API MIKROTIK

INTRODUCCIÓN

La interfaz programable de aplicación (API) permite a los usuarios crear soluciones de software personalizadas para comunicarse con RouterOS para recopilar información, ajustar la configuración y administrar el enrutador.

Se puede utilizar para crear herramientas de configuración traducidas o personalizadas para facilitar el uso en la ejecución y administración de enrutadores con RouterOS. Por defecto, la API usa el puerto # 8728 y el servicio está habilitado.

PROTOCOLO

La comunicación con el enrutador se realiza enviando oraciones al enrutador y recibiendo una o más oraciones a cambio. La oración es una secuencia de palabras terminadas por una palabra de longitud cero.

La comunicación se realiza enviando oraciones al enrutador y recibiendo respuestas a las oraciones enviadas. Cada oración enviada al enrutador mediante API debe contener un comando como primera palabra seguida de palabras sin ningún orden en particular, el final de la oración está marcado con una palabra de longitud cero. Cuando el enrutador recibe una oración completa (palabra de comando, ninguna o más palabras de atributo y palabra de longitud cero) se evalúa y ejecuta, luego se forma la respuesta y se devuelve.

PALABRAS API

Las palabras son parte de la oración. Cada palabra debe codificarse de cierta manera: longitud de la palabra seguida del contenido de la palabra. La longitud de la palabra se debe dar como recuento de bytes que se enviarán.

La longitud de la palabra se codifica de la siguiente manera:

Valor de longitud	# de bytes	Codificación
0 <= len <= 0x7F	1	len, byte más bajo
0x80 <= len <= 0x3FFF	2	len | 0x8000, dos bytes inferiores
0x4000 <= len <= 0x1FFFFF	3	len | 0xC00000, tres bytes inferiores
0x200000 <= len <= 0xFFFFFFF	4 4	len | 0xE0000000
len> = 0x10000000	5 5	0xF0 y len como cuatro bytes

- Cada palabra se codifica como longitud, seguida de esa cantidad de bytes de contenido;
- Las palabras se agrupan en oraciones. El final de la oración termina con una palabra de longitud cero;
- El esquema permite la codificación de longitud de hasta 0x7FFFFFFFFF , solo se admite una longitud de cuatro bytes;
- Si el primer byte de la palabra es > = 0xF8 , entonces es un byte de control reservado. Después de recibir un byte de control desconocido, el cliente API no puede continuar, porque no puede saber cómo interpretar los siguientes bytes;
- Actualmente no se utilizan bytes de control;

En general, las palabras se pueden describir así << longitud de palabra codificada > < contenido de palabra >>.

Palabra de comando

La primera palabra de la oración debe ser un comando seguido de palabras de atributo y palabra de longitud cero o palabra de terminación. El nombre de la palabra de comando debe comenzar con '/'. Los nombres de los comandos siguen de cerca la CLI, con espacios reemplazados por '/'.

Palabra de atributo

Cada palabra de comando tiene su propia lista de palabras de atributos según el contenido.

La estructura de palabras del atributo consta de 5 partes en este orden:

- longitud codificada
- prefijo de contenido es igual a signo - =
- Nombre del Atributo
- separando signo igual - =
- valor del atributo si hay uno

Palabra de atributo API

La estructura de palabras del atributo API está en orden estricto:

- longitud codificada
- prefijo de contenido con punto .
- Nombre del Atributo
- nombre pospuesto con igual = signo
- valor de atributo

Palabra de consulta

Las oraciones pueden tener parámetros de consulta adicionales que restringen su alcance. Las palabras de consulta comienzan con '?'.

Palabra de respuesta

Solo lo envía el enrutador. Solo se envía en respuesta a la oración completa enviada por el cliente.

- La primera palabra de respuesta comienza con '!';
- Cada oración enviada genera al menos una respuesta (si la conexión no se termina);
- Última respuesta para cada frase es la respuesta que tiene la primera palabra *!done* ;
- Los errores y las condiciones excepcionales comienzan con *! Trap* ;
- Las respuestas de datos comienzan con *! Re*
- Si la conexión API está cerrada, RouterOS envía *! Fatal* con una razón como respuesta y luego cierra la conexión;

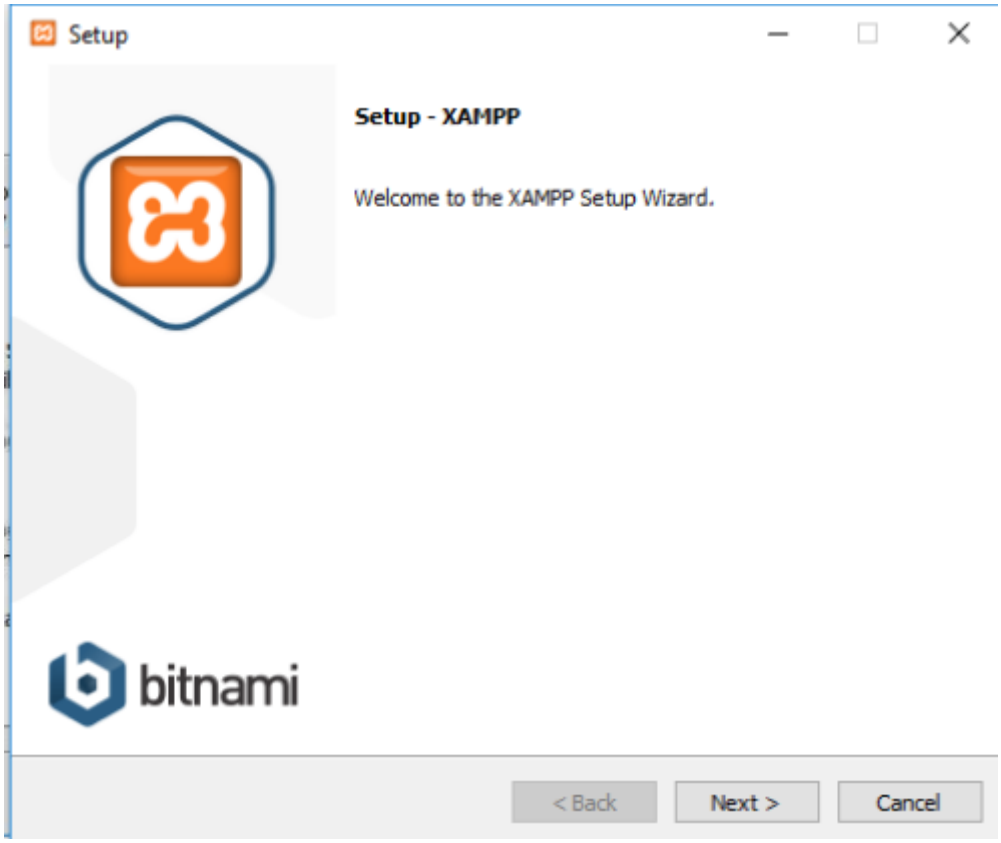
Oraciones API

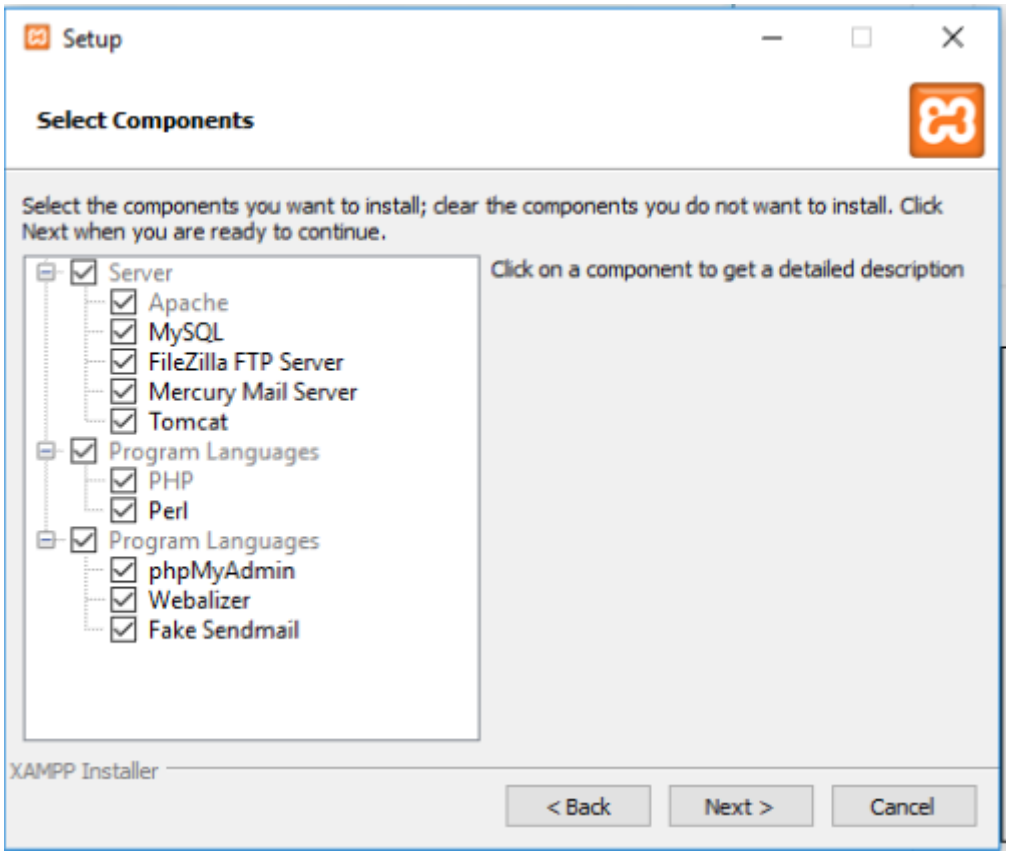
La oración API es el principal objeto de comunicación mediante API.

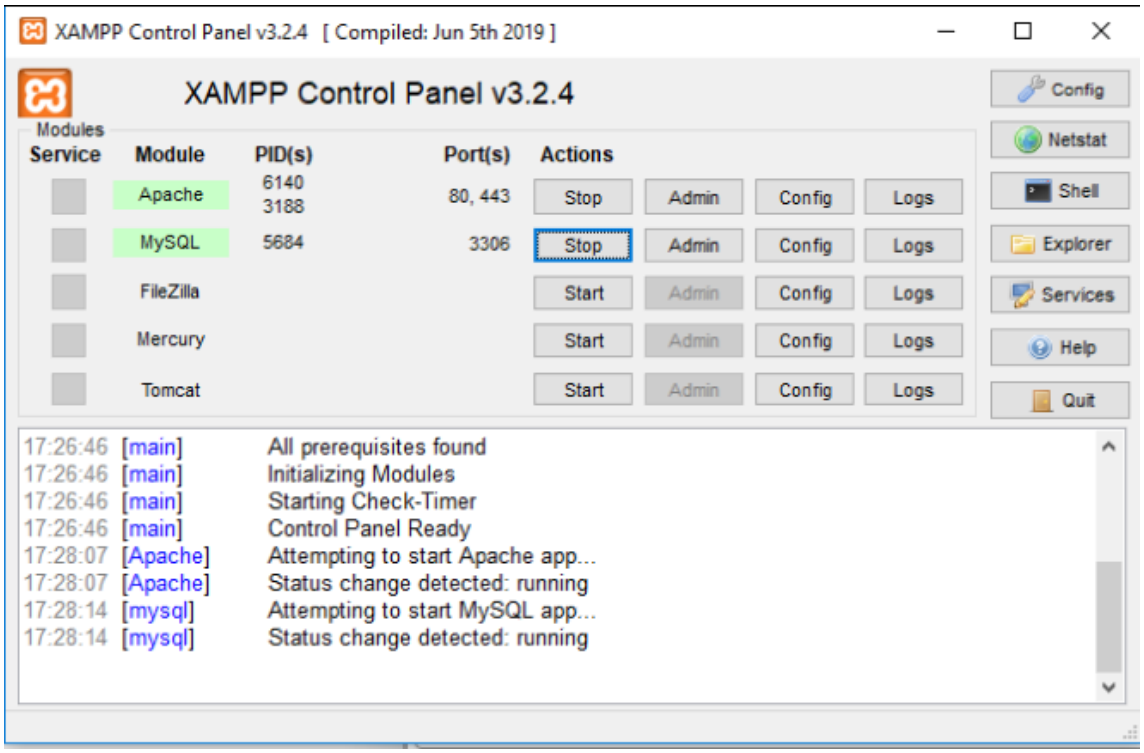
- Las oraciones vacías se ignoran.
- La oración se procesa después de recibir la palabra de longitud cero.
- Existe un límite en el número y el tamaño de las oraciones que el cliente puede enviar antes de iniciar sesión.
- No se debe confiar en el orden de las palabras de atributo. Como el orden y el recuento se pueden cambiar mediante el atributo *.proplist* .
- La estructura de la oración es la siguiente:
    - La primera palabra debe contener la palabra de comando;
    - Debe contener una palabra de longitud cero para terminar la oración;
    - Puede contener ninguna o varias palabras de atributo. No hay un orden particular en cuanto a qué palabras de atributo deben enviarse en la oración, el orden no es importante para las palabras de atributo;
    - Puede contener ninguna o varias palabras de consulta. El orden de las palabras de consulta en la oración es importante.

INSTALACION DE XAMPP

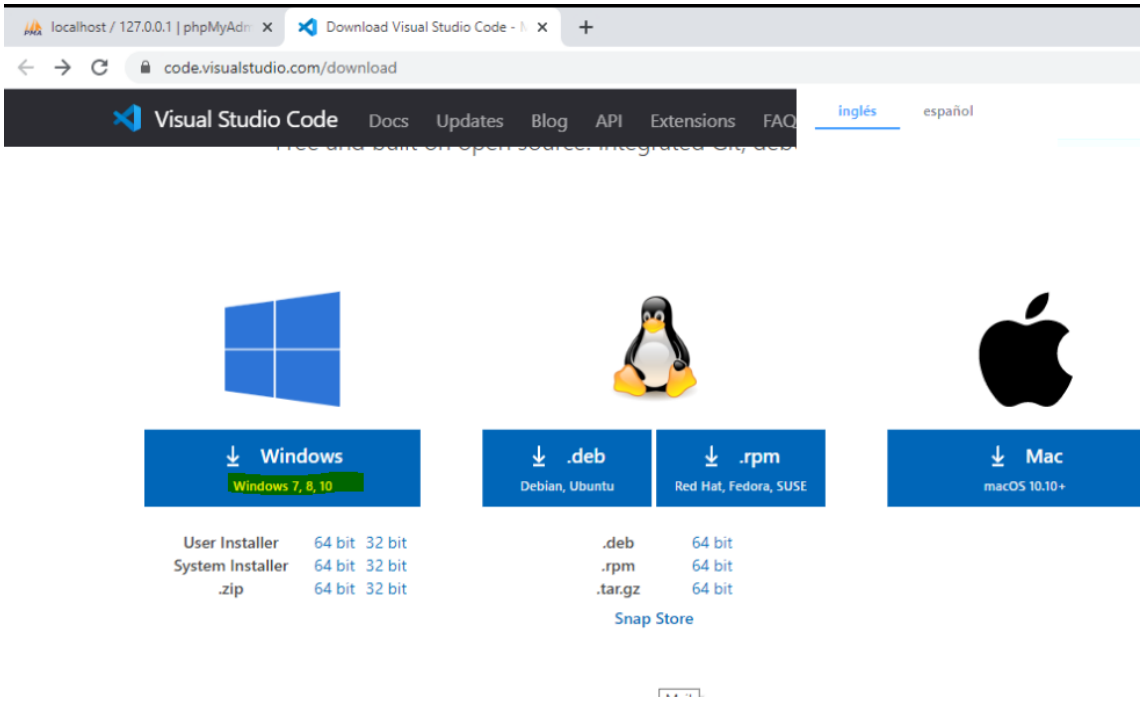
Descargamos el instalador de la página oficial de apachefriends.

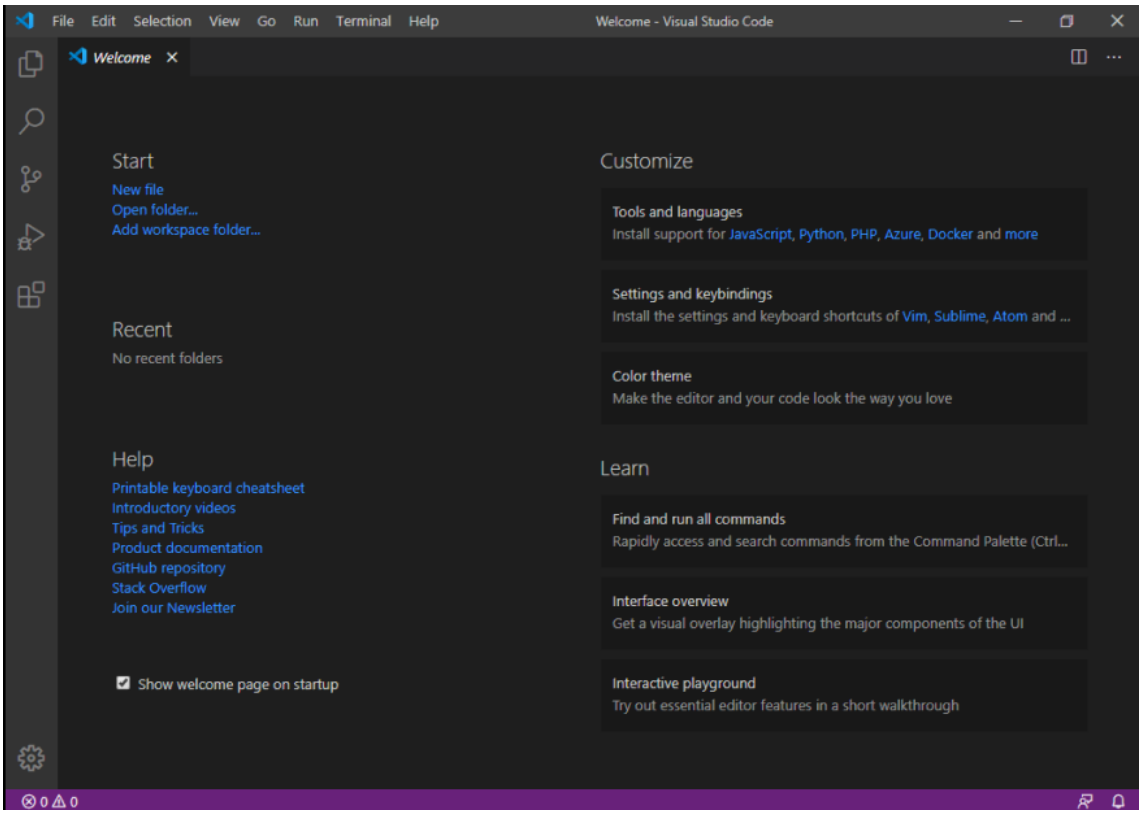






INSTALAR VISUAL STUDIO CODE





PHPMixBill

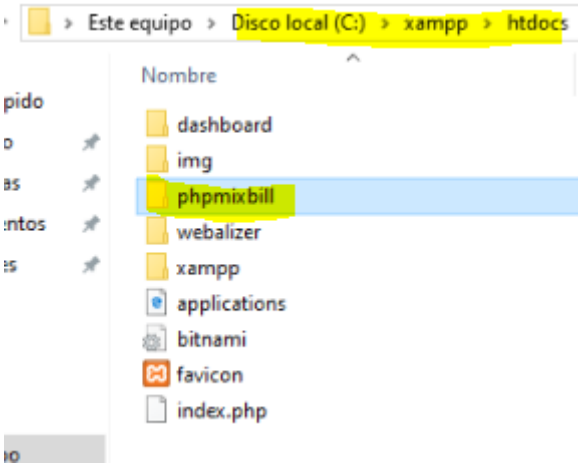
¿Qué es PHPMixBill?

PHPMixBill es el administrador de API MikroTik desarrollado en PHP para administrar MikroTik Hotspot, servidor PPPOE, velocidad de ancho de banda, monitor y usuarios. Es una GUI de API web PHP para administrar múltiples enrutadores MikroTik desde la misma interfaz web.

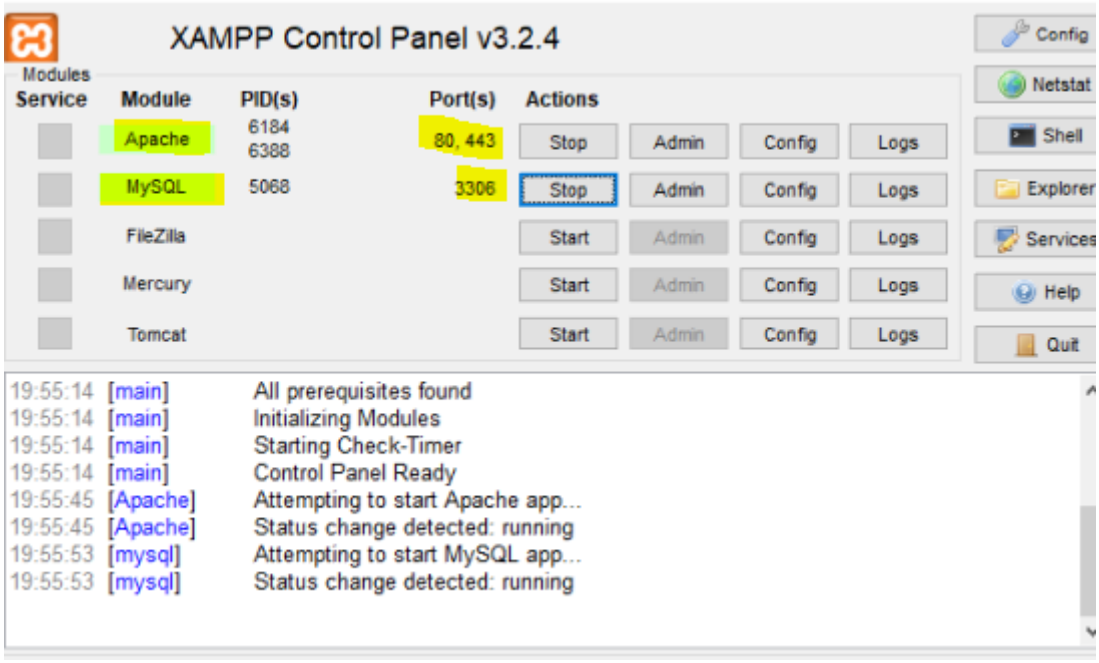
INSTALACIÓN

Descargamos el archivo phpmixbill

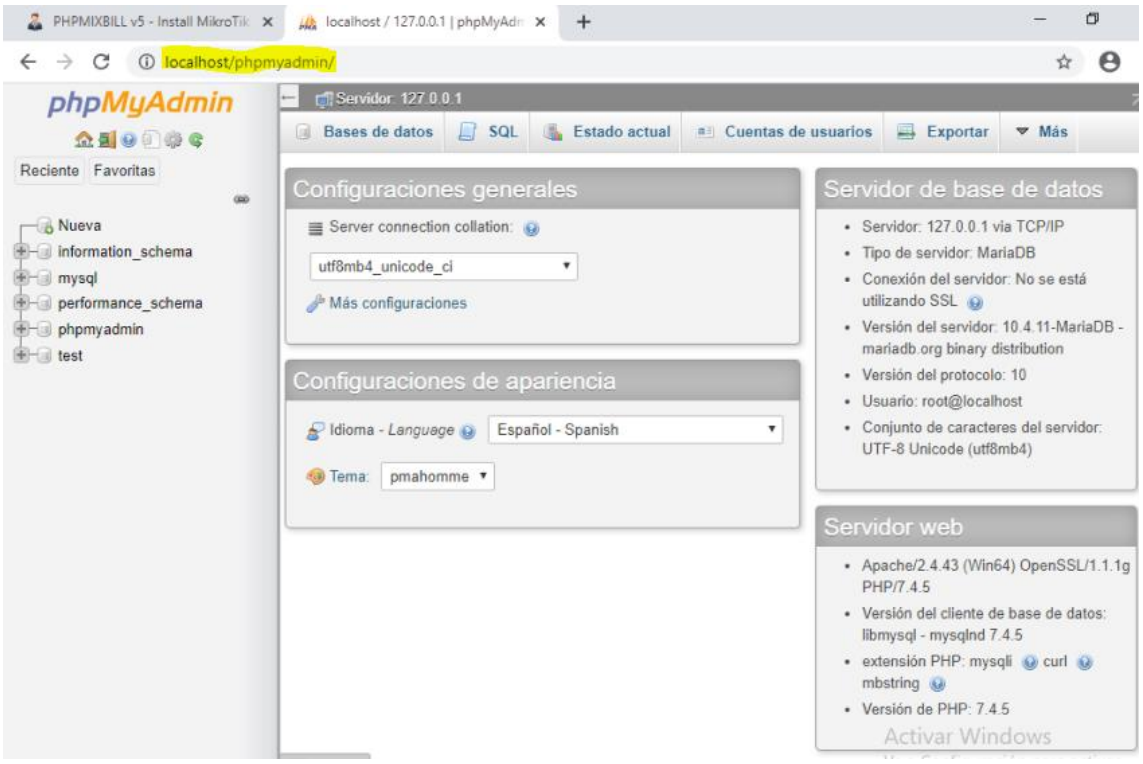
Extraemos los archivos y copiamos la carpeta extraída en C:/xampp/htdocs y le cambiamos el nombre a phpmixbill



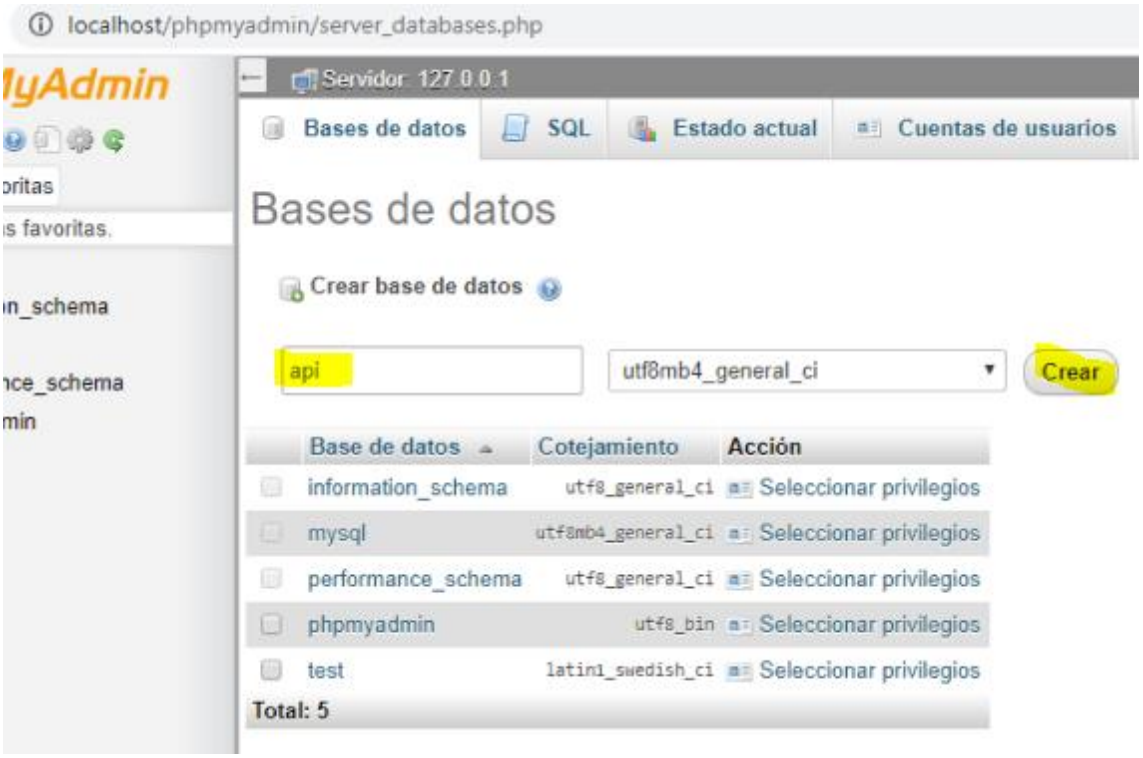
Iniciamos apache y MySQL en XAMPP



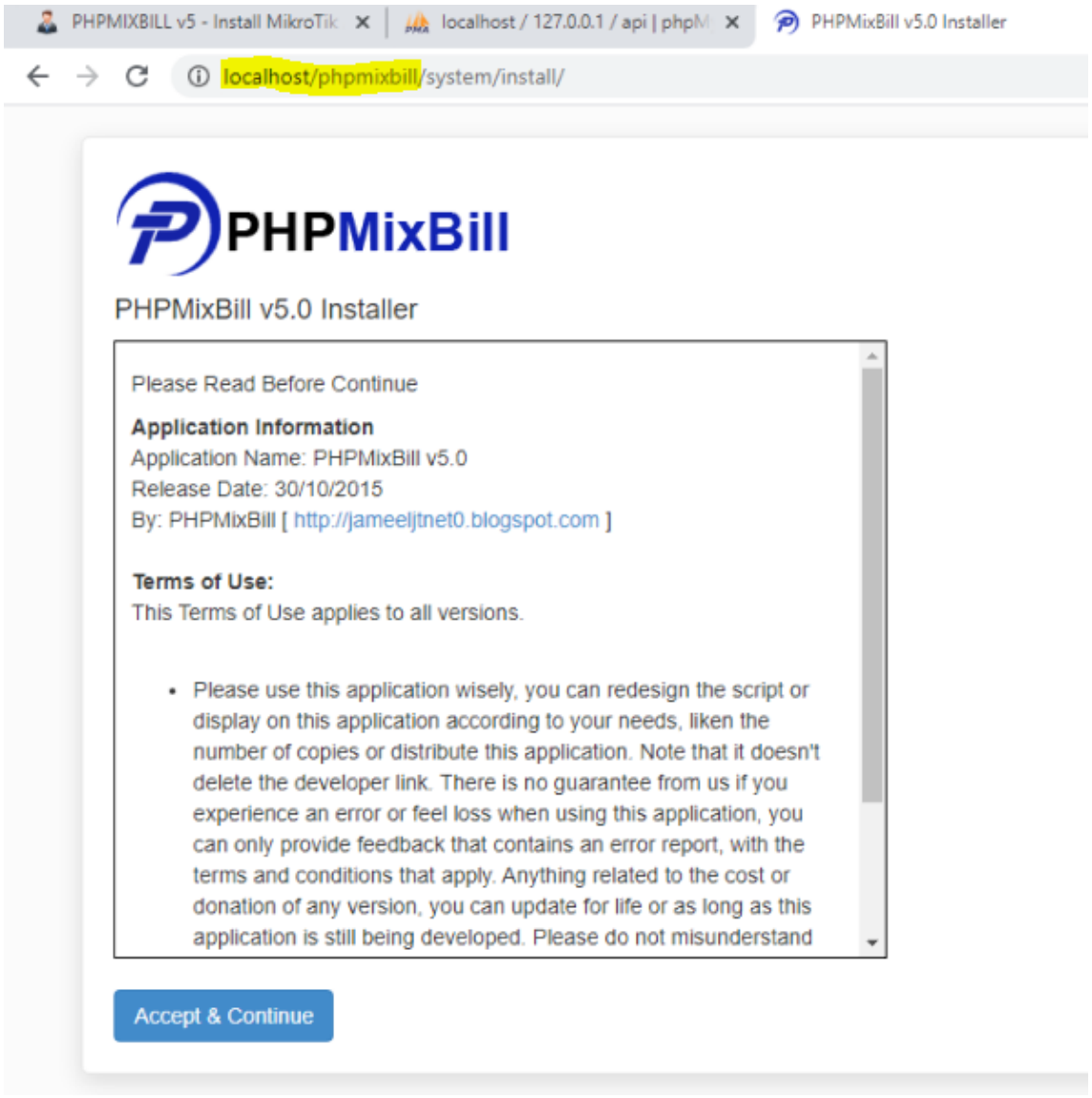
Entramos con Chrome en phpMyAdmin



Creamos una nueva base de datos

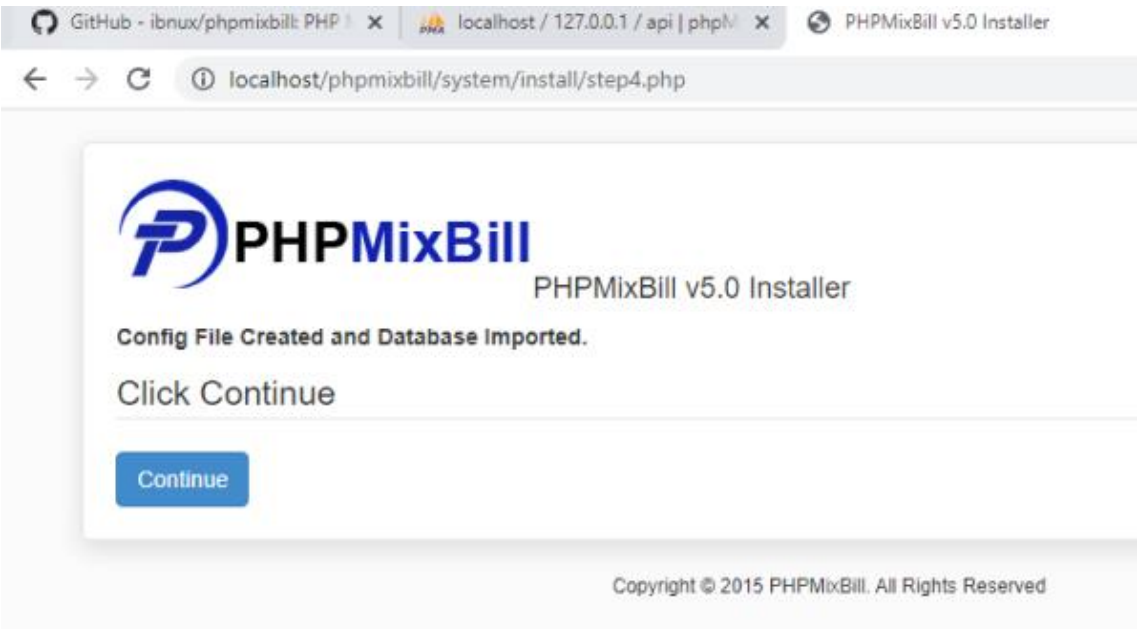


Vamos al navegador y escribimos localhost/’nombre_de_la_carpeta’ , en mi caso seria localhost/phpmixbill



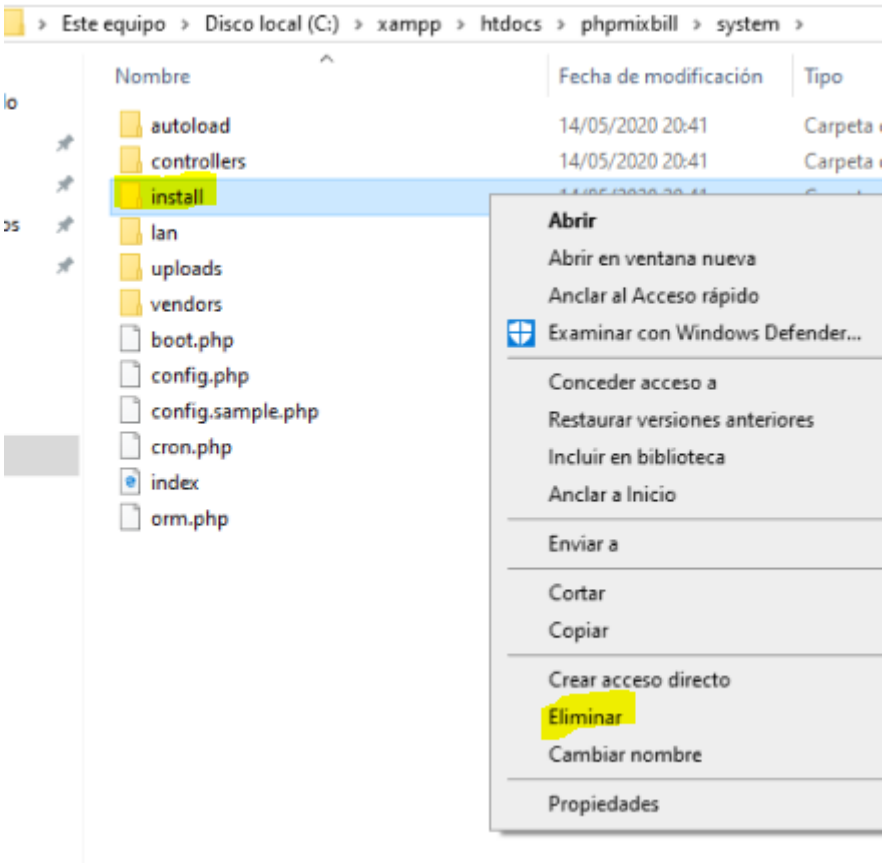
Nos dirige a la página para la instalación y rellenamos los campos con los datos de cada uno. En mi caso la base de datos la he llamado api, mi usuario es root y no tiene contraseña.


PHPMixBill
PHPMixBill v5.0 Installer
Database Connection & Site config
Application URL
http://localhost/phpmixbill
Application url without trailing slash at the end of url (e.g. http://172.16.10.10). Plea
Database Host
localhost
Database Username
root
Database Password
Database Name
api
Submit

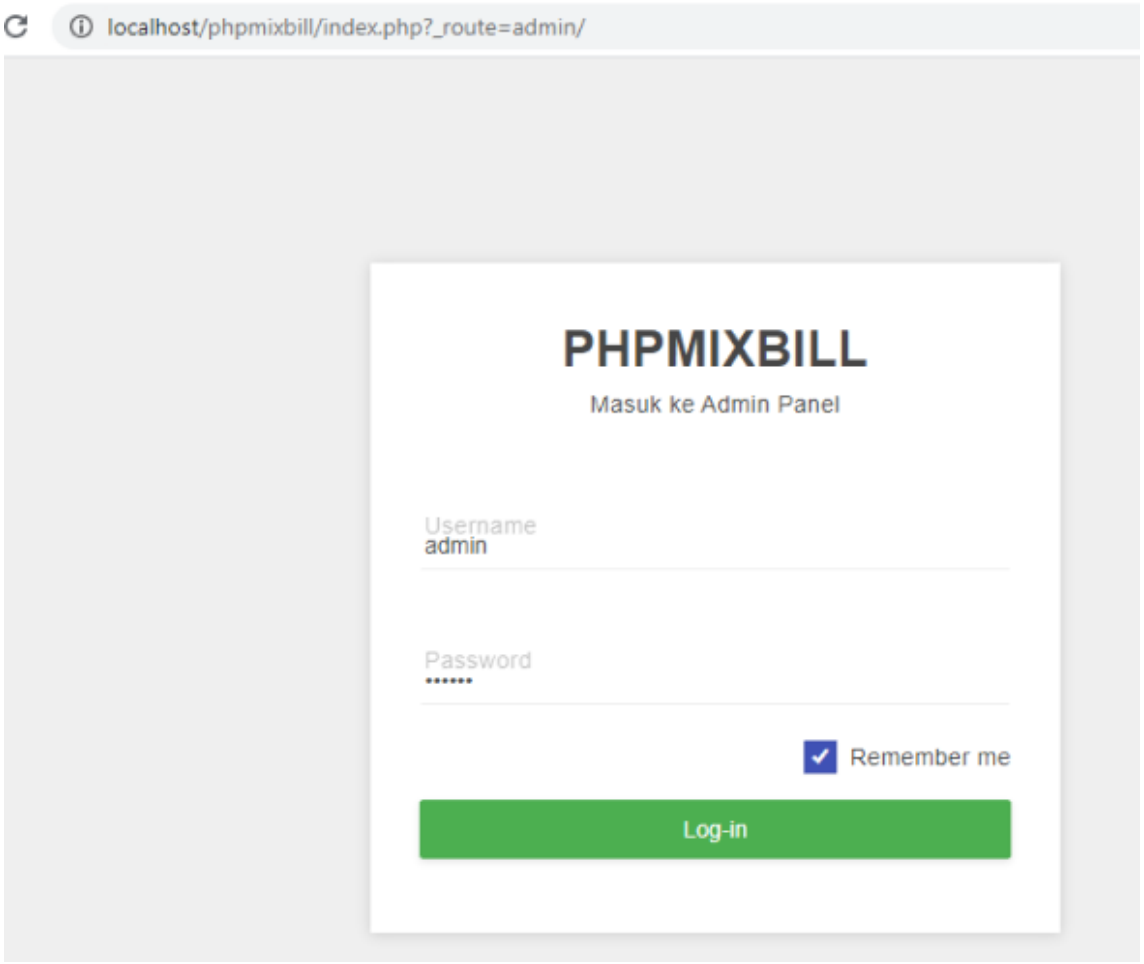

GitHub - ibnux/phpmixbill: PHP
localhost / 127.0.0.1 / api | phpM
PHPMixBill v5.0 Installer
localhost/phpmixbill/system/install/step4.php
PHPMixBill
PHPMixBill v5.0 Installer
Config File Created and Database Imported.
Click Continue
Continue
Copyright © 2015 PHPMixBill. All Rights Reserved

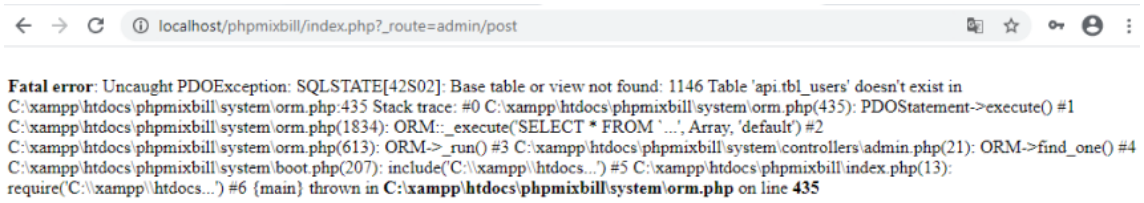
Estas son nuestras credenciales, debemos guardarlas o echarle captura

A continuación, borramos el archivo de instalación.

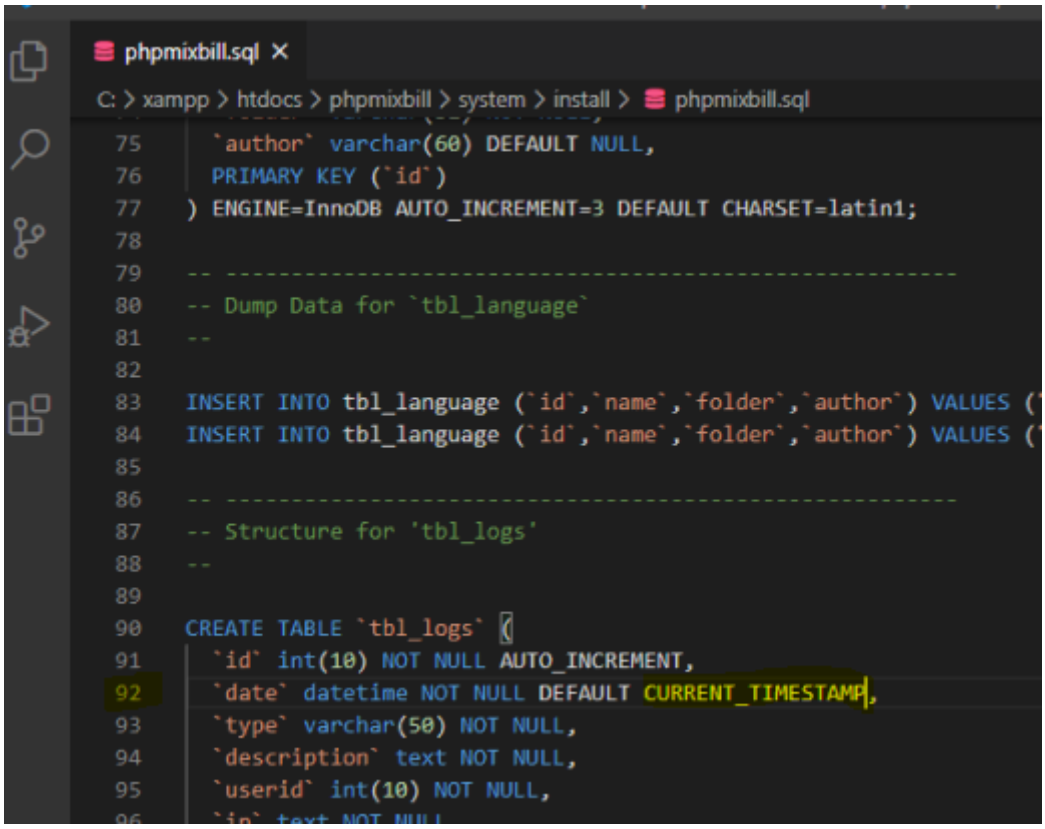


Nos logeamos usando el enlace que nos proporciona y las credenciales (captura anterior)



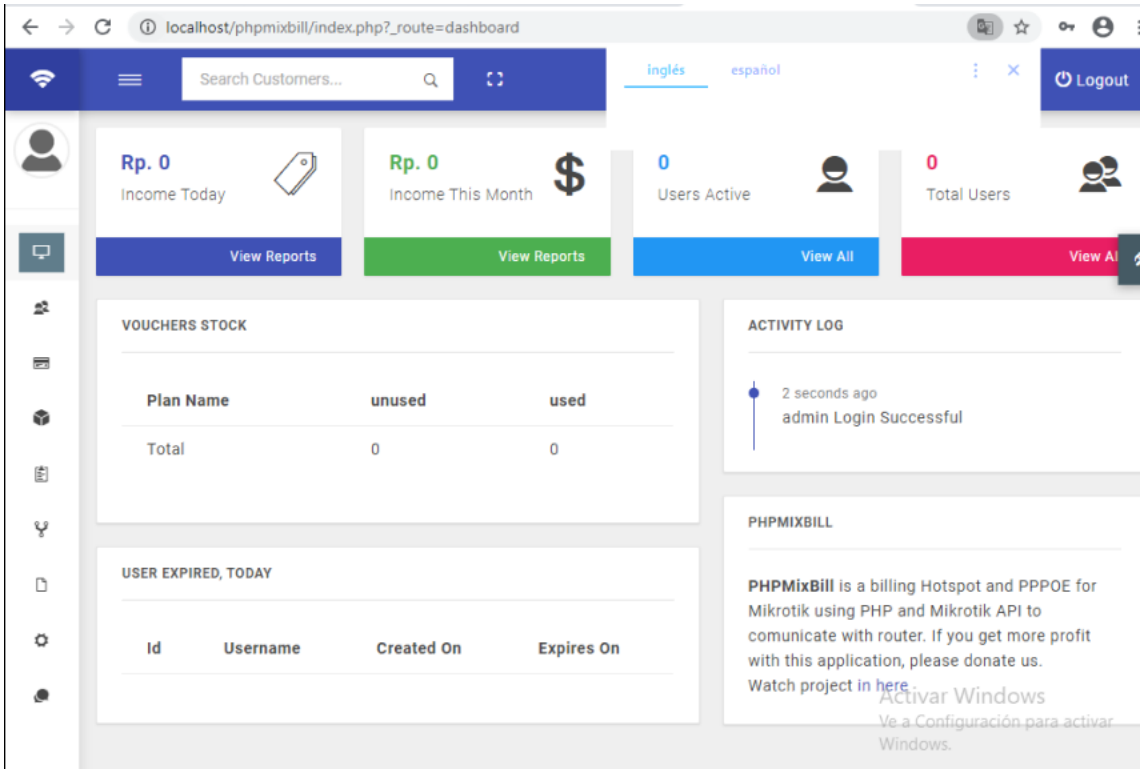


Como vemos al iniciar sesión da error, para arreglarlo en el archivo C:\xampp\htdocs\phpmixbill\system\install\phpmixbill.sql, en la línea 92 cambiamos '0000-00-00 00:00:00' por CURRENT_TIMESTAMP

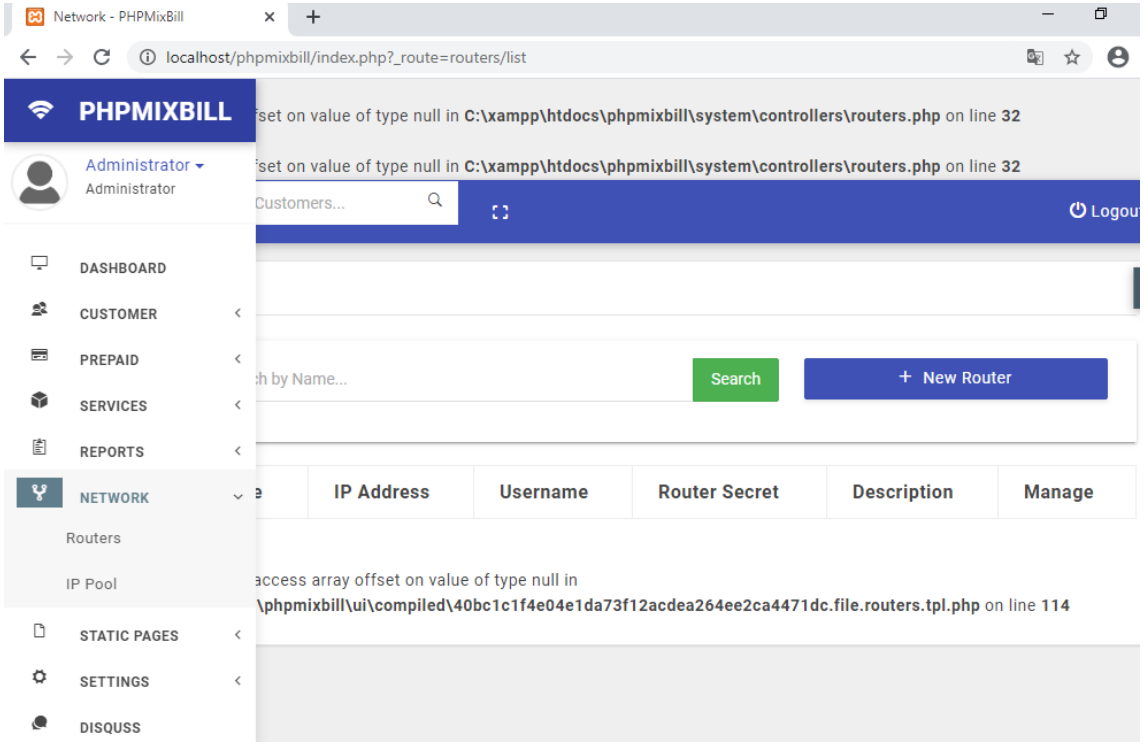


Una vez solucionado tenemos que borrar todo e instalar de nuevo el phpmixbill (lo recomendable es usar máquinas virtuales y debemos hacer snapshots)

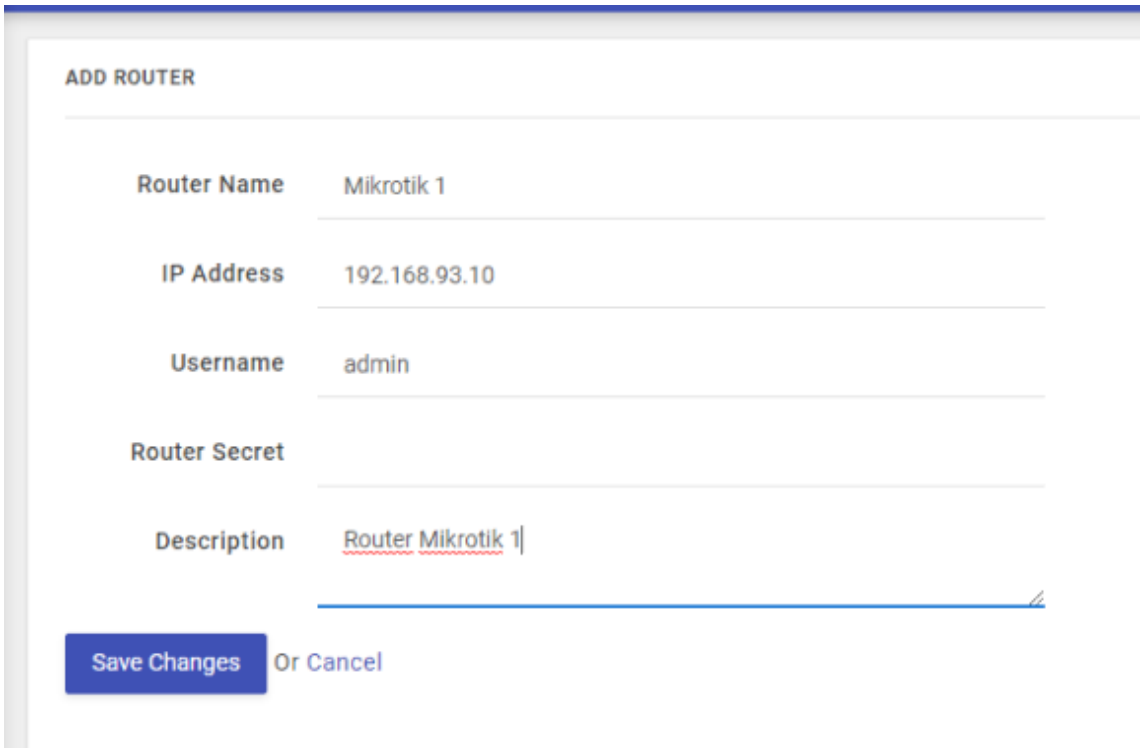
Y ya está arreglado



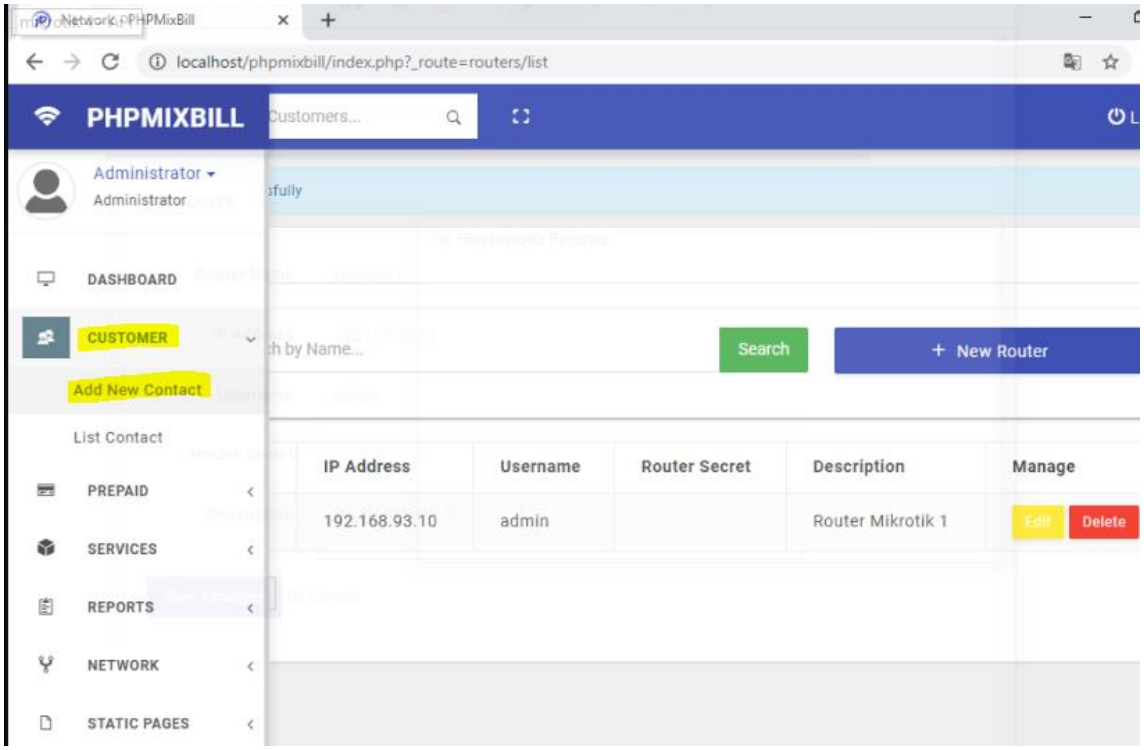
Conexión de Mikrotik con PHPMixBill

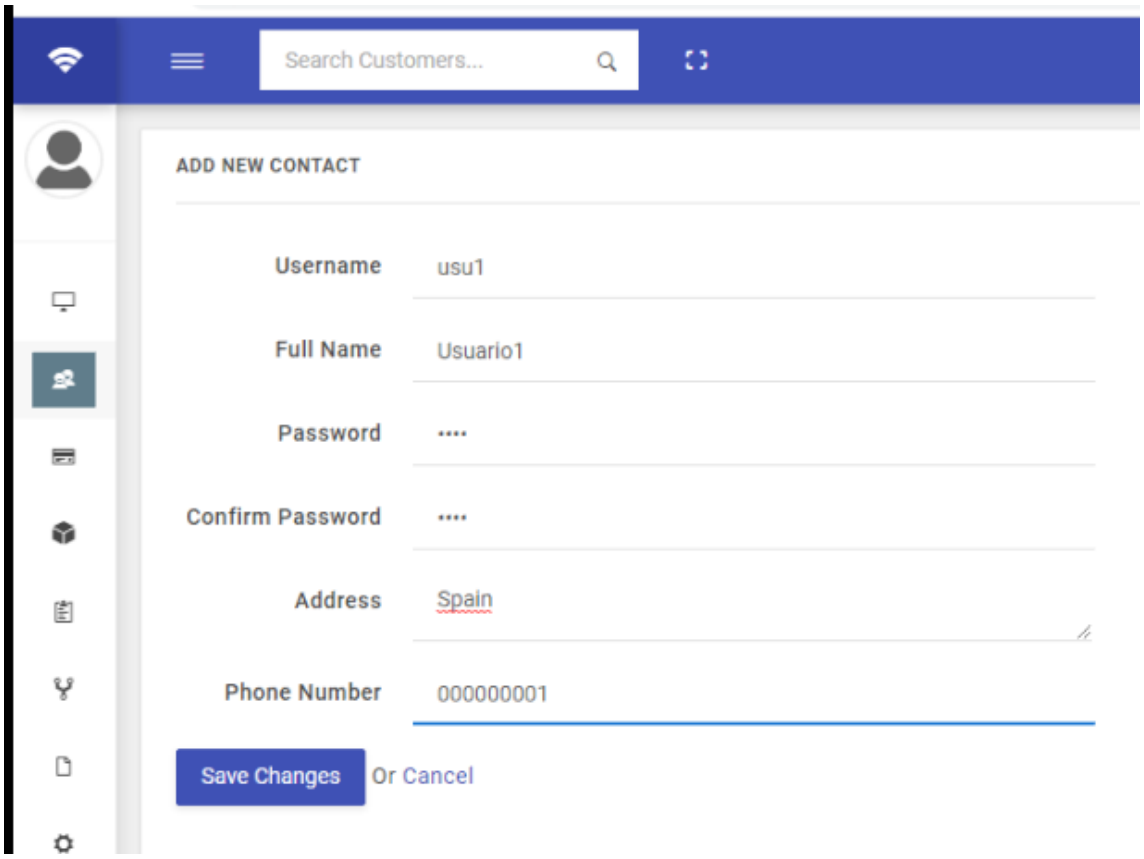


En network pinchamos en routers, luego en new router



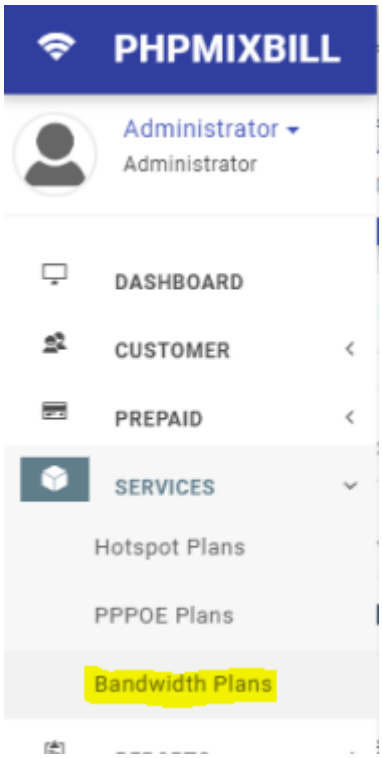
Añadir usuarios para que se conecten al mikrotik



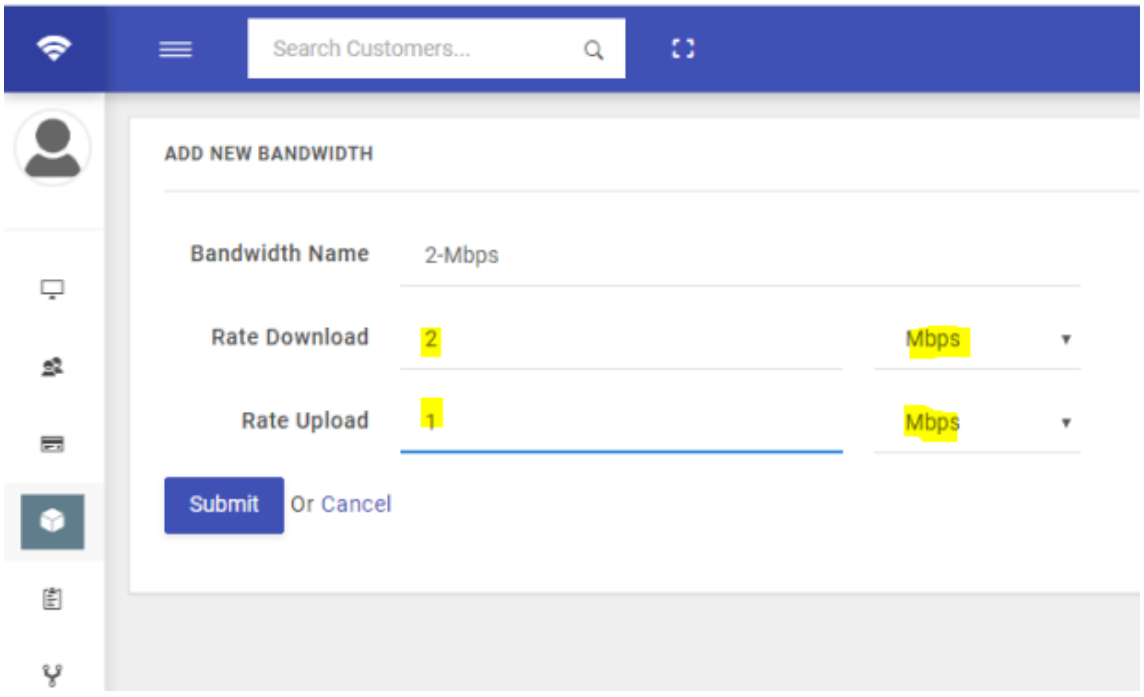


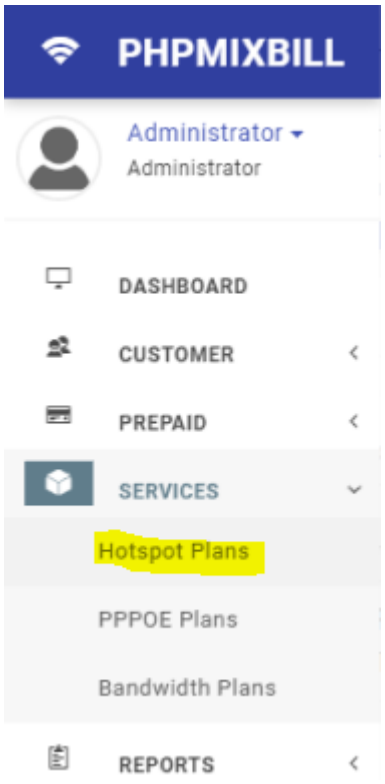
La contraseña es 1234

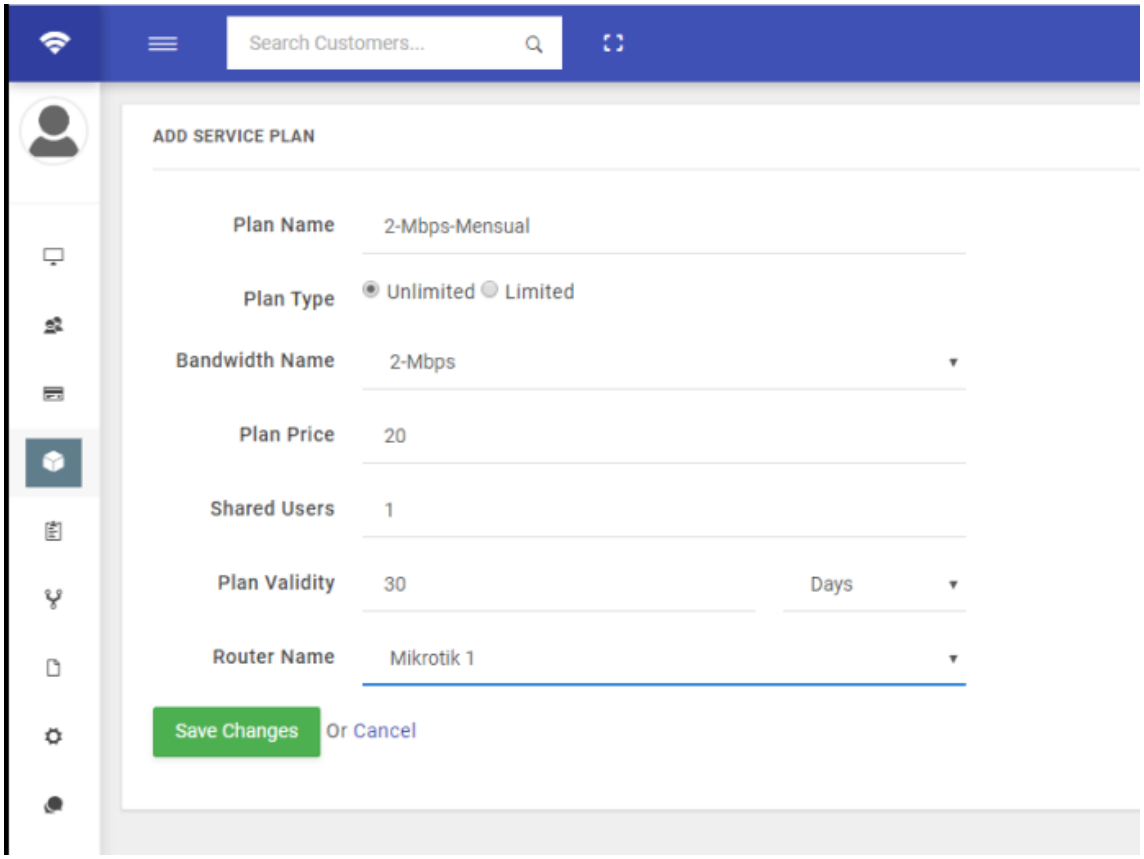
Creación de un plan de datos Hotspot con PHPMixBill

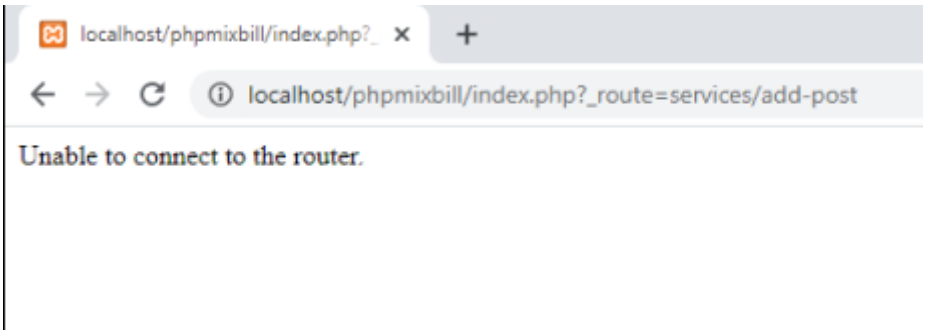


Limitaremos la conexión a 2 megas cada segundo de descarga, y a 1mb de subida



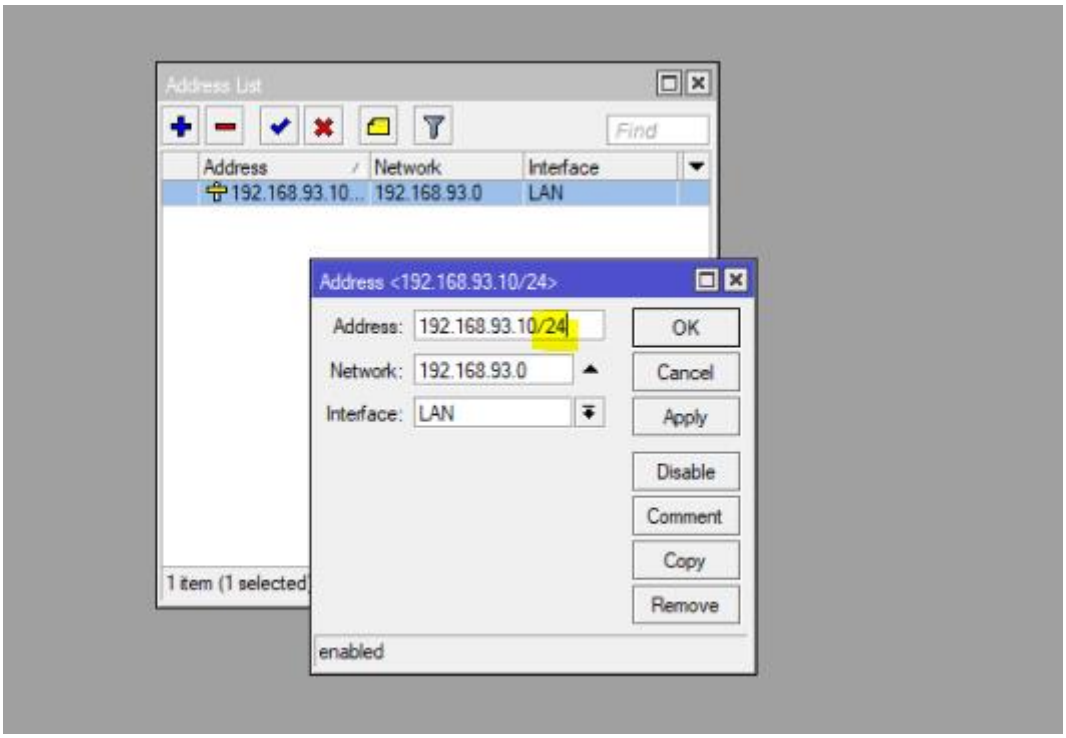

PHPMIXBILL
Administrator
Administrator
DASHBOARD
CUSTOMER
PREPAID
SERVICES
Hotspot Plans
PPPOE Plans
Bandwidth Plans
REPORTS


Search Customers...
ADD SERVICE PLAN
Plan Name
2-Mbps-Mensual
Plan Type
Unlimited
Limited
Bandwidth Name
2-Mbps
Plan Price
20
Shared Users
1
Plan Validity
30
Days
Router Name
Mikrotik 1
Save Changes
Or Cancel



Da error de conexión con el Mikrotik

Después de probar muchas cosas, encontré que el error era que la ip de la tarjeta de red no tenía el /24 y no permitía la conexión correctamente.



Debido a que la librería es muy antigua, genera fallos que no permiten la conexión con el Mikrotik. Por eso dejo este apartado por si en un futuro diera tiempo terminarlo y arreglar la librería del PHPMixBill.

MIKROTIK API PHP

Vamos a usar la API y su librería para hacer pruebas con Mikrotik, sacar información, reiniciar sistema, crear usuarios, etc.

Mostrar todas las tarjetas de re de Mikrotik

routeros_api.class.php | example1.php

C: > xampp > htdocs > api > examples > example1.php

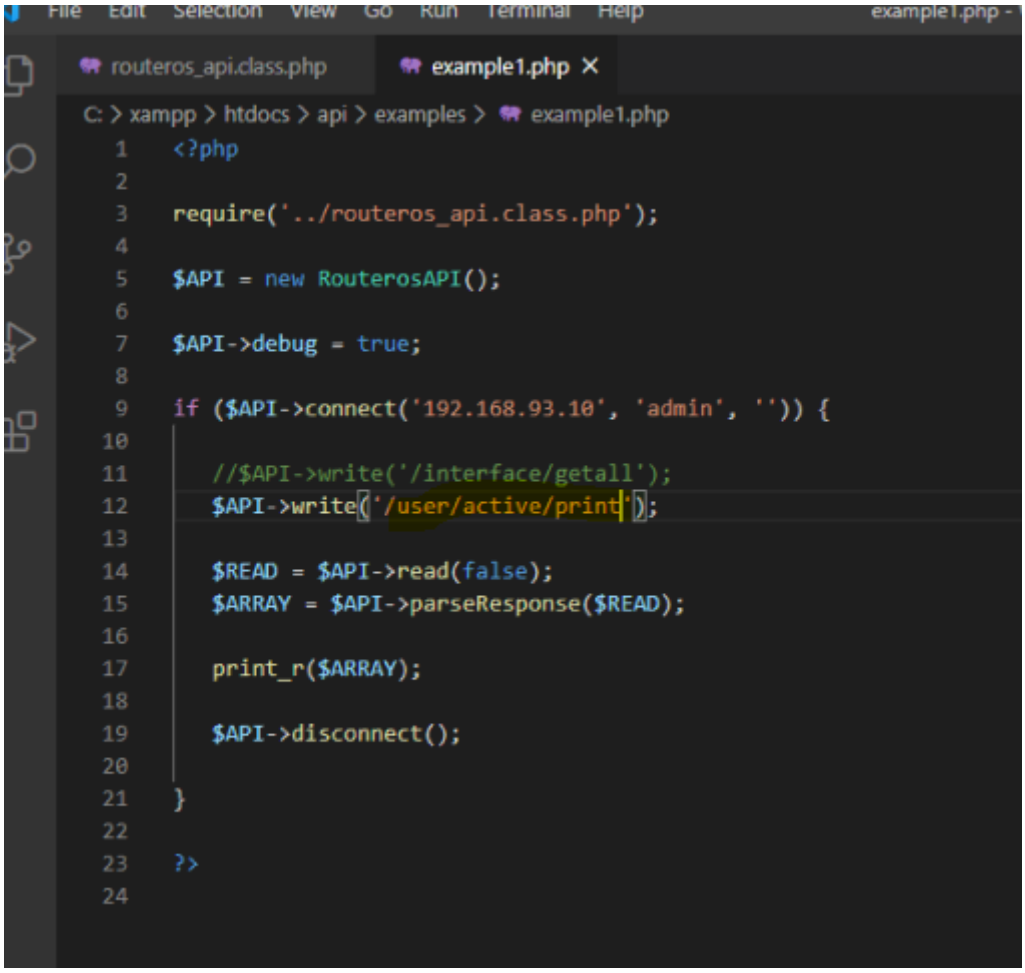
```php
<?php

require('../routeros_api.class.php');

$API = new RouterosAPI();

$API->debug = true;

if ($API->connect('192.168.93.10', 'admin', '')) {

   $API->write('/interface/getall');


   $READ = $API->read(false);
   $ARRAY = $API->parseResponse($READ);

   print_r($ARRAY);

   $API->disconnect();

}

?>
```

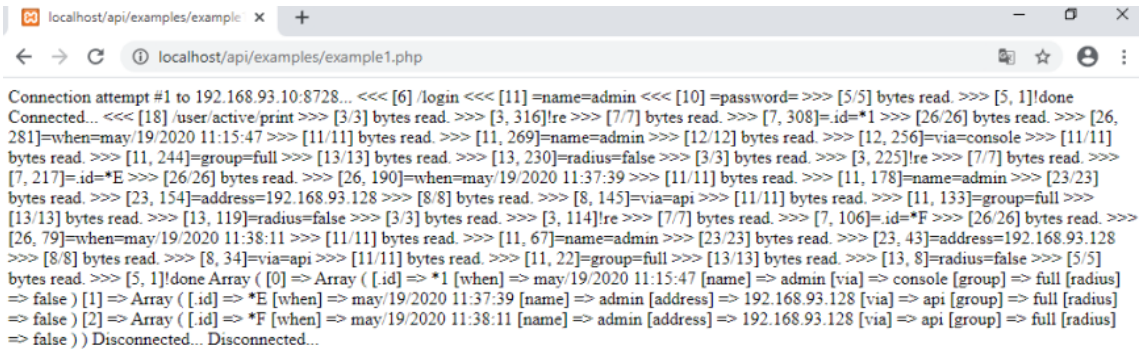
localhost/api/examples/example1.php

Connection attempt #1 to 192.168.93.10:8728... <<< [6] /login <<< [11] =name=admin <<< [10] =password= >>> [5/5] bytes read. >>> [5, 1]!done Connected... <<< [17] /interface/getall >>> [3/3] bytes read. >>> [3, 380]!re >>> [7/7] bytes read. >>> [7, 372]=.id=*1 >>> [9/9] bytes read. >>> [9, 362]=name=LAN >>> [20/20] bytes read. >>> [20, 341]=default-name=ether1 >>> [11/11] bytes read. >>> [11, 329]=type=ether >>> [9/9] bytes read. >>> [9, 319]=mtu=1500 >>> [16/16] bytes read. >>> [16, 302]=actual-mtu=1500 >>> [30/30] bytes read. >>> [30, 271]=mac-address=00:0C:29:09:4B:A4 >>> [39/39] bytes read. >>> [39, 231]=last-link-up-time=may/19/2020 11:15:30 >>> [13/13] bytes read. >>> [13, 217]=link-downs=0 >>> [13/13] bytes read. >>> [13, 203]=rx-byte=9706 >>> [13/13] bytes read. >>> [13, 189]=tx-byte=7198 >>> [13/13] bytes read. >>> [13, 175]=rx-packet=90 >>> [13/13] bytes read. >>> [13, 161]=tx-packet=56 >>> [10/10] bytes read. >>> [10, 150]=rx-drop=0 >>> [10/10] bytes read. >>> [10, 139]=tx-drop=0 >>> [16/16] bytes read. >>> [16, 122]=tx-queue-drop=0 >>> [11/11] bytes read. >>> [11, 110]=rx-error=0 >>> [11/11] bytes read. >>> [11, 98]=tx-error=0 >>> [13/13] bytes read. >>> [13, 84]=fp-rx-byte=0 >>> [13/13] bytes read. >>> [13, 70]=fp-tx-byte=0 >>> [15/15] bytes read. >>> [15, 54]=fp-rx-packet=0 >>> [15/15] bytes read. >>> [15, 38]=fp-tx-packet=0 >>> [13/13] bytes read. >>> [13, 24]=running=true >>> [15/15] bytes read. >>> [15, 8]=disabled=false >>> [5/5] bytes read. >>> [5, 1]!done Array ([0] => Array ([.id] => *1 [name] => LAN [default-name] => ether1 [type] => ether [mtu] => 1500 [actual-mtu] => 1500 [mac-address] => 00:0C:29:09:4B:A4 [last-link-up-time] => may/19/2020 11:15:30 [link-downs] => 0 [rx-byte] => 9706 [tx-byte] => 7198 [rx-packet] => 90 [tx-packet] => 56 [rx-drop] => 0 [tx-drop] => 0 [tx-queue-drop] => 0 [rx-error] => 0 [tx-error] => 0 [fp-rx-byte] => 0 [fp-tx-byte] => 0 [fp-rx-packet] => 0 [fp-tx-packet] => 0 [running] => true [disabled] => false)) Disconnected... Disconnected...

Nos devuelve toda la información sobre las tarjetas de red que tiene el Mikrotik

Mostrar todos los usuarios activos





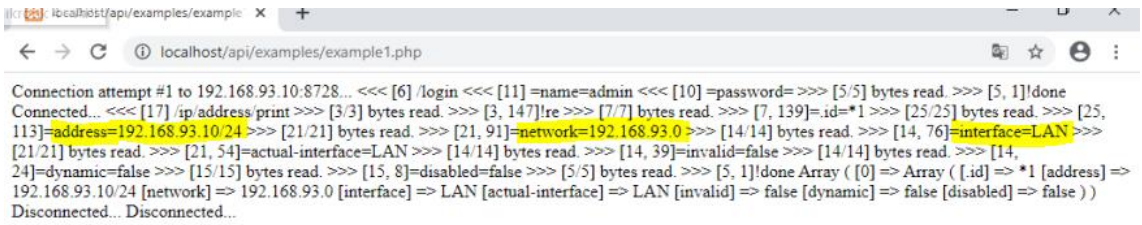
Nos muestra que solo esta admin conectado

Mostrar direcciones ip asignadas

routeros_api.class.php example1.php ×

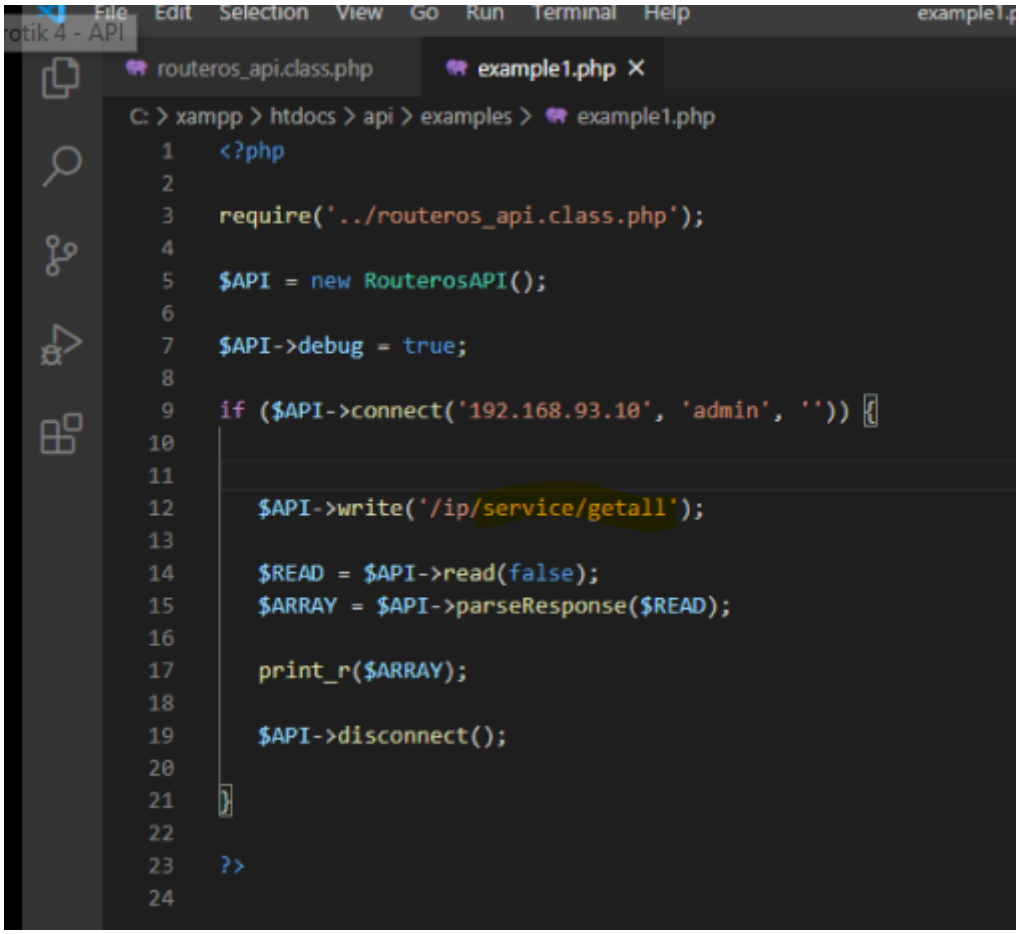
C: > xampp > htdocs > api > examples > example1.php

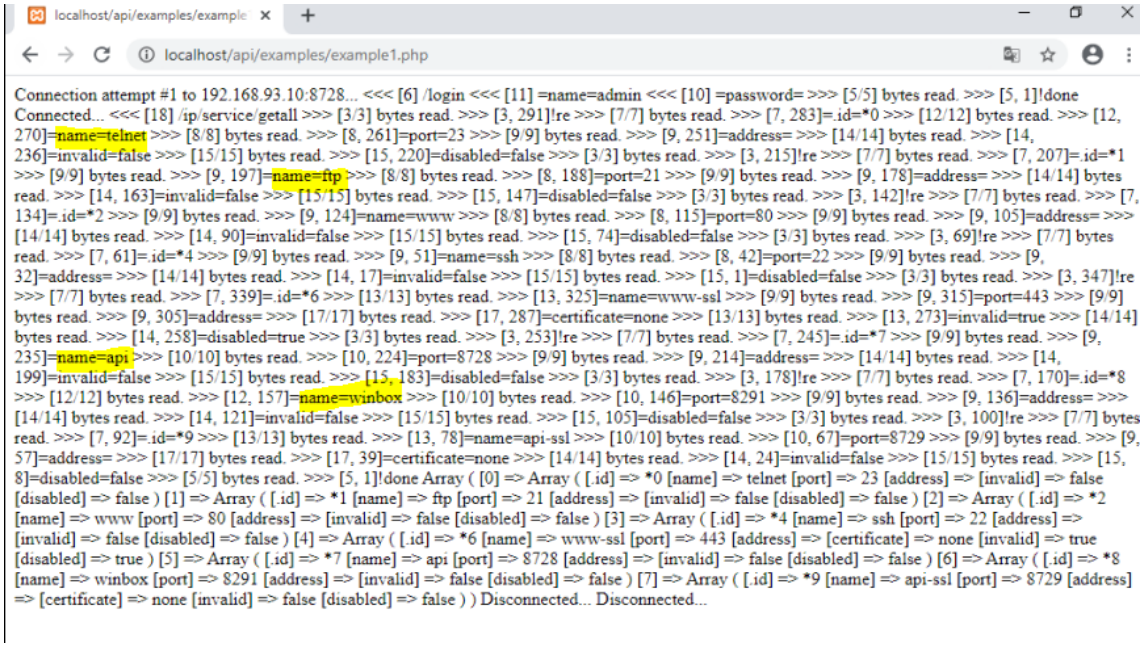
```php
<?php

require('../routeros_api.class.php');

$API = new RouterosAPI();

$API->debug = true;

if ($API->connect('192.168.93.10', 'admin', '')) {

   //$API->write('/interface/getall');
   $API->write('/ip/address/print');

   $READ = $API->read(false);
   $ARRAY = $API->parseResponse($READ);

   print_r($ARRAY);

   $API->disconnect();

}

?>
```



Nos muestra la dirección asignada(192.168.93.10), la red(192.168.93.0) y la tarjeta de red(LAN)

Mostrar todos los servicios de mikrotik

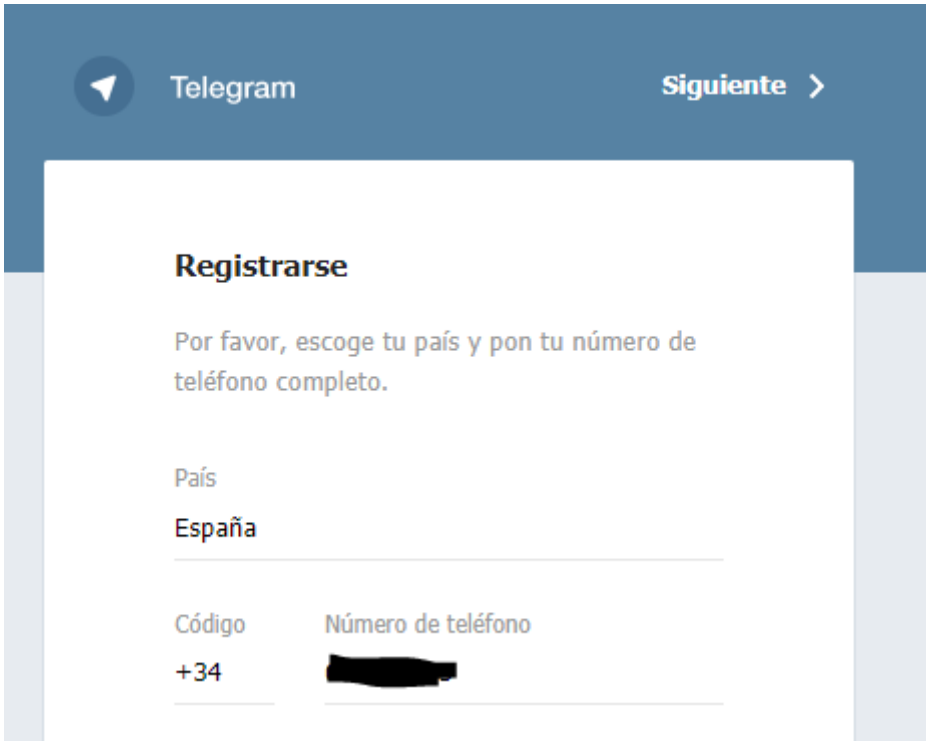




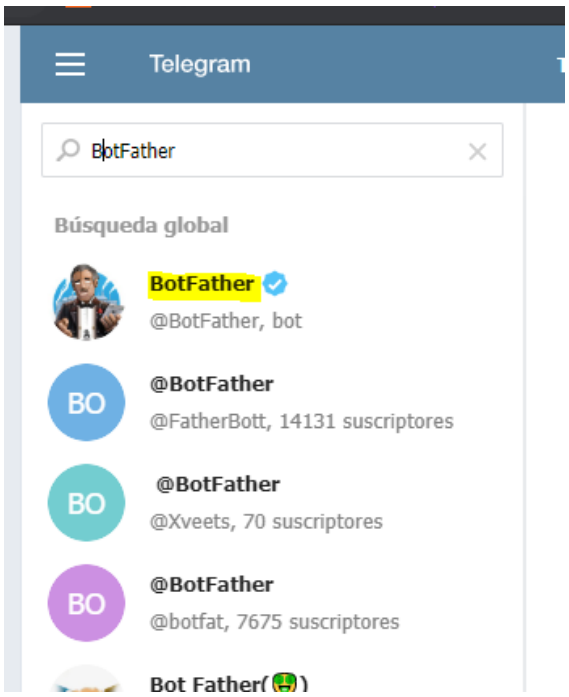
Como podemos ver esos son todos los servicios y la información correspondiente de cada uno.

BOT DE TELEGRAM

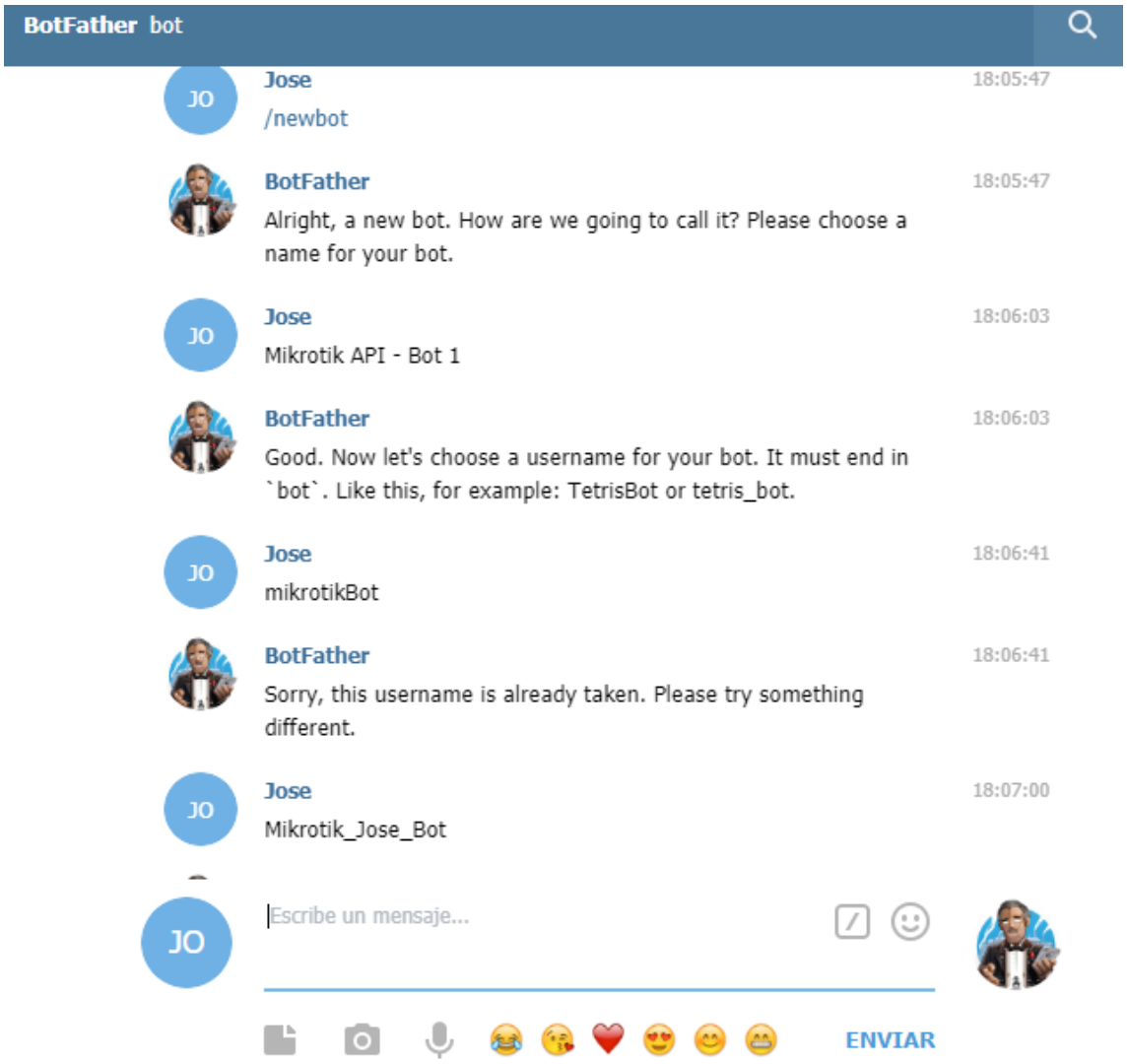
Primero debemos registrarnos en Telegram



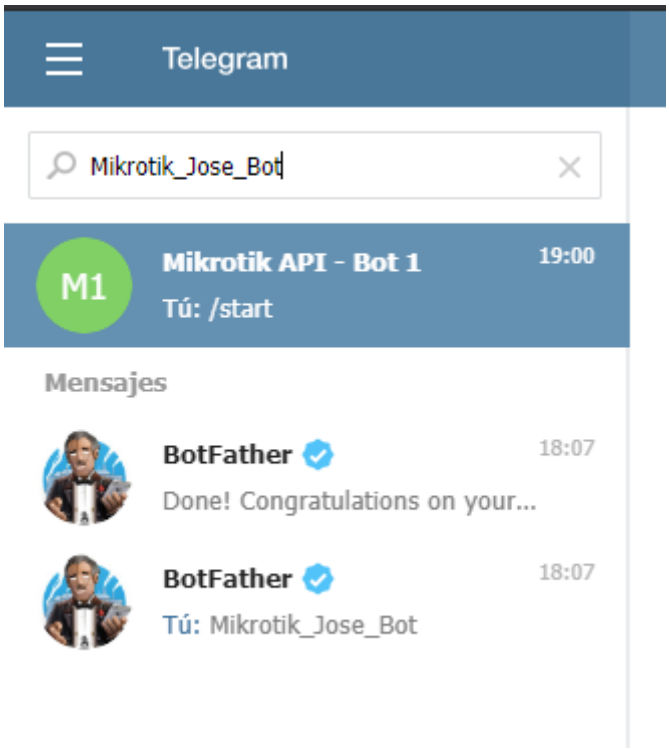
Una vez dentro buscamos el BotFather, con el gestionaremos los bots y podremos crear nuevos

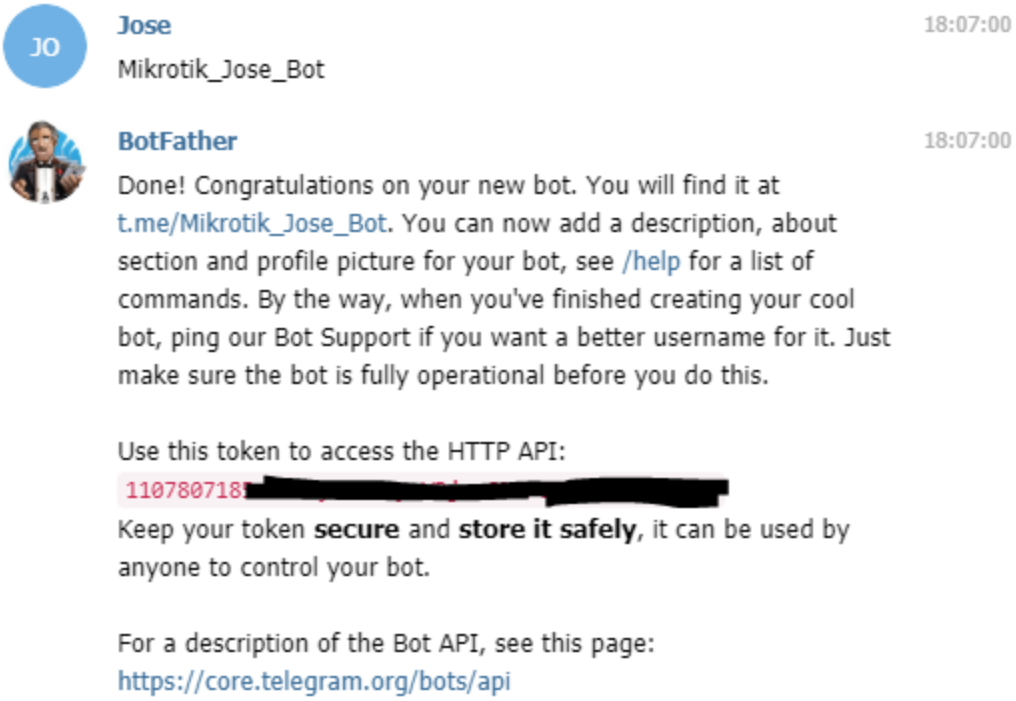


Crearemos un nuevo bot



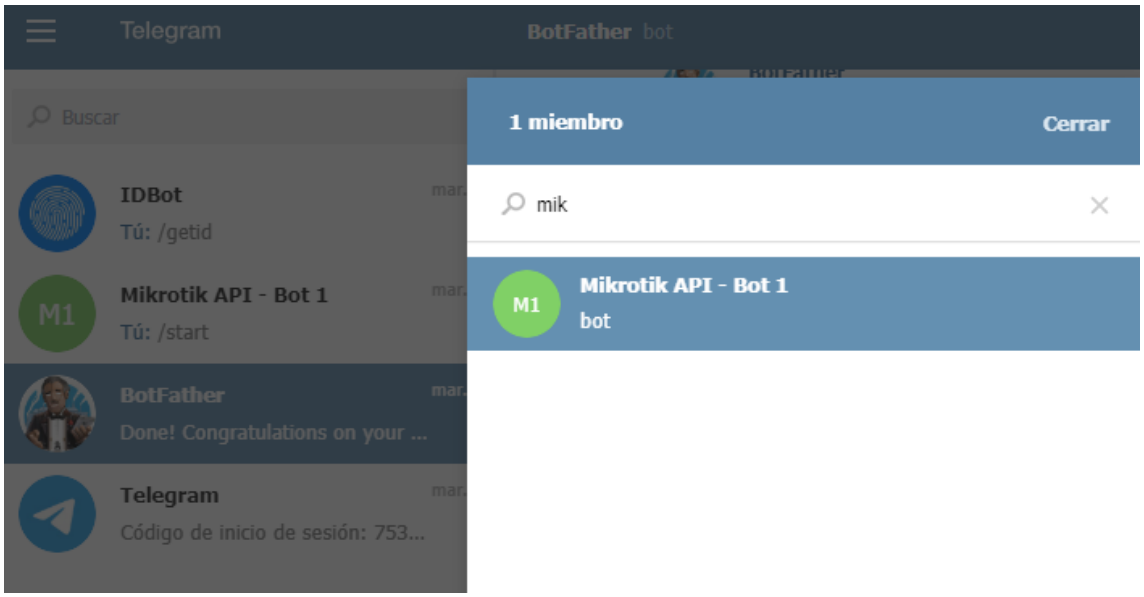
Buscamos al bot



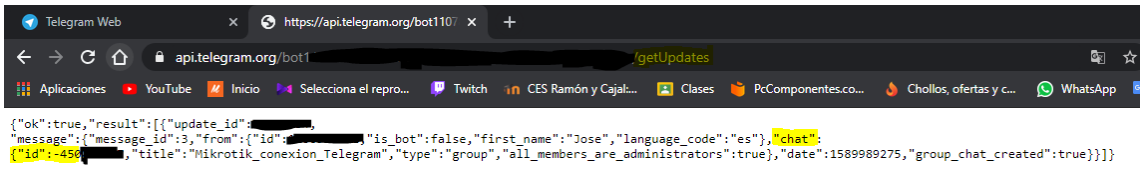


Ese token lo usaremos después

Creamos un grupo con el Bot de telegram

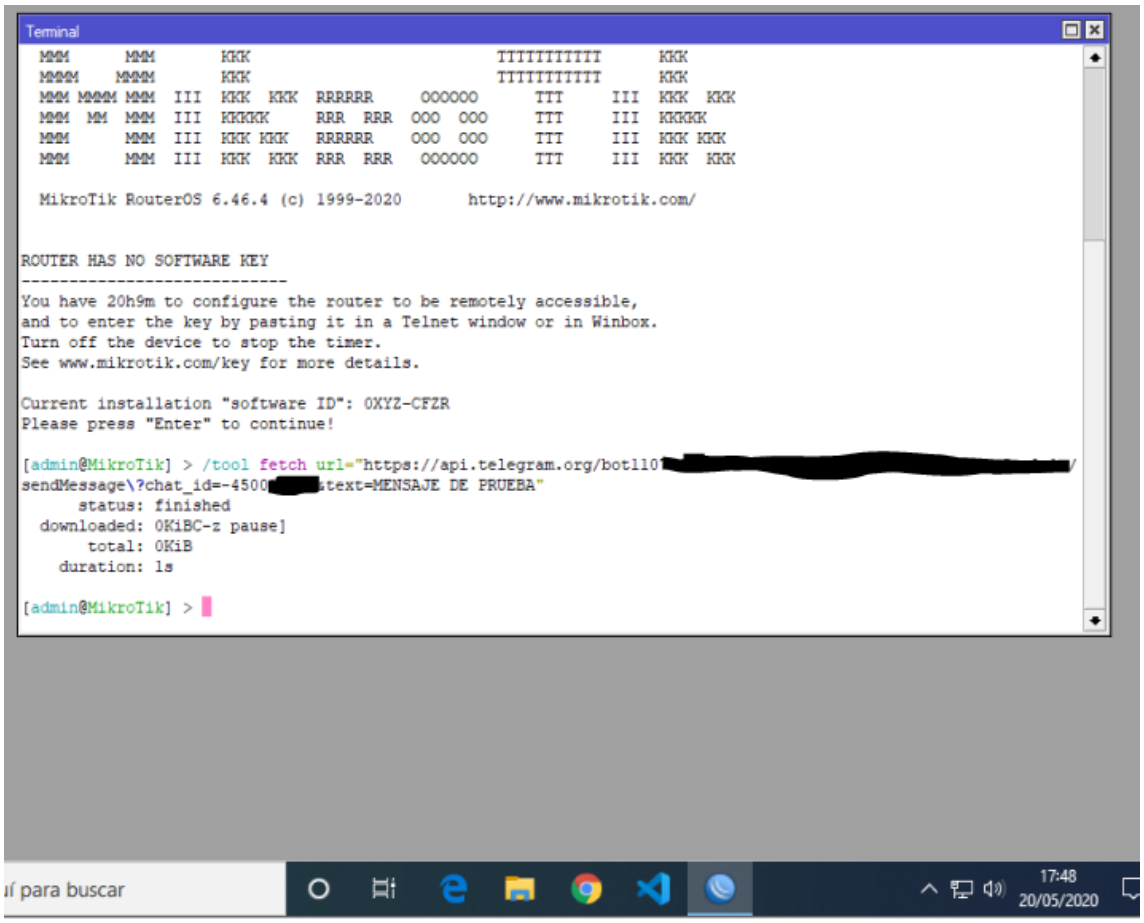


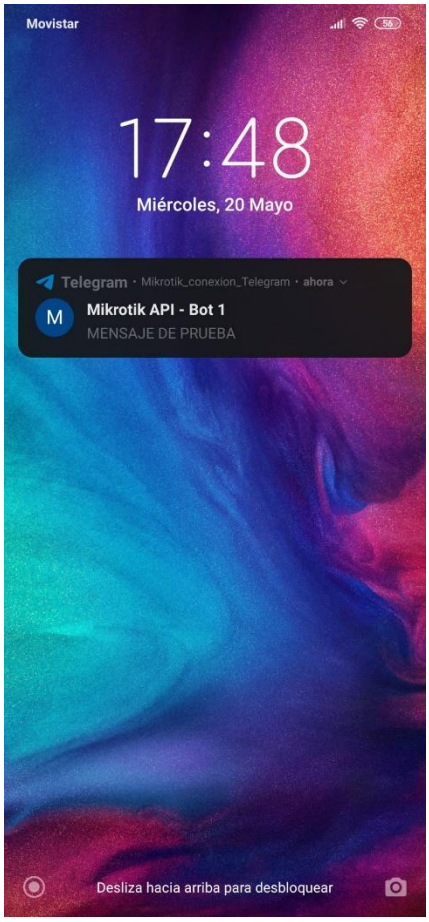
Ahora buscaremos el ID del chat, buscamos: https://api.telegram.org/bot(poner el token)/getUpdates



Vamos al Mikrotik y escribimos

/tool fetch url="https://api.telegram.org/bot(TOKEN)/sendMessage\?chat_id=(ID DEL CHAT)&text=MENSAJE DE PRUEBA"


Terminal
MMM MMM KKK TTTTTTTTTTT KKK
MMMM MMMM KKK TTTTTTTTTTT KKK
MMM MMMM MMM III KKK KKK RRRRRR OOOOOO TTT III KKK KKK
MMM MM MMM III KKKKK RRR RRR OOO OOO TTT III KKKKK
MMM MMM III KKK KKK RRRRRR OOO OOO TTT III KKK KKK
MMM MMM III KKK KKK RRR RRR OOOOOO TTT III KKK KKK
MikroTik RouterOS 6.46.4 (c) 1999-2020 http://www.mikrotik.com/
ROUTER HAS NO SOFTWARE KEY
You have 20h9m to configure the router to be remotely accessible,
and to enter the key by pasting it in a Telnet window or in Winbox.
Turn off the device to stop the timer.
See www.mikrotik.com/key for more details.
Current installation "software ID": 0XYZ-CFZR
Please press "Enter" to continue!
[admin@MikroTik] > /tool fetch url="https://api.telegram.org/bot110[redacted]/
sendMessage\?chat_id=-4500[redacted]text=MENSAJE DE PRUEBA"
status: finished
downloaded: 0KiB-z pause]
total: 0KiB
duration: ls
[admin@MikroTik] >
17:48
20/05/2020


Movistar
17:48
Miércoles, 20 Mayo
Telegram · Mikrotik_conexion_Telegram · ahora
Mikrotik API - Bot 1
MENSAJE DE PRUEBA
Desliza hacia arriba para desbloquear

Vemos que ha llegado el mensaje correctamente

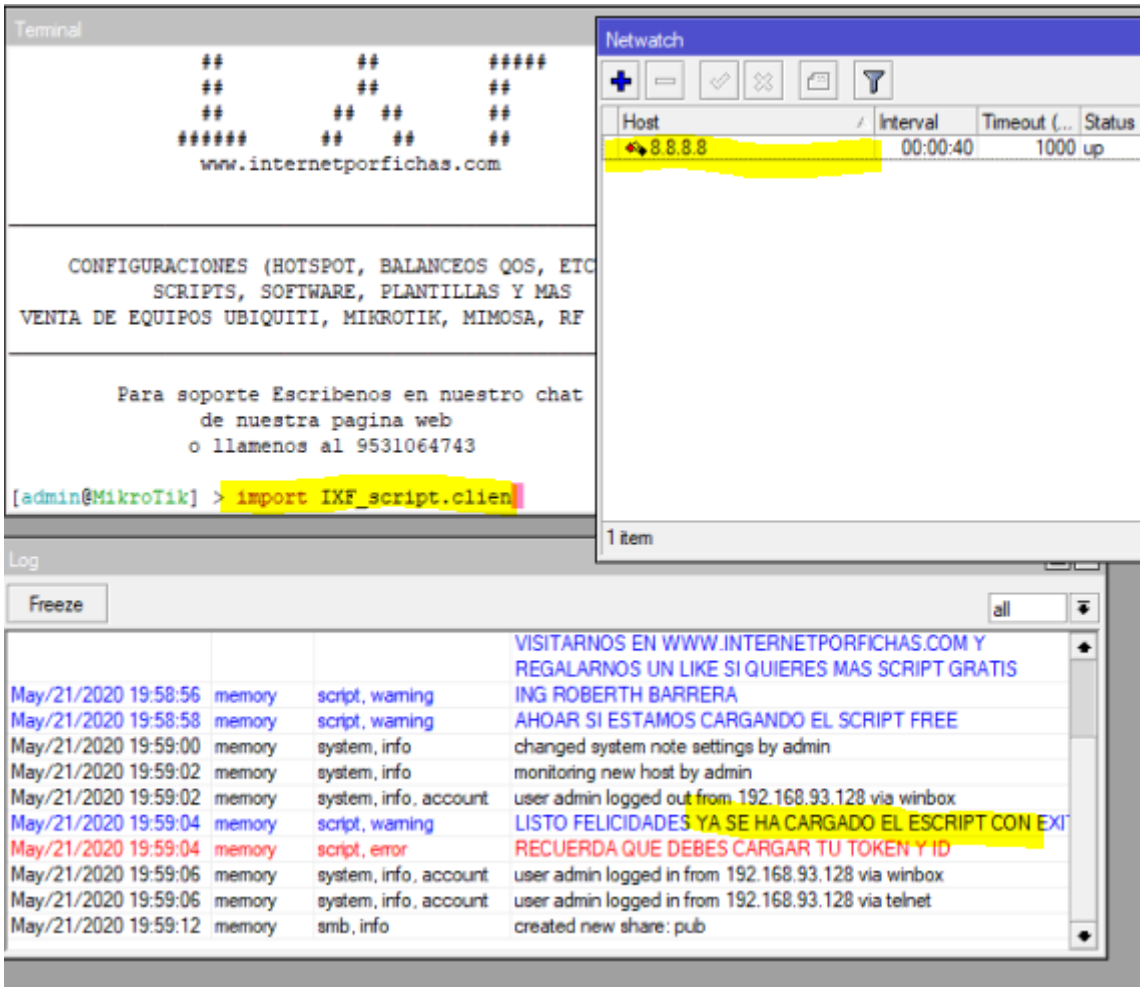
Programado de alertas

Ahora mediante un script programaremos el Mikrotik para que envié mensajes al grupo de telegram y nos notifique de las alertas

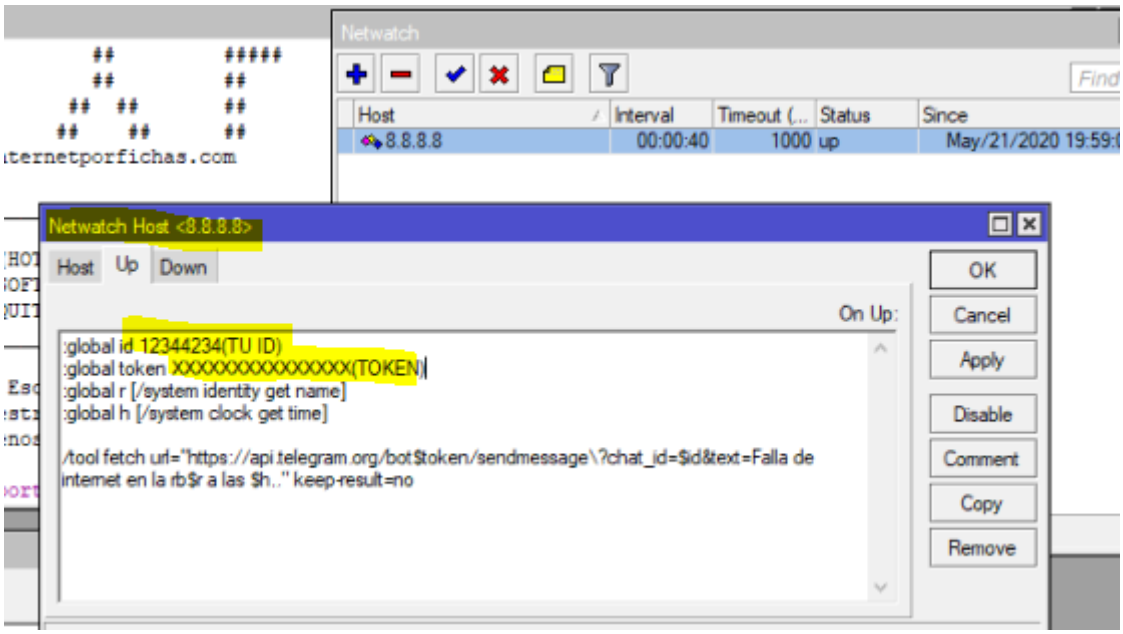
Primero nos descargamos el script desde http://rycdigitalworld.com/free.des/IXF/IXF_script.clien , este script envía una alerta cuando se queda sin internet el Mikrotik.

```
[admin@MikroTik] > /tool fetch url="http://rycdigitalworld.com/free.des/IXF/IXF_script.clien" mode=http
      status: finished
  downloaded: 2KiBC-z pause]
       total: 2KiB
    duration: 0s

[admin@MikroTik] >
```

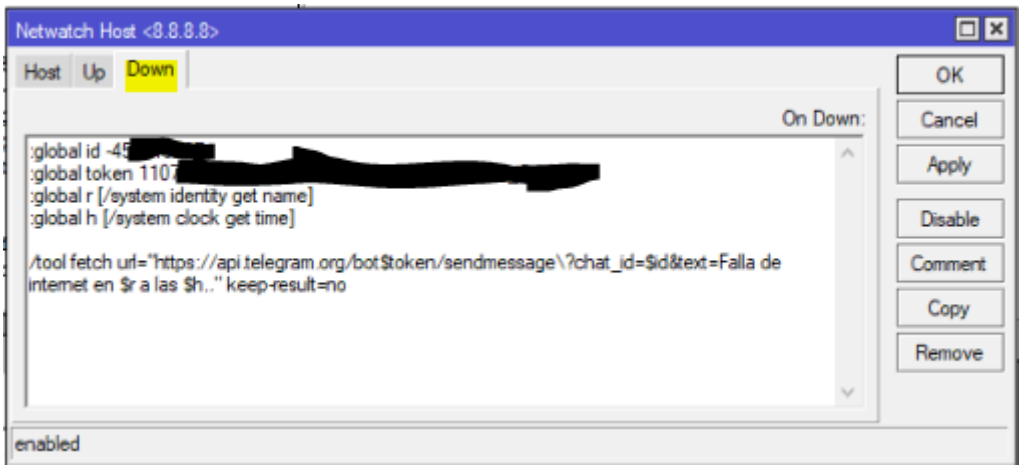
Abrimos el script usando ‘ import IXF_script.clien ’

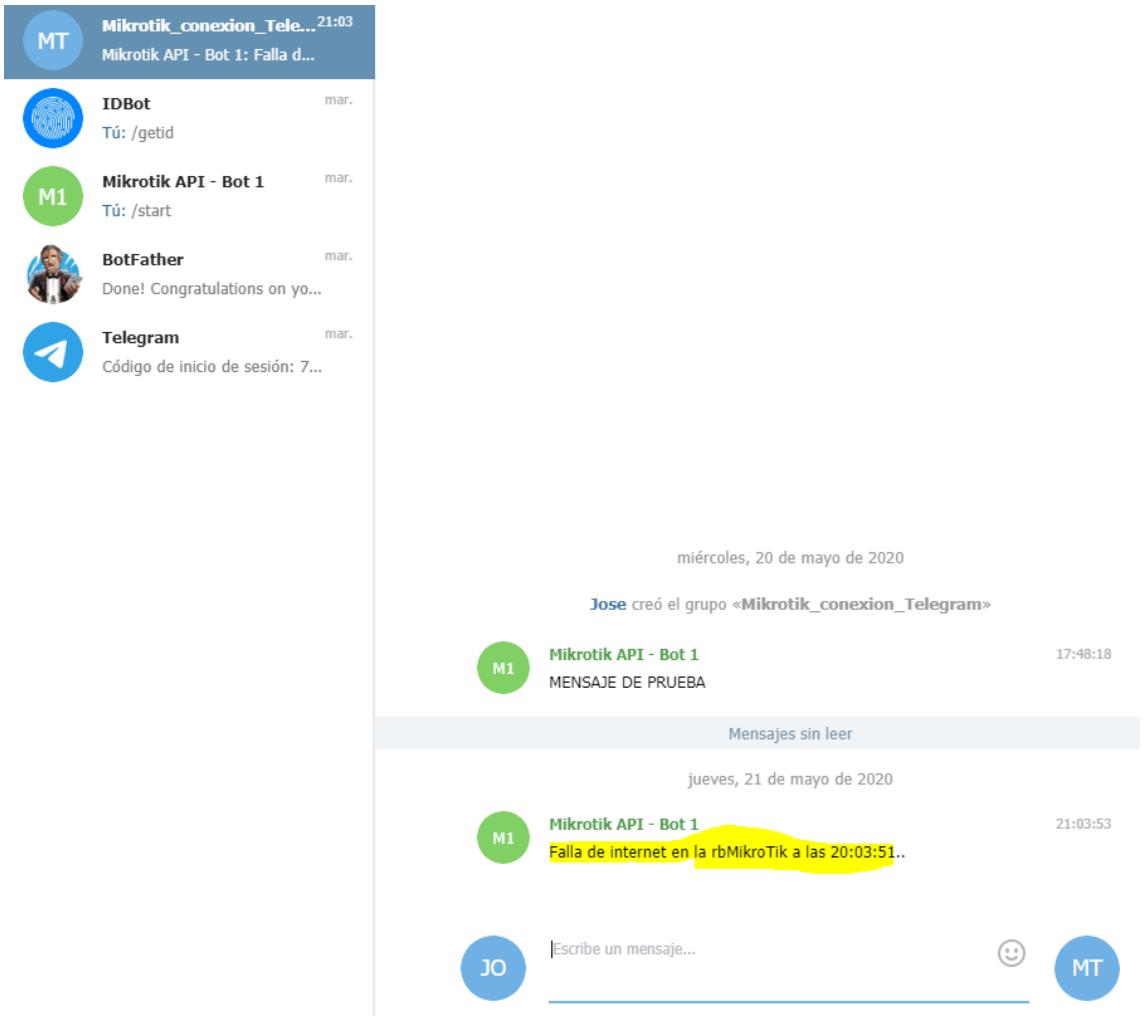


Ahora cambiamos el token y el ID, en netwatch



Copiamos la información en Down, así que cuando Mikrotik se cae envía el mensaje. Si lo dejamos en Up se envía una notificación cada vez que el Mikrotik se enciende.

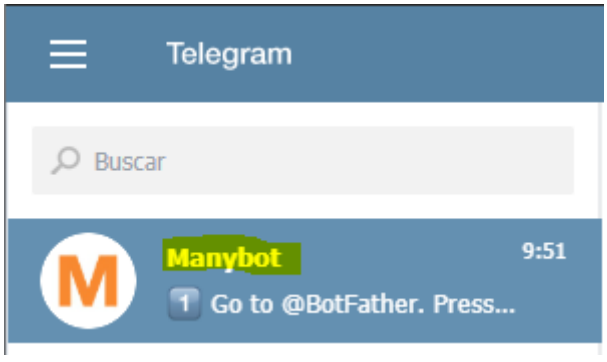




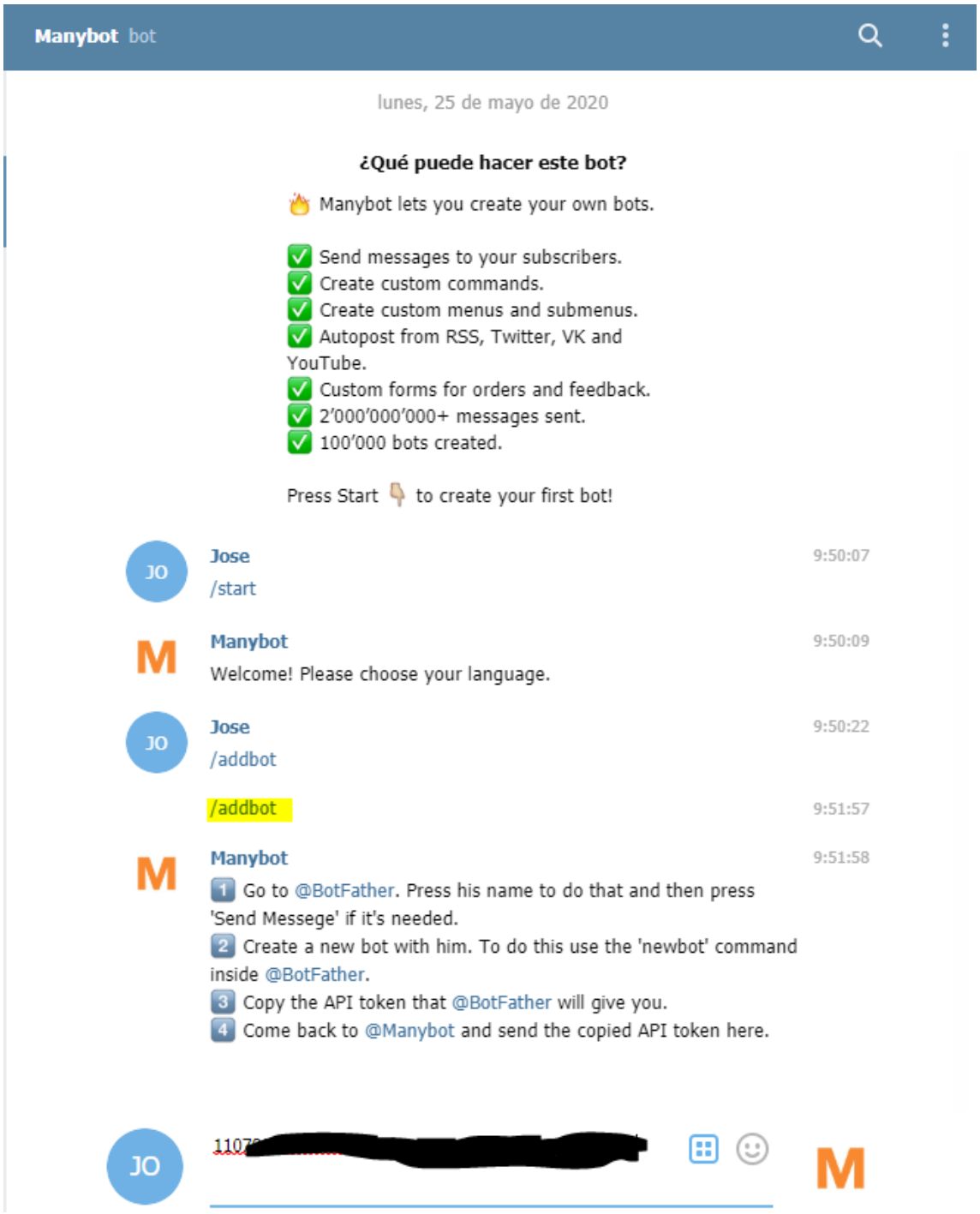
Vemos que funciona correctamente

CREAR COMANDOS PERSONALIZADOS EN EL BOT DE INSTAGRAM

Lo primero es buscar a ManyBot



Ahora añadimos el token del bot



Manybot

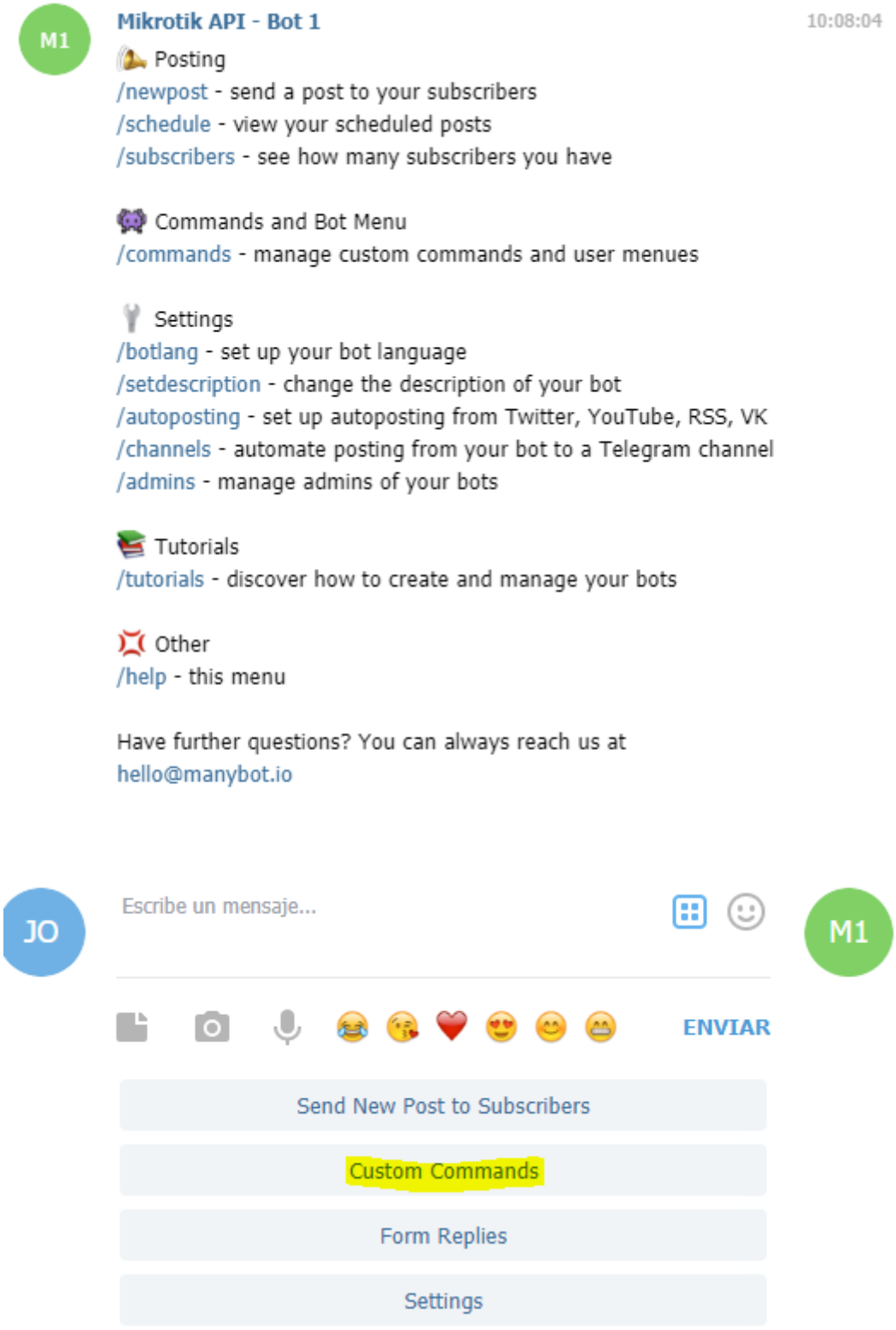
Token for @Mikrotik_Jose_Bot accepted!

Last step.

Write a few words about @Mikrotik_Jose_Bot. What does it do?

..or /skip

Vamos al setting del bot



Pulsaremos custom commands y luego créate command

JO **Jose** 10:19:23

Custom Commands

M1 **Mikrotik API - Bot 1** 10:19:25

You can create custom commands that your bot will reply to with predefined messages. Use the menu below to create new custom commands, change the look of the bot's menu or select a command to edit it.

JO **Jose** 10:19:25

Create Command

M1 **Mikrotik API - Bot 1** 10:19:27

Enter the command name. Please use only latin letters, numbers and '_'.

Some examples:
/website
/pricelist
/contacts
/best_music
/best_photos

JO **Jose** 10:19:32

/saludo

M1 **Mikrotik API - Bot 1** 10:19:33

Bot can reply with one or more messages to a custom command. You can use text, pictures, videos or any other file type.

Send everything that you want to add as a reply to this command and press 'Save'.

JO Escribe un mensaje... M1

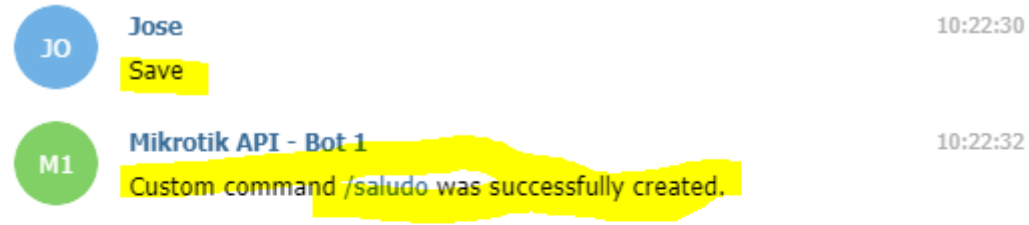
ENVIAR

Add Question

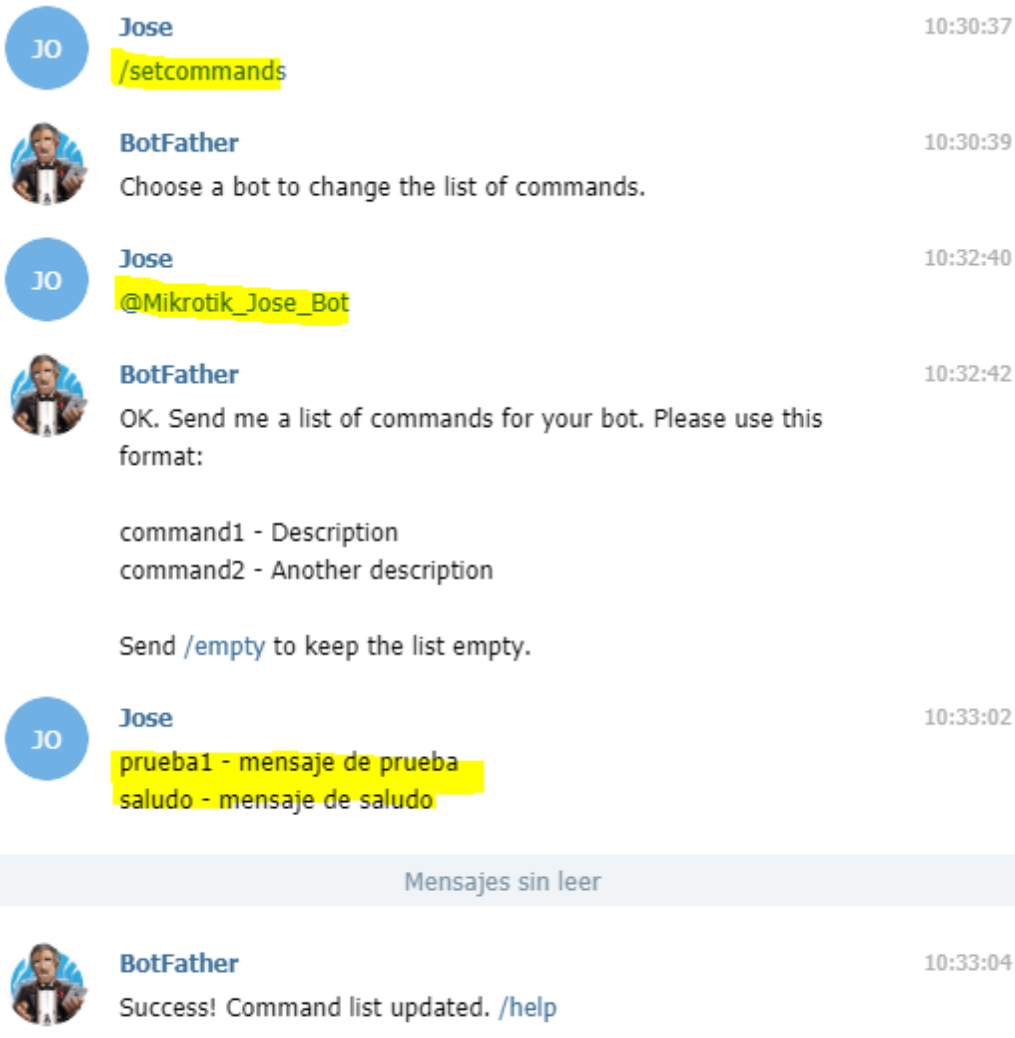
Enable Random-message Mode

Save

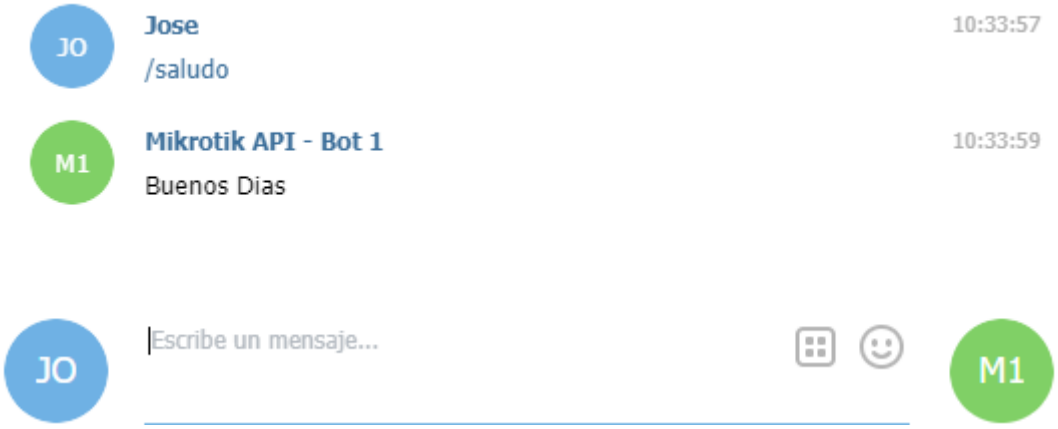
Cancel



Ahor vamos al botfather para especificarle los comandos



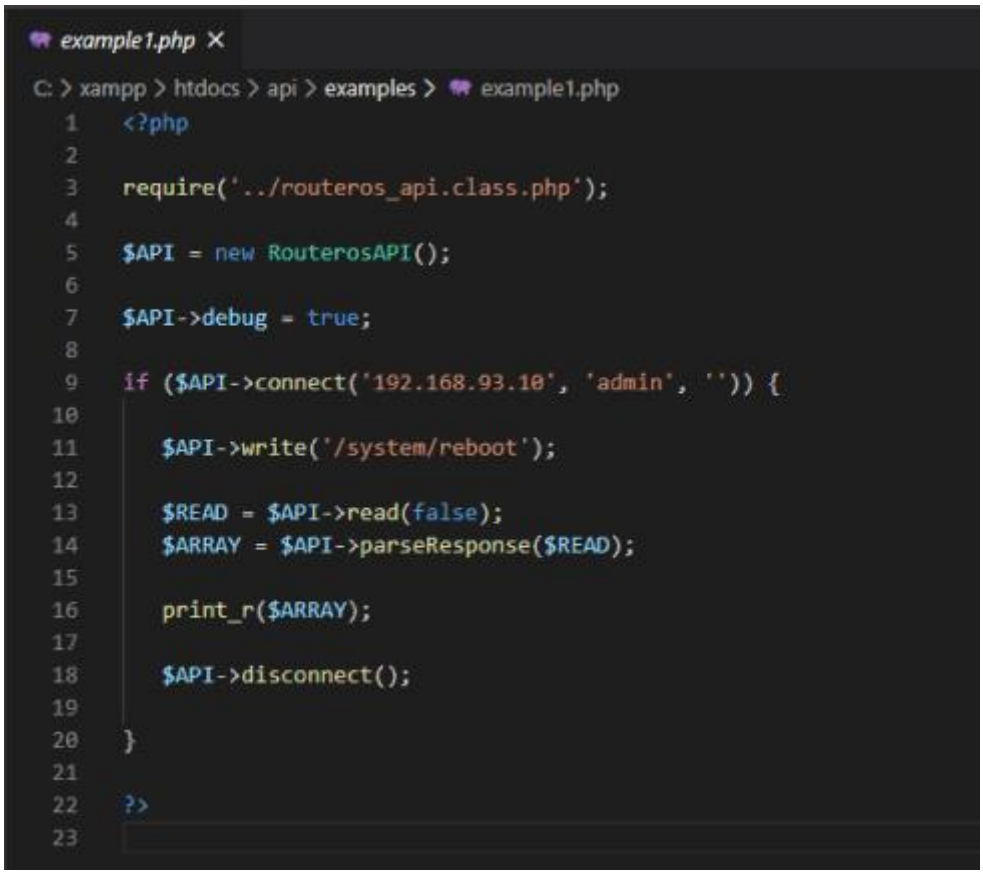
Ahora probamos en el bot



También funciona en el grupo

CONTROLAR MIKROTIK DESDE TELEGRAM POR COMANDOS

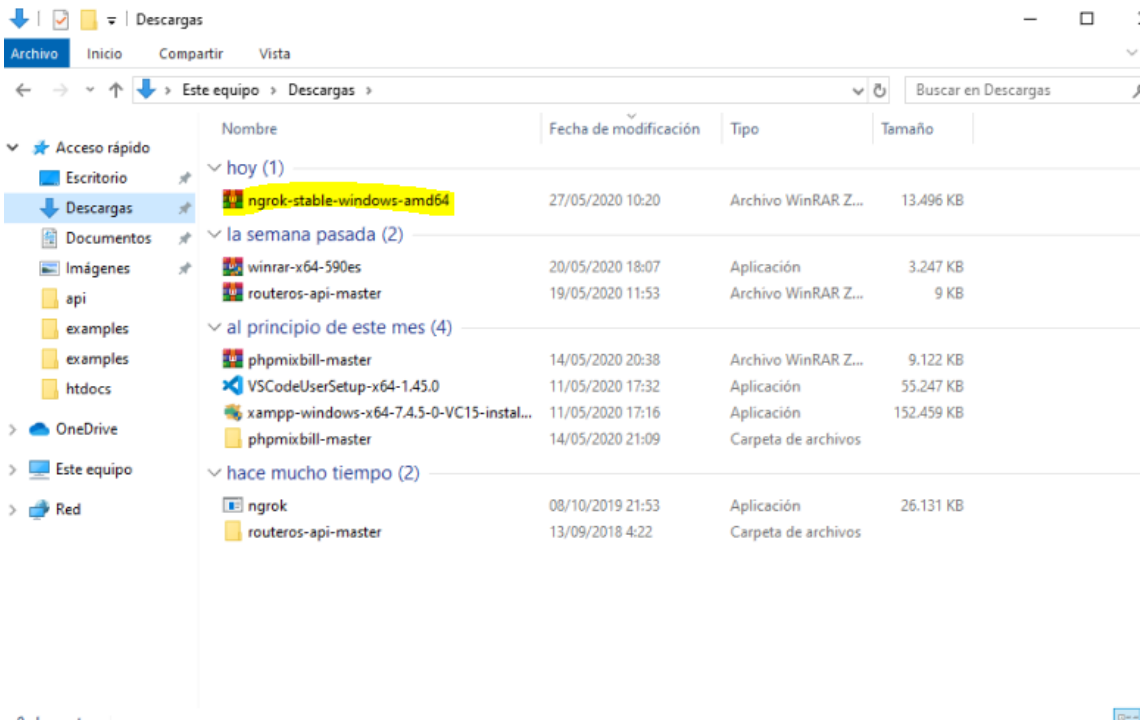
Lo primero que debemos hacer es usar crear un programa con php usando una API de Mikrotik para resetear el Mikrotik.



Usando esta API creamos un programa para que reinicie el Mikrotik.

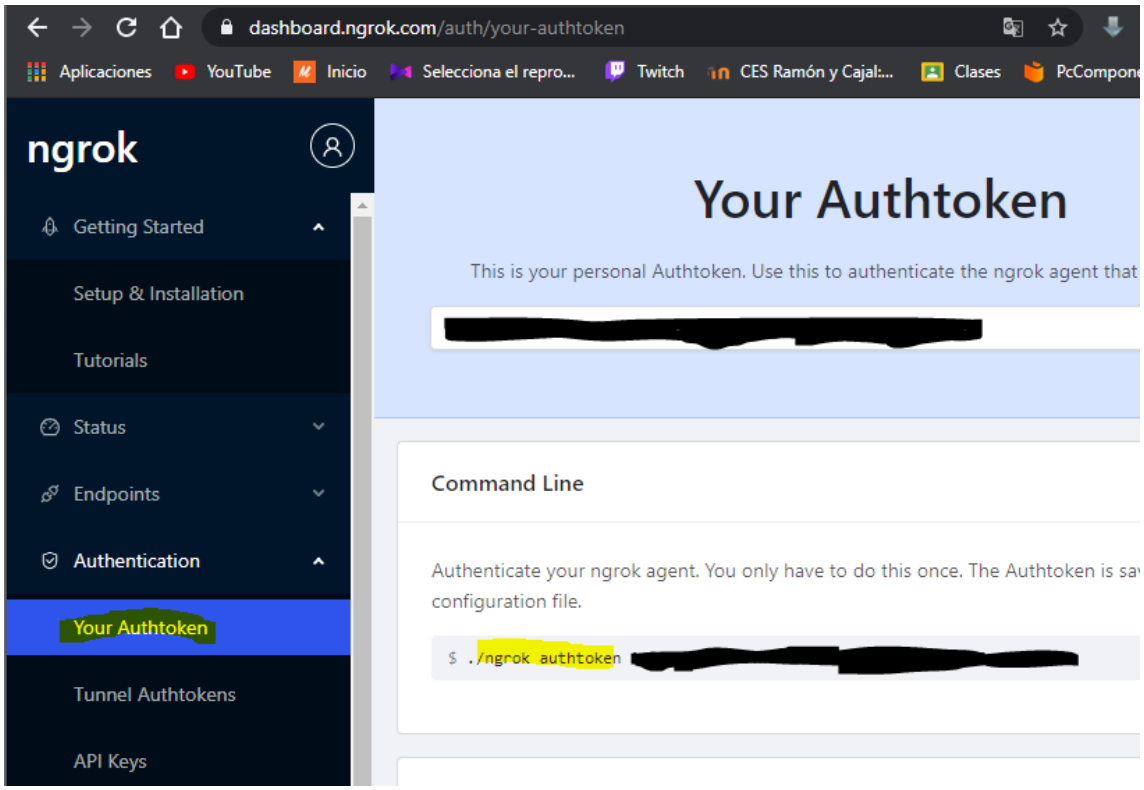
Ahora debemos subir nuestra pagina en localhost a internet para que se pueda acceder desde telegram, para ello usaremos ngrok :

Creamos una cuenta de Ngrok y descargamos el archivo en la maquina virtual donde tenemos el xampp y el script.

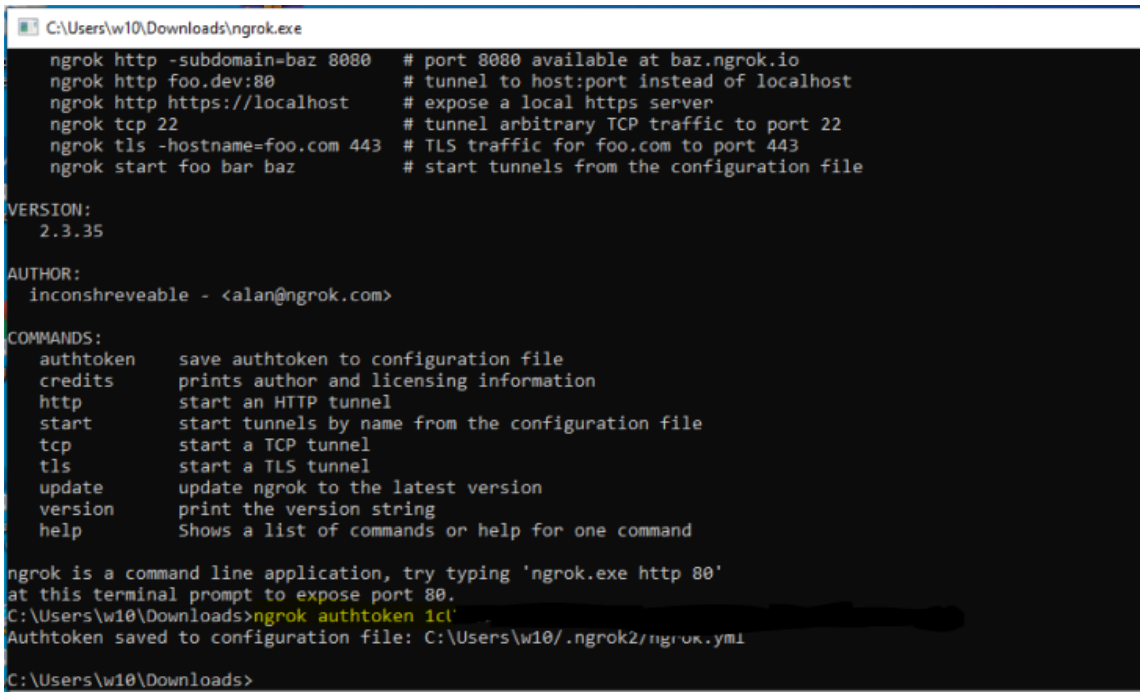


Extraemos e iniciamos ngrok, saldrá la consola de w10.

Debemos ir a nuestra cuenta de ngrok y buscar el token

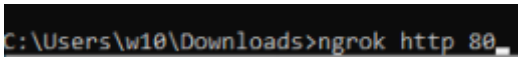


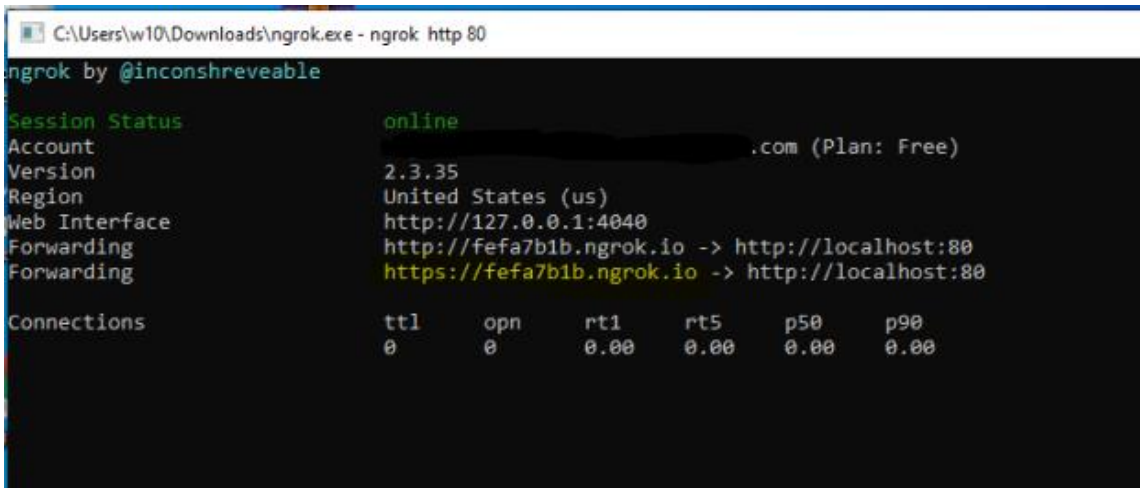
Ahora en la consola que se ha abierto antes escribiremos: *ngrok authtoken TU TOKEN*



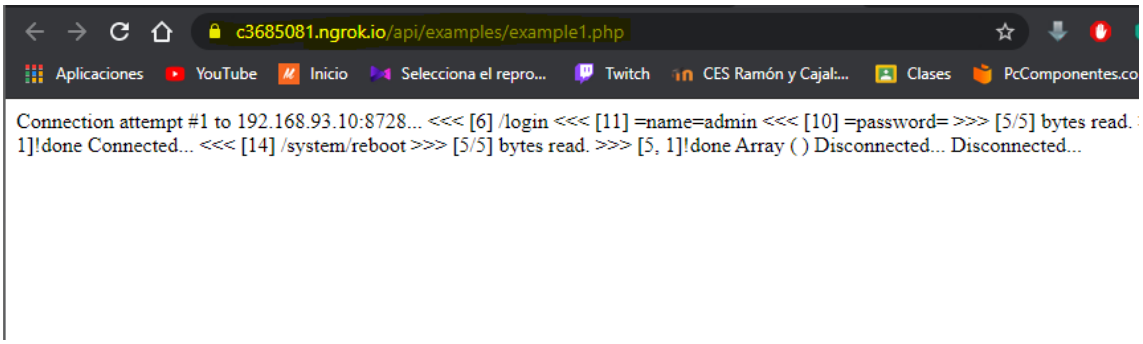
Ahora abriremos una conexión a nuestro server

ngrok http 80





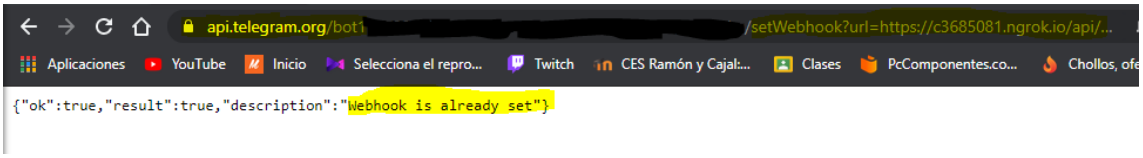
Deberemos usar la dirección https para que acceda al Mikrotik y lo reinicie



Vemos que al acceder se reinicia el Mikrotik.

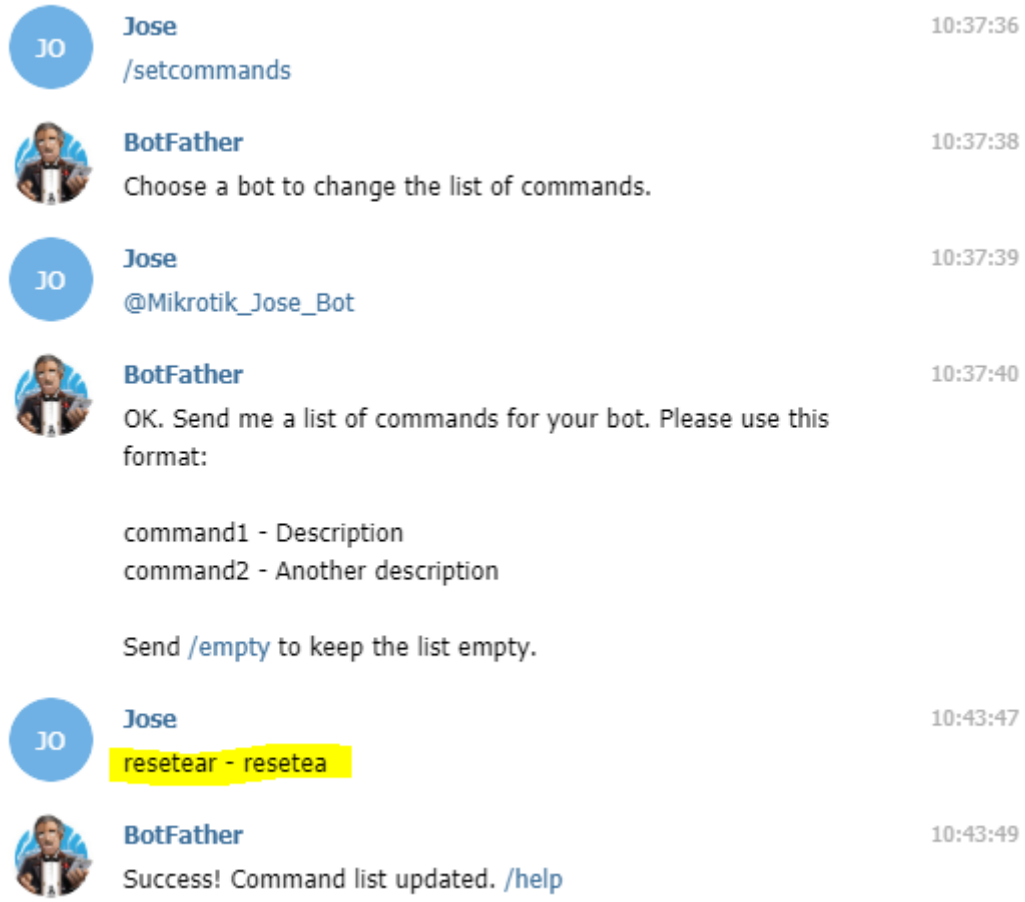
Ahora debemos usar un webhoot par enlazar telegram con la pagina y cada vez que lancemos un comando se reinicie el Mikrotik. Ejemplo:

https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11/setWebhook?url=https://www.example.com/my-telegram-bot

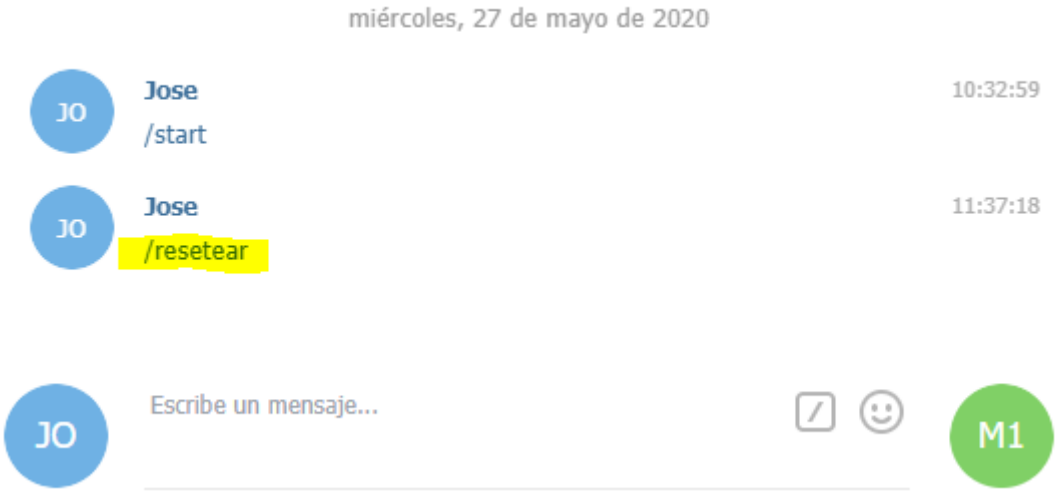


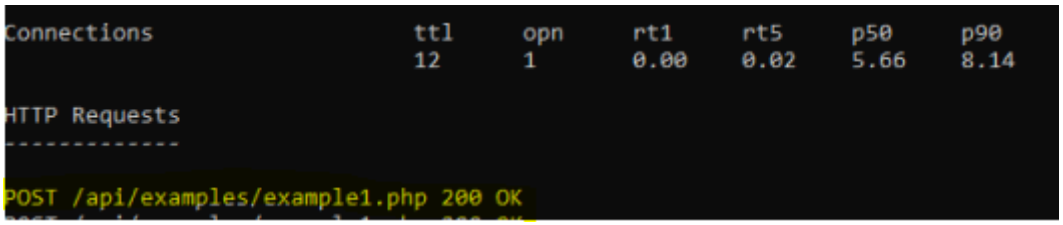
Para borrar el webhoot : https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11/setWebhook

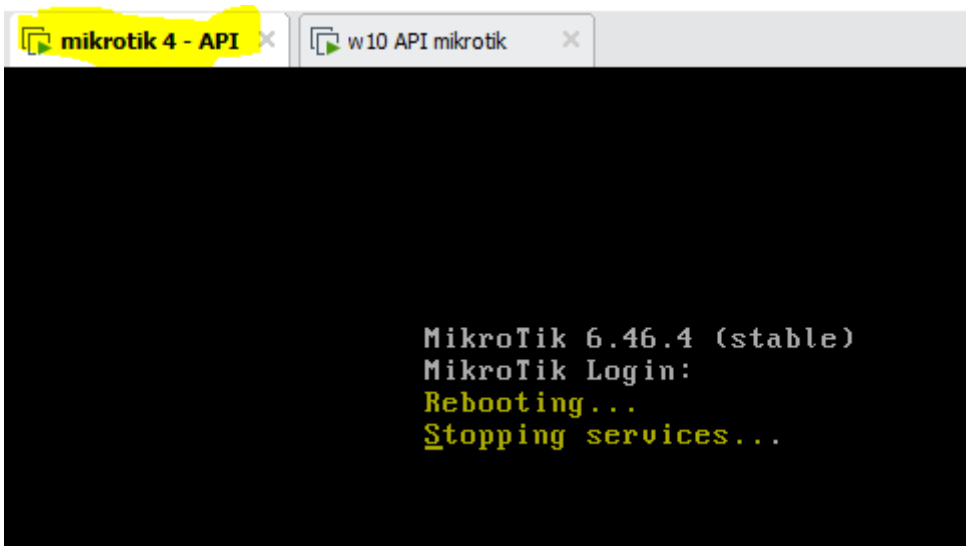
Ahora vamos a Telegram para modificar los comandos del bot, en mi caso voy a dejar un solo comando:



Cada vez que lanzamos un comando desde el bot se reinicia el Mikrotik.







Ahora modificare el script para que distinga entre comandos y que podamos enviar comandos para otras funciones.

Principal.php

```
<?php
//FUNCIONES
function reseteado(){
  include_once ('Resetear.php');
}

function sendMessage($chatId, $response){
  $url = $GLOBALS[website] . '/sendMessage?chat_id=' . $chatId .
'&parse_mode=HTML&text=' . urlencode($response);
  file_get_contents($url);
}

$botToken = "1107807185:AAGfyusehJynVBjszClRFtQGeQi_Gtz0y1w";
```

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

$update = file_get_contents('php://input');
$update = json_decode($update, TRUE);

$chatId = $update["message"]["chat"]["id"];
$chatType = $update["message"]["chat"]["type"];

$message = $update["message"]["text"];

switch ($message) {
  case '/resetear':
    $response = "Voy a resetear el router";
    sendMessage($chatId, $response);
    reseteado();
    sleep(16);
    $response = "Ya he reseteado el router";
    sendMessage($chatId, $response);
  break;

case '/start':
    $response = "Ya estoy activo";
    sendMessage($chatId, $response);
  break;

  default:
    $response = "Ese comando no lo reconozco";
    sendMessage($chatId, $response);
```

```
  break;
}

?>
```

Resetear.php

```
<?php

require('../routeros_api.class.php');

$API = new RouterosAPI();

$API->debug = true;

if ($API->connect('192.168.93.10', 'admin', '')) {

  $API->write('/system/reboot');

  $READ = $API->read(false);
  $ARRAY = $API->parseResponse($READ);

  print_r($ARRAY);

  $API->disconnect();
}
?>
```

Con los dos scripts he podido reiniciar el router desde Mikrotik y que responda a los comandos. Con más tiempo se podrían personalizar los comandos y por ejemplo crear un comando para que nos devuelva las personas conectadas al router.

Bibliografía

https://jameeljtnet0.blogspot.com/2017/02/phpmixbill-v5-install-mikrotik-api-for.html

https://github.com/ibnux/phpmixbill

https://github.com/ibnux/phpmixbill/issues/12

https://wiki.mikrotik.com/wiki/Manual:API

https://github.com/BenMenking/routeros-api

https://soporte.syscom.mx/es/articles/3140860-mikrotik-integracion-de-telegram-con-mikrotik-para-envio-de-notificaciones-automatizadas

https://www.youtube.com/watch?v=YztlYfzeZf4

https://www.youtube.com/watch?time_continue=128&v=G_KvLv3G_LE&feature=emb_logo

https://medium.com/@xabaras/setting-your-telegram-bot-webhook-the-easy-way-c7577b2d6f72

https://www.luciano.im/blog/comparti-tu-localhost-de-forma-segura-con-ngrok-y-localtunnel/