

---

# **Framebot Documentation**

*Versão 0.1.0*

**Giancarlo and Ródgger**

**15 mar, 2019**



<b>1</b>	<b>Começando</b>	<b>3</b>
1.1	Instalação . . . . .	3
1.2	Configuração . . . . .	4
1.3	Utilização . . . . .	4
1.4	Exemplos . . . . .	4
<b>2</b>	<b>Tutorial</b>	<b>5</b>
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
<b>3</b>	<b>Objetos</b>	<b>9</b>
3.1	Objetos exclusivos . . . . .	9
3.2	Referência completa . . . . .	9
<b>4</b>	<b>Funções</b>	<b>19</b>
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 chamada 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);
    }

    Bot *echo;
    echo = 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_init()`, 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