



Istituto di Scienza e Tecnologie  
dell'Informazione "A. Faedo"  
Consiglio Nazionale delle Ricerche



## *ISTI Technical Reports*

# How to effectively implement a multimedia telegram bot

Giulio Galesi, ISTI-CNR, Pisa, Italy  
Massimo Martinelli, ISTI-CNR, Pisa, Italy

**ISTI-TR-2020/006**



How to effectively implement a multimedia telegram bot

Galesi G., Martinelli M.

ISTI-TR-2020/006

The purpose of this document is to describe the creation of a multimedia BOT for Telegram 1. A bot is a third-party software application that run inside Telegram and performs automated jobs, responding to requests from human users. Users can interact with bots by sending them messages and commands. It is possible to control a bot using HTTPS requests by means of the Telegram Bot API 2 . For the server part a script written in PHP will be described.

Keywords: Bot, Telegram, PHP.

#### Citation

Galesi G., Martinelli M. *How to effectively implement a multimedia telegram bot*. ISTI Technical Reports 2020/006. DOI: 10.32079/ISTI-TR-2020/006.

---

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

Area della Ricerca CNR di Pisa

Via G. Moruzzi 1

56124 Pisa Italy

<http://www.isti.cnr.it>

---

# How to Effectively Implement a Multimedia Telegram Bot

---

8/6/2020

Giulio Galesi and Massimo Martinelli

*Signals and Images Laboratory*  
Institute of Information Science and Technologies  
National Research Council of Italy

1	Introduction.....	3
2	How to implement a Telegram Bot .....	3
2.1	Download Telegram .....	3
2.2	Setup Telegram .....	3
2.3	Setup the bot.....	5
2.4	Set up a webhook.....	6
3	Create the bot app .....	7
3.1	key features of the code.....	7
3.2	The complete listing .....	7
3.3	The App .....	10
3.3.1	Client side (Telegram) .....	10
3.3.2	Server Side .....	12
4	Conclusions and future work .....	13
5	Bibliography.....	13

## 1 Introduction

The purpose of this document is to describe the creation of a multimedia BOT for Telegram <sup>1</sup>. A bot is a third-party software application that runs inside Telegram and performs automated jobs, responding to requests from human users. Users can interact with bots by sending them messages and commands. It is possible to control a bot using HTTPS requests by means of the Telegram Bot API <sup>2</sup>. For the server part a script written in PHP will be described.

## 2 How to implement a Telegram Bot

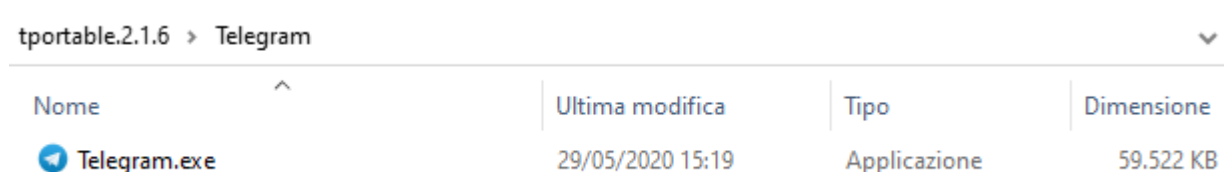
### 2.1 Download Telegram

First download Telegram on your device, depending on the operative system you can find your version at the following addresses:

- Windows portable: [https://telegram.org/dl/desktop/win\\_portable](https://telegram.org/dl/desktop/win_portable)
- Windows installer: <https://telegram.org/dl/desktop/win>
- macOS: <https://telegram.org/dl/macos>
- Android (Google Play):  
<https://play.google.com/store/apps/details?id=org.telegram.messenger>
- Apple (App Store):  
<https://apps.apple.com/app/telegram-messenger/id686449807>

### 2.2 Setup Telegram

Locate the downloaded file and run it (Figg. 1, 2).




Nome	Ultima modifica	Tipo	Dimensione
 Telegram.exe	29/05/2020 15:19	Applicazione	59.522 KB

Figure 1 - Browser's downloads window

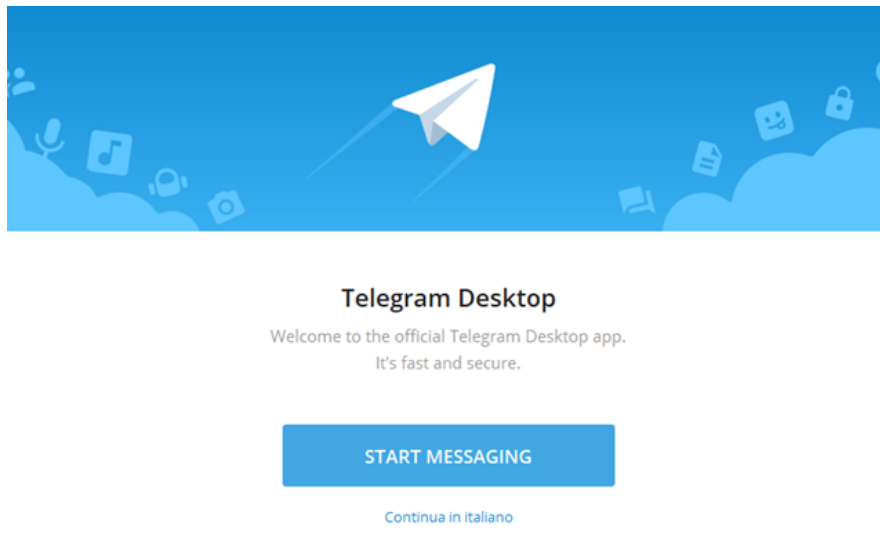


Figure 2 - Telegram Desktop welcome page

Click on Start Messaging then insert your mobile phone number (Fig. 3).

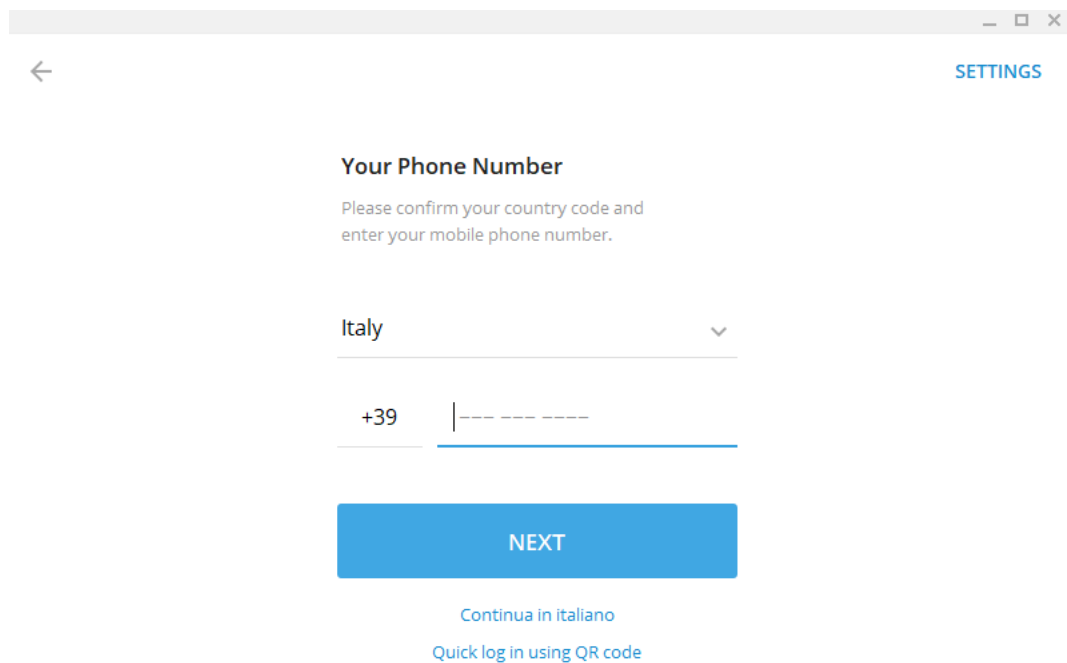
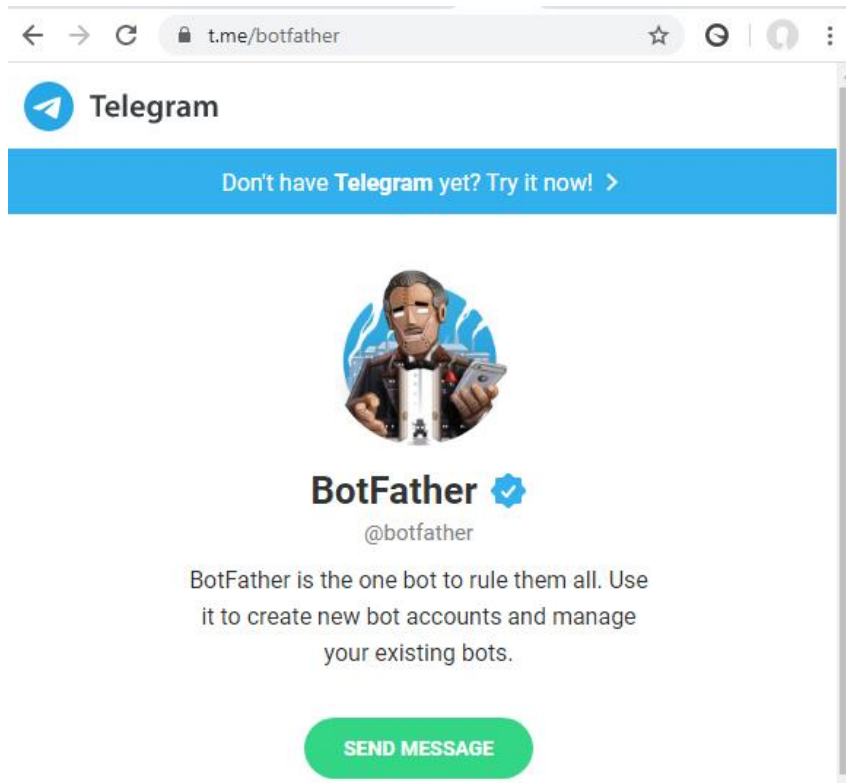


Figure 3 - Phone number setup page

## 2.3 Setup the bot

Once set up, using your browser go to <https://t.me/botfather> and click on “send message” (Fig. 4).



*Figure 4 - BotFather welcome and setup page*

Write `/newbot` in order to generate a new bot and send the message to BotFather <sup>3</sup>.

We choose **mimi** (“ear” in japanese) as name for our bot and **mimicrbot** as username (Fig 5).

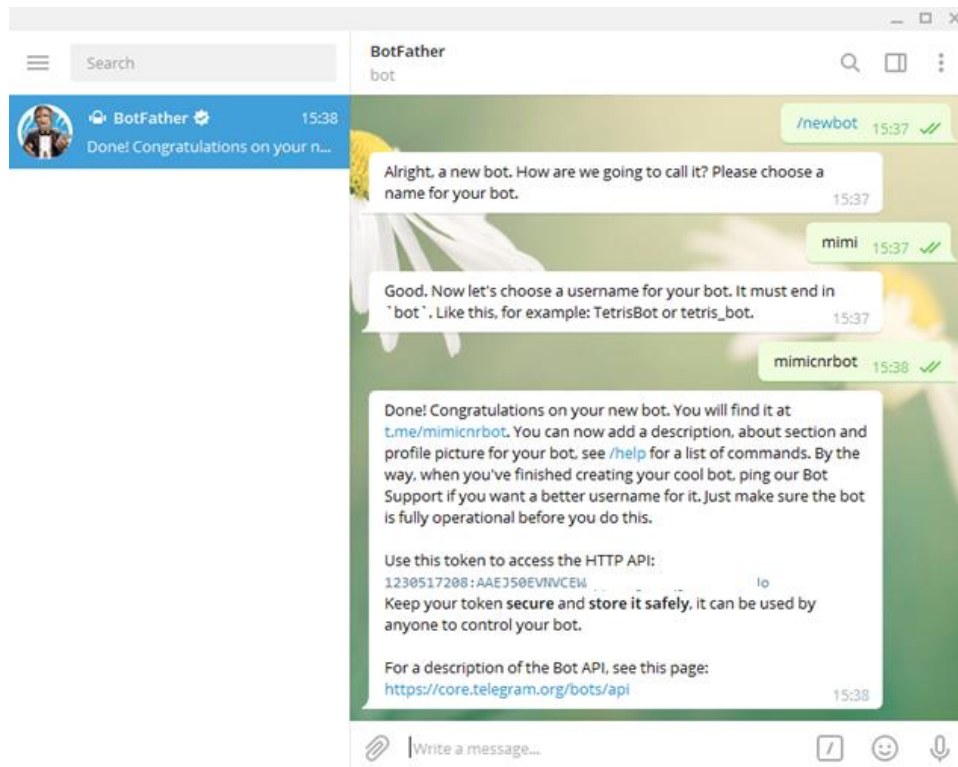


Figure 5 - Bot data configuration

In a few steps we activated our bot which is available at [t.me/mimicrbot](https://t.me/mimicrbot) and we received a secret bot token to access the HTTP API:

**1230517208:AAEJ50EVNVCEWxxxxxxxxxxxxFcJo**

## 2.4 Set up a webhook

Webhook helps to create an automatic bot, that will communicate real time with users, immediately after user writes to bot. Using webhook, every action with bot will send the request to our application.

You can simply setup webhook from the browser search input:

<https://api.telegram.org/bot<SecretBotToken>/setWebhook?url=<webhookPageUrl>>

The webhook page url is where we put the php app, in our case e.g. <https://xxx.altervista.org.xxx.xx/mimi/>

The result is (Fig. 6):



<https://api.telegram.org/bot1230517208:AAEJ50EVNVCEWxxxxxxxxxxxxFcJo/setWebhook?url=https://xxxxxxxxxxxxx/mimi/>



← → ↻ [api.telegram.org/bot1230517208:AAEJ50EVNVCEWxxxxxxxxxxxxFcJo/setWebhook?url=https://xxxxxxxxxxxxx/mimi/](https://api.telegram.org/bot1230517208:AAEJ50EVNVCEWxxxxxxxxxxxxFcJo/setWebhook?url=https://xxxxxxxxxxxxx/mimi/)

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

Figure 6 - Webhook generation and verification

## 3 Create the bot app

### 3.1 key features of the code

Some comments on the key features of the code:

- to handle the call incoming from webhook Telegram server the `file_get_contents` function can be used with a single input parameter, that receives an array:

```
$update = file_get_contents('php://input')
```

- to decode the message we can use the `json_decode` function in order to get a JSON structure, with in input the `$update` message receive and a boolean, that when is set to `TRUE`, will convert objects into associative arrays:

```
json_decode($update, TRUE)
```

- to exit the script if the JSON command is not valid the `exit` function is used:

```
if (!$update) { exit; }
```

- to print the message if a valid not empty command was requested:

```
$message
```

### 3.2 The complete listing

Here follows the complete listing for the server.

```
<?php

$botToken = "1230517208:AAEJ50EVNVCExxxxxxxxxxxxFcJo";

$botAPI = "https://api.telegram.org/bot" . $botToken;

#handle update
```

```
$content = file_get_contents("php://input");
$update = json_decode($content, true);

####

if(!$update)
{
    exit;
}

$message = isset($update['message']) ? $update['message'] : "";
$messageId = isset($message['message_id']) ? $message['message_id'] : "";
$voice = isset($message['voice']) ? $message['voice'] : "";
$voiceId = isset($voice['file_id']) ? $voice['file_id'] : "";
$chatId = isset($message['chat']['id']) ? $message['chat']['id'] : "";
$date = isset($message['date']) ? $message['date'] : "";
$text = isset($message['text']) ? $message['text'] : "";

$gmdate = date('d M Y H:i:s',$date);
$gmora = date('H:i:s',$date);
$gmnick = date('dYHis',$date) ;
$gmrand = rand(1000, 9999);
$gmnickname = $gmnick . $gmrand;

header("Content-Type: application/json");

$response = "";

if(strpos($text, "/start") === 0 || $text=="Ciao")
{
    $response = "Benvenuto! Per favore registra un messaggio vocale. Grazie";
```



```
}
```

```
#voice message
```

```
else if (isset($message['voice'])) {
```

```
    $response = "Grazie! Abbiamo ricevuto il tuo messaggio vocale. Il tuo Numero Identificativo Anonimo è $gmnickname."; # Thanks, we received your vocal message. Your id number is $gmnickname
```

```
    $fileAPI = "https://api.telegram.org/file/bot" . $botToken;
```

```
    $url = $GLOBALS[botAPI] . '/getFile?file_id=' . $voiceld;
```

```
    $fileinfo = file_get_contents($url);
```

```
    $fileinfo = json_decode($fileinfo, TRUE);
```

```
    $filePath = $fileinfo["result"]["file_path"];
```

```
    $url2 = $GLOBALS[fileAPI] . '/' . $filePath;
```

```
    $file_name = basename($url2);
```

```
    /// Download file from URL
```

```
    $ch = curl_init($url2);
```

```
    // Open file
```

```
    $fp = fopen('utenti/' . $gmnickname . $file_name, 'wb');
```

```
    // Set an option for a cURL transfer
```

```
    curl_setopt($ch, CURLOPT_FILE, $fp);
```

```
    curl_setopt($ch, CURLOPT_HEADER, 0);
```

```
    // Perform a cURL session
```

```
    curl_exec($ch);
```

```
// Closes a cURL session and frees all resources

curl_close($ch);

// Close file

fclose($fp);

}

else

{

    $response = "Per favore registra un messaggio vocale tenendo premuto il pulsante in basso a destra
dello schermo che ha come icona un microfono."; # Please record a vocal message by pushing the right
bottom button that has a microphon as icon

}

$parameters = array('chat_id' => $chatId, "text" => $response);

$parameters["method"] = "sendMessage";

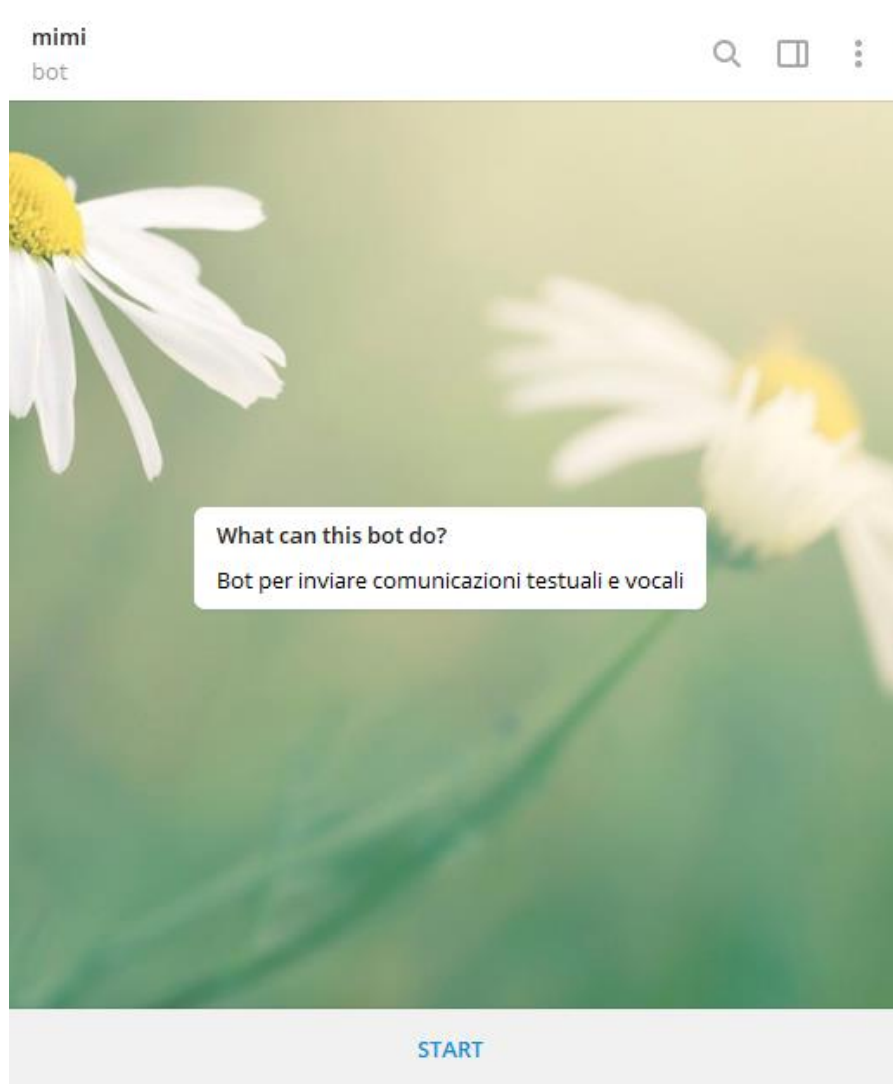
echo json_encode($parameters)
```

## 3.3 The App

### 3.3.1 Client side (Telegram)

Connecting to <https://t.me/mimicnrbot> the bot will show a textual description of what the bot can do.

By clicking the “start” button on the bottom (akin “/start” command), the bot will be activated (Fig. 7).

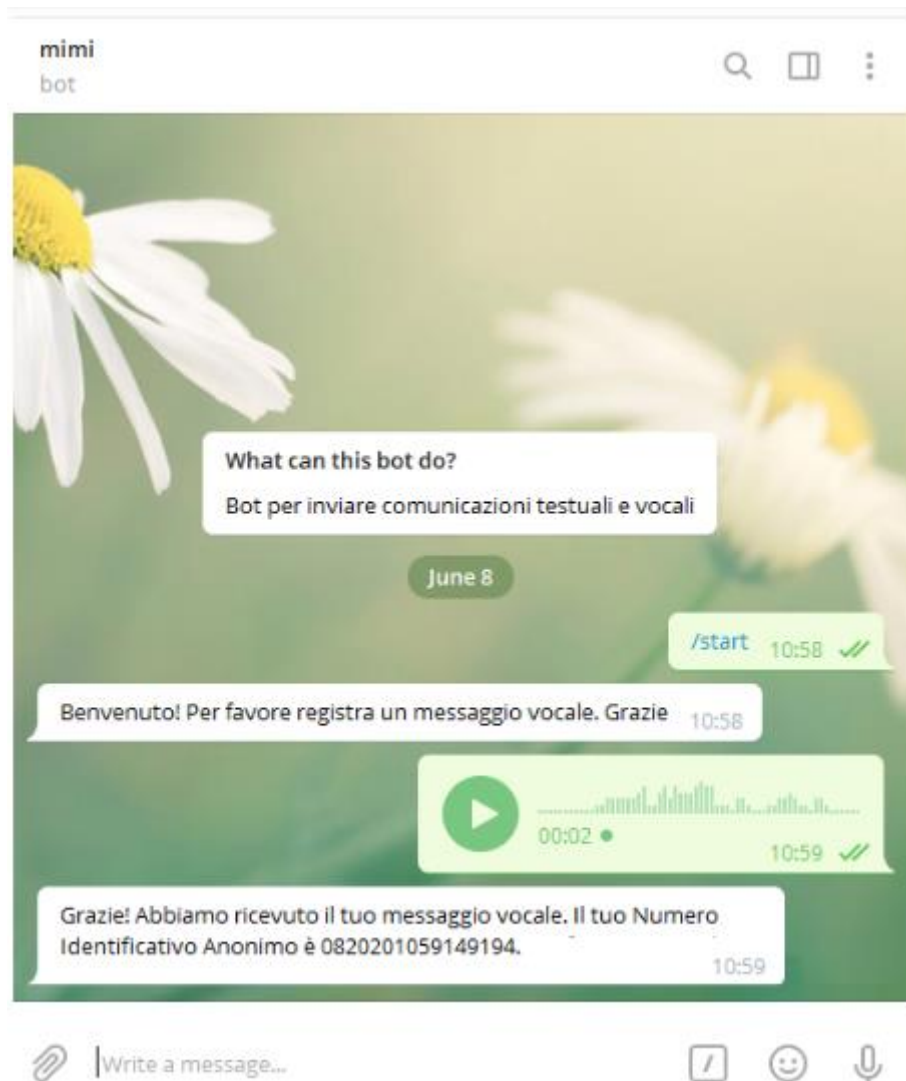


*Figure 7 - Main page of our bot  
(Bot to send text and audio communications)*

The bot will ask the user to send a voice message (*"Welcome, please record a vocal message, thanks"*).

By pressing on the mic icon, it is possible to record a voice message that will be sent to the server.

The bot will assign the user an anonymous ID useful for eventual further processing (*"Thanks we have received your message. Your anonymous id is ... "* (Fig. 8).





*Figure 8 - User interaction with the mimicrbot*

### 3.3.2 Server Side

The voice messages in .oga format (open container format for audio only files) coming from the users are anonymously saved on the server.

The first part of the filename is the ID given provided to the user (Fig. 9).

/mimi/utenti   **PHP** **FTP** Invia files Sposta Copia Cancella


<input type="checkbox"/>	File	Dimensione	Data
-	↑ Parent	-	-
<input type="checkbox"/>	 0820201059149194file_87.oga	14 KB	08/06/2020

Figure 9 - Voice message (.oga) saved on server

## 4 Conclusions and future work

In this document an effective implementation of a multimedia BOT for Telegram has been shown, and its implementation has been described. In a forthcoming document a sort of working session for each user connecting to the bot will be described.

## 5 Bibliography

1. Telegram Messenger <https://telegram.org/>
2. Telegram Bot API <https://core.telegram.org/bots/api>
3. Bots: An introduction for developers <https://core.telegram.org/bots>
4. Bot code examples <https://core.telegram.org/bots/samples>