_count (Bot *bot, char *chat_id)

int get_chat_members_count_chat (Bot *bot, long int chat_id)

ChatMember *get_chat_member (Bot *bot, char *chat_id, long int user_id)

```

ChatMember ***get_chat_member_chat** (*Bot* *bot, long int chat_id, long int user_id)

Message ***edit_message_text** (*Bot* *bot, char *chat_id, long int message_id, char *inline_message_id, char *text, char *parse_mode, bool disable_web_page_preview, char *reply_markup)

Message ***edit_message_text_chat** (*Bot* *bot, long int chat_id, long int message_id, char *inline_message_id, char *text, char *parse_mode, bool disable_web_page_preview, char *reply_markup)

Message ***edit_message_caption** (*Bot* *bot, char *chat_id, long int message_id, char *inline_message_id, char *caption, char *reply_markup)

Message ***edit_message_caption_chat** (*Bot* *bot, long int chat_id, long int message_id, char *inline_message_id, char *caption, char *reply_markup)

Message ***edit_message_reply_markup** (*Bot* *bot, char *chat_id, long int message_id, char *inline_message_id, char *reply_markup)

Message ***edit_message_reply_markup_chat** (*Bot* *bot, long int chat_id, long int message_id, char *inline_message_id, char *reply_markup)

bool **answerInlineQuery** (*Bot* *bot, char *inline_query_id, char *results, long int cache_time, bool is_personal, char *next_offset, char *switch_pm_text, char *switch_pm_parameter)

void **set_notification** (bool disable_notification)

bool **get_notification** ()

void **set_parse_mode** (int32_t mode)

int32_t **get_parse_mode** ()

A

animation (função C), 20
animation_free (função C), 20
answerInlineQuery (função C), 25
audio (função C), 19
Audio (tipo C), 10
Audio.duration (membro C), 10
Audio.file_id (membro C), 10
Audio.file_size (membro C), 10
Audio.mime_type (membro C), 10
Audio.performer (membro C), 10
Audio.title (membro C), 10
audio_free (função C), 19

B

bot (função C), 20
Bot (tipo C), 9
Bot.token (membro C), 9, 10
Bot.user (membro C), 9, 10
bot_free (função C), 20

C

callback_query (função C), 21
callback_query_free (função C), 21
CallbackGame (tipo C), 16
CallbackGame.chat_id (membro C), 17
CallbackGame.disable_edit_message (membro C), 17
CallbackGame.force (membro C), 17
CallbackGame.inline_message_id (membro C), 17
CallbackGame.message_id (membro C), 17
CallbackGame.score (membro C), 16
CallbackGame.user_id (membro C), 16
CallbackQuery (tipo C), 15
CallbackQuery.chat_instance (membro C), 15
CallbackQuery.data (membro C), 15
CallbackQuery.from (membro C), 15
CallbackQuery.game_short_name (membro C), 15
CallbackQuery.id (membro C), 15
CallbackQuery.inline_message_id (membro C), 15

CallbackQuery.message (membro C), 15
Chat (tipo C), 10
Chat.all_members_are_administrators (membro C), 10
Chat.can_set_sticker_set (membro C), 10
Chat.description (membro C), 10
Chat.first_name (membro C), 10
Chat.id (membro C), 10
Chat.invite_link (membro C), 10
Chat.last_name (membro C), 10
Chat.photo (membro C), 10
Chat.pinned_message (membro C), 10
Chat.sticker_set_name (membro C), 10
Chat.title (membro C), 10
Chat.type (membro C), 10
Chat.username (membro C), 10
chat_free (função C), 19
chat_member (função C), 20
chat_member_add (função C), 21
chat_member_free (função C), 21
chat_member_get (função C), 21
chat_member_len (função C), 21
chat_photo (função C), 22
chat_photo_free (função C), 22
ChatMember (tipo C), 16
ChatMember.can_add_web_page_previews (membro C),
16
ChatMember.can_be_edited (membro C), 16
ChatMember.can_change_info (membro C), 16
ChatMember.can_delete_messages (membro C), 16
ChatMember.can_edit_messages (membro C), 16
ChatMember.can_invite_users (membro C), 16
ChatMember.can_pin_messages (membro C), 16
ChatMember.can_post_messages (membro C), 16
ChatMember.can_promote_members (membro C), 16
ChatMember.can_restrict_members (membro C), 16
ChatMember.can_send_media_messages (membro C), 16
ChatMember.can_send_messages (membro C), 16
ChatMember.can_send_other_messages (membro C), 16
ChatMember.next (membro C), 16
ChatMember.status (membro C), 16

ChatMember.until_date (membro C), 16
ChatMember.user (membro C), 16
ChatPhoto (tipo C), 10
ChatPhoto.big_file_id (membro C), 10
ChatPhoto.small_file_id (membro C), 10
chosen_inline_result (função C), 20
chosen_inline_result_free (função C), 20
ChosenInlineResult (tipo C), 15
ChosenInlineResult.from (membro C), 15
ChosenInlineResult.inline_message_id (membro C), 15
ChosenInlineResult.location (membro C), 15
ChosenInlineResult.query (membro C), 15
ChosenInlineResult.result_id (membro C), 15
contact (função C), 20
Contact (tipo C), 12
Contact.first_name (membro C), 12
Contact.last_name (membro C), 12
Contact.phone_number (membro C), 12
Contact.user_id (membro C), 12
contact_free (função C), 20

D

delete_chat_photo (função C), 24
delete_chat_photo_chat (função C), 24
document (função C), 19
document_free (função C), 20

E

edit_message_caption (função C), 25
edit_message_caption_chat (função C), 25
edit_message_live_location (função C), 23
edit_message_live_location_chat (função C), 23
edit_message_reply_markup (função C), 25
edit_message_reply_markup_chat (função C), 25
edit_message_text (função C), 25
edit_message_text_chat (função C), 25
error (função C), 22
error_free (função C), 22
export_chat_invite_link (função C), 24
export_chat_invite_link_chat (função C), 24

F

file (função C), 21
File (tipo C), 16
File.file_id (membro C), 16
File.file_path (membro C), 16
File.file_size (membro C), 16
file_free (função C), 21
ForceReply (tipo C), 17
ForceReply.force_reply (membro C), 17
ForceReply.selective (membro C), 18
ForceReply.type (membro C), 17
forward_message (função C), 22
forward_message_chat (função C), 22

forward_message_from (função C), 22
forward_message_from_chat (função C), 22
framebot (função C), 22
Framebot (tipo C), 15
Framebot.callback_query (membro C), 15
Framebot.channel_post (membro C), 15, 16
Framebot.chosen_inline_result (membro C), 15, 16
Framebot.edited_channel_post (membro C), 15, 16
Framebot.edited_message (membro C), 15, 16
Framebot.inline_query (membro C), 15
Framebot.message (membro C), 15, 16
Framebot.next (membro C), 15
Framebot.pre_checkout_query (membro C), 15, 16
Framebot.shipping_query (membro C), 15
Framebot.update_id (membro C), 15
framebot_add (função C), 22
framebot_free (função C), 22
framebot_init (função C), 22

G

game (função C), 20
Game (tipo C), 11
Game.animation (membro C), 11
Game.description (membro C), 11
Game.file_id (membro C), 11
Game.file_name (membro C), 11
Game.file_size (membro C), 11
Game.mime_type (membro C), 11
Game.text (membro C), 11
Game.thumb (membro C), 11
Game.title (membro C), 11
game_free (função C), 20
generic_method_call (função C), 22
get_chat (função C), 24
get_chat_administrators (função C), 24
get_chat_administrators_chat (função C), 24
get_chat_chat (função C), 24
get_chat_member (função C), 24
get_chat_member_chat (função C), 24
get_chat_members_count (função C), 24
get_chat_members_count_chat (função C), 24
get_error (função C), 22
get_file (função C), 23
get_me (função C), 22
get_notification (função C), 25
get_parse_mode (função C), 25
get_updates (função C), 22
get_user_profile_photos (função C), 23
get_user_profile_photos_chat (função C), 23

I

inline_query (função C), 21
inline_query_free (função C), 21
InlineKeyboardButton (tipo C), 17

- InlineKeyboardButton.callback_data (membro C), 17
 - InlineKeyboardButton.callback_game (membro C), 17
 - InlineKeyboardButton.pay (membro C), 17
 - InlineKeyboardButton.switch_inline_query_current_chat (membro C), 17
 - InlineKeyboardButton.switch_nline_query (membro C), 17
 - InlineKeyboardButton.text (membro C), 17
 - InlineKeyboardButton.url (membro C), 17
 - InlineKeyboardMarkup (tipo C), 17
 - InlineKeyboardMarkup.type (membro C), 17
 - InlineQuery (tipo C), 15
 - InlineQuery.from (membro C), 15
 - InlineQuery.id (membro C), 15
 - InlineQuery.location (membro C), 15
 - InlineQuery.offset (membro C), 15
 - InlineQuery.query (membro C), 15
 - invoice (função C), 21
 - Invoice (tipo C), 12
 - Invoice.currency (membro C), 12
 - Invoice.description (membro C), 12
 - Invoice.start_parameter (membro C), 12
 - Invoice.title (membro C), 12
 - Invoice.total_amount (membro C), 12
 - invoice_free (função C), 21
- ## K
- Keyboard (tipo C), 18
 - Keyboard._force_reply (membro C), 18
 - Keyboard.inline_keyboard_markup (membro C), 18
 - Keyboard.reply_keyboard_markup (membro C), 18
 - Keyboard.reply_keyboard_remove (membro C), 18
 - Keyboard.type (membro C), 18
 - KeyboardButton (tipo C), 17
 - KeyboardButton.next (membro C), 17
 - KeyboardButton.request_contact (membro C), 17
 - KeyboardButton.request_location (membro C), 17
 - KeyboardButton.text (membro C), 17
 - kick_chat_member (função C), 23
 - kick_chat_member_chat (função C), 23
- ## L
- leave_chat (função C), 24
 - leave_chat_chat (função C), 24
 - location (função C), 20
 - Location (tipo C), 12
 - Location.latitude (membro C), 12
 - Location.longitude (membro C), 12
 - location_free (função C), 20
- ## M
- Message (tipo C), 13
 - Message.audio (membro C), 14
 - Message.author_signature (membro C), 14
 - Message.caption (membro C), 14
 - Message.caption_entities (membro C), 14
 - Message.channel_chat_created (membro C), 14
 - Message.chat (membro C), 14
 - Message.connected_website (membro C), 14
 - Message.contact (membro C), 14
 - Message.date (membro C), 13
 - Message.delete_chat_photo (membro C), 14
 - Message.document (membro C), 14
 - Message.edit_date (membro C), 14
 - Message.forward_date (membro C), 14
 - Message.forward_from (membro C), 14
 - Message.forward_from_chat (membro C), 14
 - Message.forward_from_message_id (membro C), 14
 - Message.forward_signature (membro C), 14
 - Message.from (membro C), 13
 - Message.game (membro C), 14
 - Message.group_chat_created (membro C), 14
 - Message.invoice (membro C), 14
 - Message.left_chat_member (membro C), 14
 - Message.location (membro C), 14
 - Message.media_group_id (membro C), 14
 - Message.message_id (membro C), 13
 - Message.migrate_from_chat_id (membro C), 14
 - Message.migrate_to_chat_id (membro C), 14
 - Message.new_chat_members (membro C), 14
 - Message.new_chat_title (membro C), 14
 - Message.pinned_message (membro C), 14
 - Message.reply_to_message (membro C), 14
 - Message.sticker (membro C), 14
 - Message.successful_payment (membro C), 14
 - Message.supergroup_chat_created (membro C), 14
 - Message.text (membro C), 14
 - Message.venue (membro C), 14
 - Message.video (membro C), 14
 - Message.video_note (membro C), 14
 - Message.voice (membro C), 14
 - message_entity_add (função C), 19
 - message_entity_free (função C), 19
 - message_entity_get (função C), 19
 - message_entity_len (função C), 19
 - message_free (função C), 20
 - MessageEntity (tipo C), 10
 - MessageEntity.length (membro C), 10
 - MessageEntity.next (membro C), 10
 - MessageEntity.offset (membro C), 10
 - MessageEntity.type (membro C), 10
 - MessageEntity.url (membro C), 10
 - MessageEntity.user (membro C), 10
- ## O
- order_info (função C), 21
 - order_info_free (função C), 21
 - OrderInfo (tipo C), 13

OrderInfo.email (membro C), 13
OrderInfo.name (membro C), 13
OrderInfo.phone_number (membro C), 13
OrderInfo.shipping_address (membro C), 13

P

photo_size_add (função C), 19
photo_size_free (função C), 19
photo_size_get (função C), 19
photo_size_len (função C), 19
PhotoSize (tipo C), 11
PhotoSize.file_id (membro C), 11
PhotoSize.file_size (membro C), 11
PhotoSize.height (membro C), 11
PhotoSize.next (membro C), 11
PhotoSize.width (membro C), 11
pin_chat_message (função C), 24
pin_chat_message_chat (função C), 24
pre_checkout_query_free (função C), 21
PreCheckoutQuery (tipo C), 13
PreCheckoutQuery.currency (membro C), 13
PreCheckoutQuery.from (membro C), 13
PreCheckoutQuery.id (membro C), 13
PreCheckoutQuery.invoice_payload (membro C), 13
PreCheckoutQuery.order_info (membro C), 13
PreCheckoutQuery.shipping_option_id (membro C), 13
PreCheckoutQuery.total_amount (membro C), 13
promote_chat_member (função C), 24
promote_chat_member_chat (função C), 24

R

ReplyKeyboardMarkup (tipo C), 17
ReplyKeyboardMarkup.one_time_keyboard (membro C), 17
ReplyKeyboardMarkup.resize_keyboard (membro C), 17
ReplyKeyboardMarkup.selective (membro C), 17
ReplyKeyboardMarkup.type (membro C), 17
ReplyKeyboardRemove (tipo C), 17
ReplyKeyboardRemove.remove_keyboard (membro C), 17
ReplyKeyboardRemove.selective (membro C), 17
ReplyKeyboardRemove.type (membro C), 17
restrict_chat_member (função C), 23
restrict_chat_member_chat (função C), 24

S

send_audio (função C), 22
send_audio_chat (função C), 22
send_chat_action (função C), 23
send_chat_action_chat (função C), 23
send_contact (função C), 23
send_contact_chat (função C), 23
send_document (função C), 22
send_document_chat (função C), 22

send_location (função C), 23
send_location_chat (função C), 23
send_message (função C), 22
send_message_chat (função C), 22
send_photo (função C), 22
send_photo_chat (função C), 22
send_venue (função C), 23
send_venue_chat (função C), 23
send_video (função C), 22
send_video_chat (função C), 22
send_video_note (função C), 23
send_video_note_chat (função C), 23
send_voice (função C), 23
send_voice_chat (função C), 23
set_chat_description (função C), 24
set_chat_description_chat (função C), 24
set_chat_photo (função C), 24
set_chat_photo_chat (função C), 24
set_chat_title (função C), 24
set_chat_title_chat (função C), 24
set_notification (função C), 25
set_parse_mode (função C), 25
shipping_address (função C), 21
shipping_address_free (função C), 21
shipping_query (função C), 21
shipping_query_free (função C), 21
ShippingAddress (tipo C), 12
ShippingAddress.city (membro C), 12
ShippingAddress.country_code (membro C), 12
ShippingAddress.post_code (membro C), 13
ShippingAddress.state (membro C), 12
ShippingAddress.street_line1 (membro C), 13
ShippingAddress.street_line2 (membro C), 13
ShippingQuery (tipo C), 13
ShippingQuery.from (membro C), 13
ShippingQuery.id (membro C), 13
ShippingQuery.invoice_payload (membro C), 13
ShippingQuery.shipping_address (membro C), 13
show_error (função C), 22
sticker (função C), 20
Sticker (tipo C), 11
Sticker.emoji (membro C), 11
Sticker.file_id (membro C), 11
Sticker.file_size (membro C), 11
Sticker.height (membro C), 11
Sticker.thumb (membro C), 11
Sticker.width (membro C), 11
sticker_free (função C), 20
stop_message_live_location (função C), 23
stop_message_live_location_chat (função C), 23
successful_payment (função C), 21
successful_payment_free (função C), 21
SuccessfulPayment (tipo C), 13
SuccessfulPayment.currency (membro C), 13

SuccessfulPayment.invoice_payload (membro C), 13
 SuccessfulPayment.order_info (membro C), 13
 SuccessfulPayment.provider_payment_charge_id (membro C), 13
 SuccessfulPayment.shipping_option_id (membro C), 13
 SuccessfulPayment.telegram_payment_charge_id (membro C), 13
 SuccessfulPayment.total_amount (membro C), 13

U

unban_chat_member (função C), 23
 unban_chat_member_chat (função C), 23
 unpin_chat_message (função C), 24
 unpin_chat_message_chat (função C), 24
 update (função C), 20
 update_add (função C), 20
 update_free (função C), 20
 update_get (função C), 20
 update_len (função C), 20
 user (função C), 19
 User (tipo C), 9
 User.first_name (membro C), 9
 User.id (membro C), 9
 User.is_bot (membro C), 9
 User.language_code (membro C), 9
 User.last_name (membro C), 9
 User.next (membro C), 9
 User.username (membro C), 9
 user_add (função C), 19
 user_free (função C), 19
 user_profile_photos (função C), 21
 user_profile_photos_free (função C), 21
 UserProfilePhotos (tipo C), 16
 UserProfilePhotos.total_count (membro C), 16

V

venue (função C), 20
 Venue (tipo C), 12
 Venue.address (membro C), 12
 Venue.foursquare_id (membro C), 12
 Venue.location (membro C), 12
 Venue.title (membro C), 12
 venue_free (função C), 20
 video (função C), 20
 Video (tipo C), 11
 Video.duration (membro C), 11
 Video.file_id (membro C), 11
 Video.file_size (membro C), 11
 Video.height (membro C), 11
 Video.mime_type (membro C), 11
 Video.thumb (membro C), 11
 Video.width (membro C), 11
 video_free (função C), 20
 video_note (função C), 21

video_note_free (função C), 21
 VideoNote (tipo C), 11
 VideoNote.duration (membro C), 12
 VideoNote.file_id (membro C), 11
 VideoNote.file_size (membro C), 12
 VideoNote.length (membro C), 12
 VideoNote.thumb (membro C), 12
 voice (função C), 20
 Voice (tipo C), 12
 Voice.duration (membro C), 12
 Voice.file_id (membro C), 12
 Voice.file_size (membro C), 12
 Voice.mime_type (membro C), 12
 voice_free (função C), 20