

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
$0 \leq \text{len} \leq 0x7F$	1	len, byte más bajo
$0x80 \leq \text{len} \leq 0x3FFF$	2	len 0x8000, dos bytes inferiores
$0x4000 \leq \text{len} \leq 0x1FFFFFFF$	3	len 0xC00000, tres bytes inferiores
$0x200000 \leq \text{len} \leq 0xFFFFFFFF$	4	len 0xE0000000
$\text{len} > 0x10000000$	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 0x7FFFFFFF , solo se admite una longitud de cuatro bytes;
- Si el primer byte de la palabra es $\geq 0xF8$, 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;

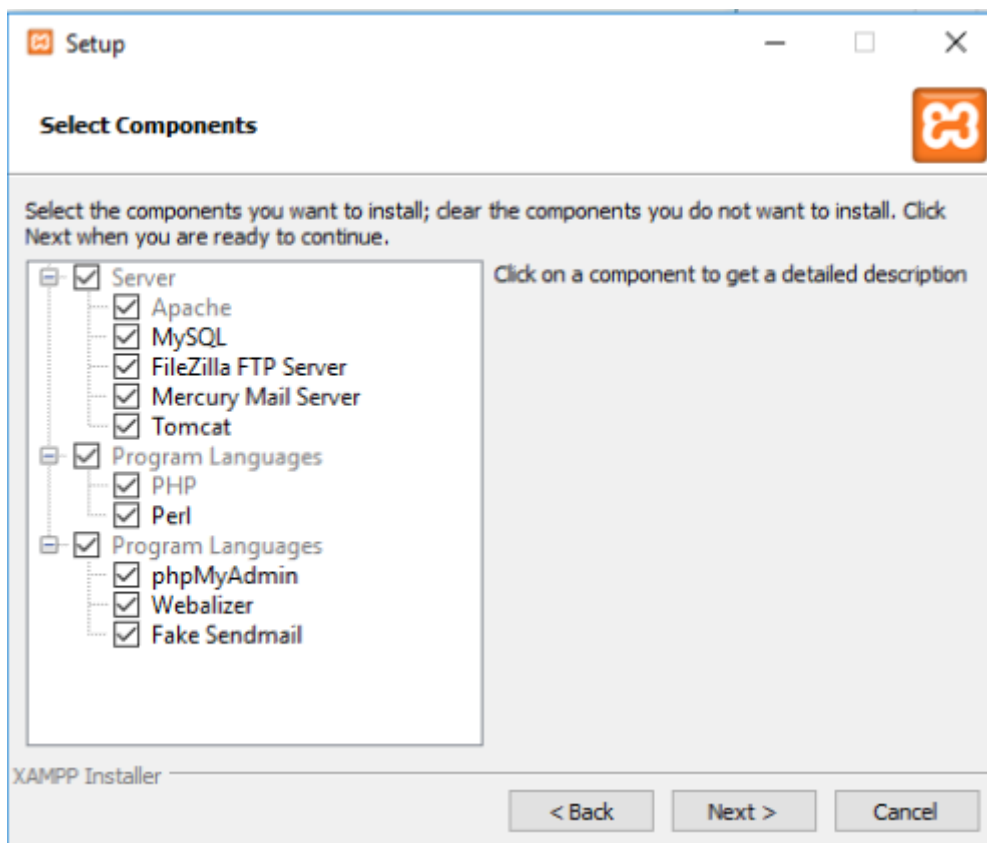
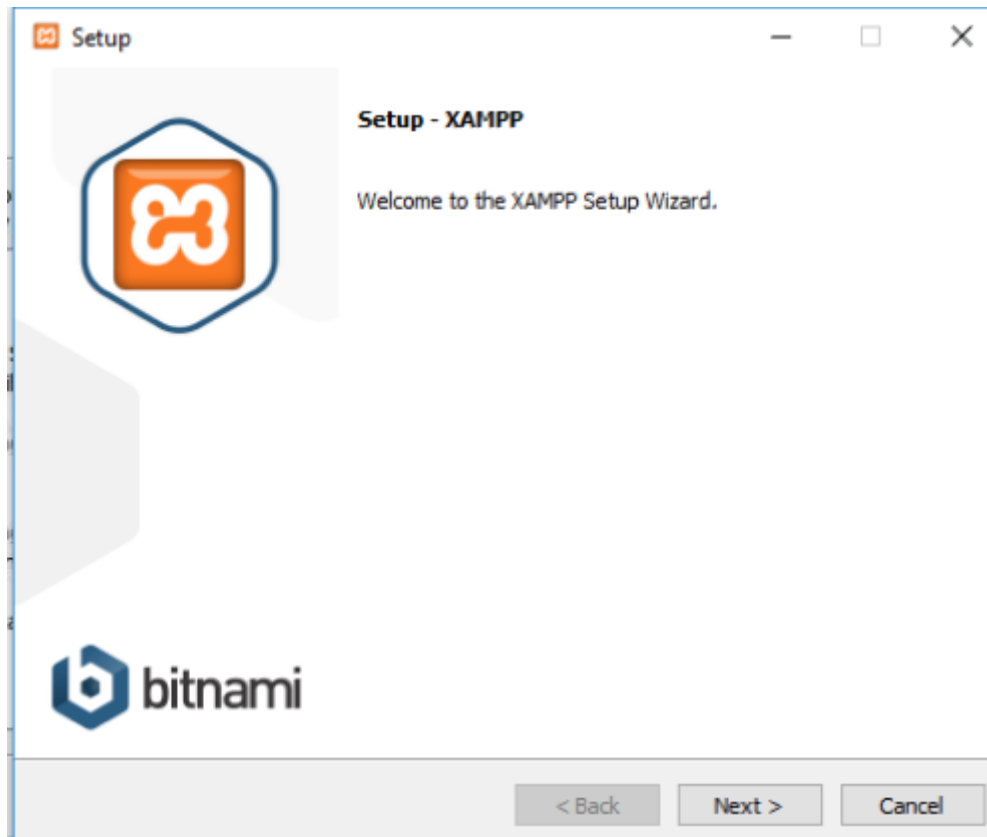
Oraciones API

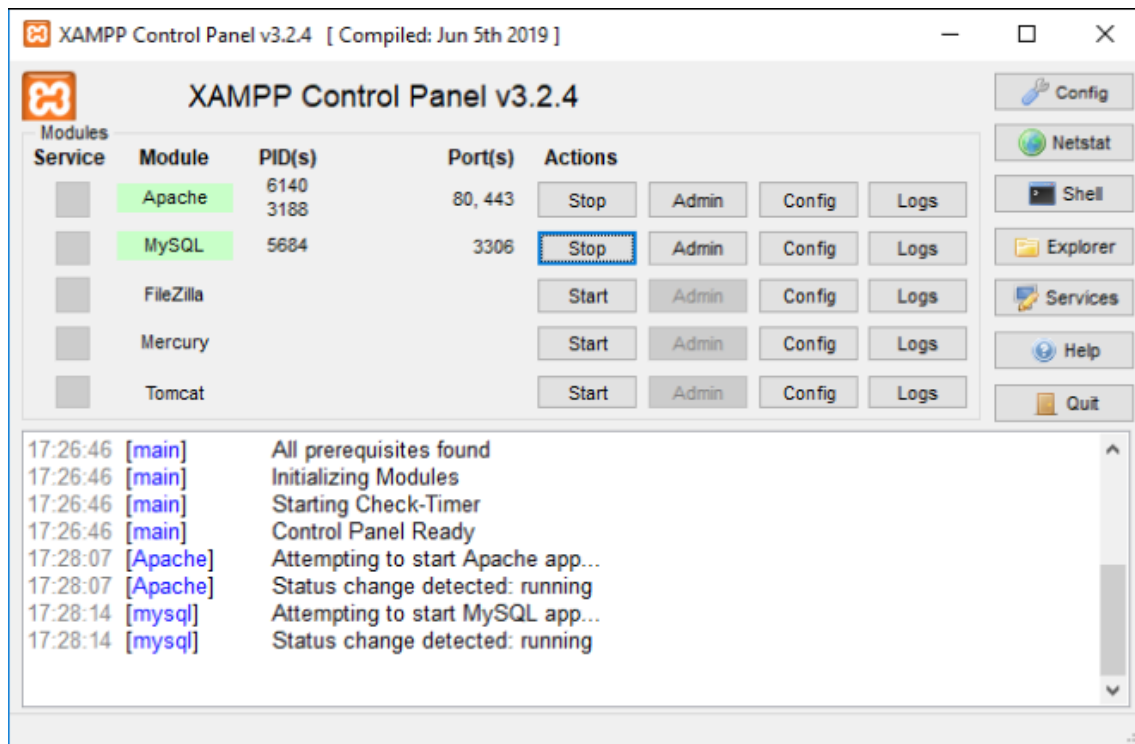
La oración API es el principal objeto de comunicación mediante API.

- 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.

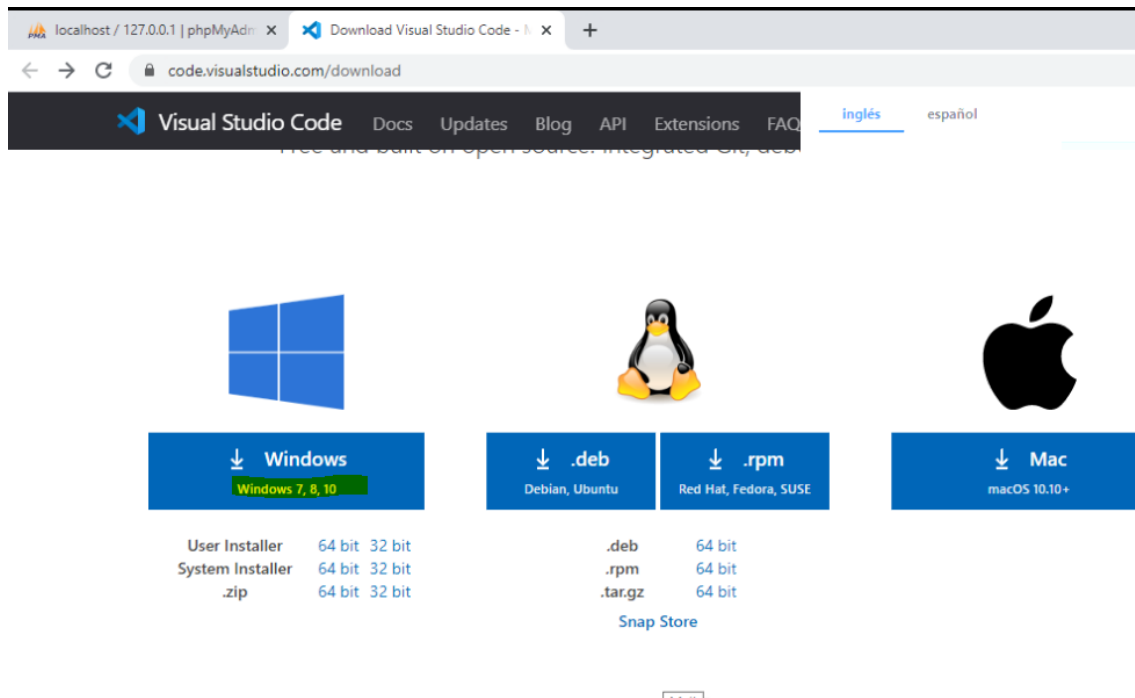
INSTALACION DE XAMPP

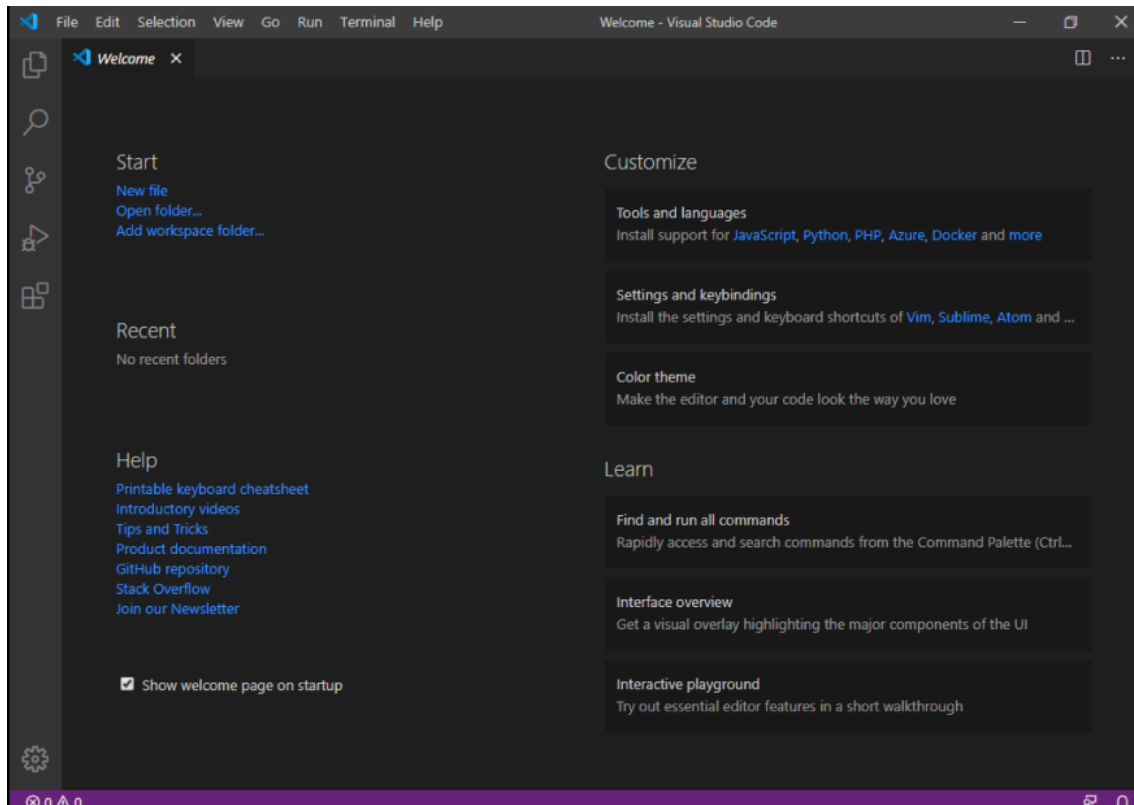
Descargamos el instalador de la página oficial de apachefriends.





INSTALAR VISUAL STUDIO CODE





PHPMixBill

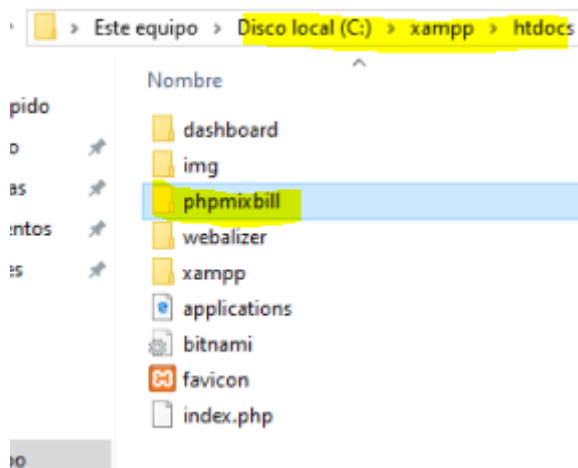
¿Qué es PHPMixBill?

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.

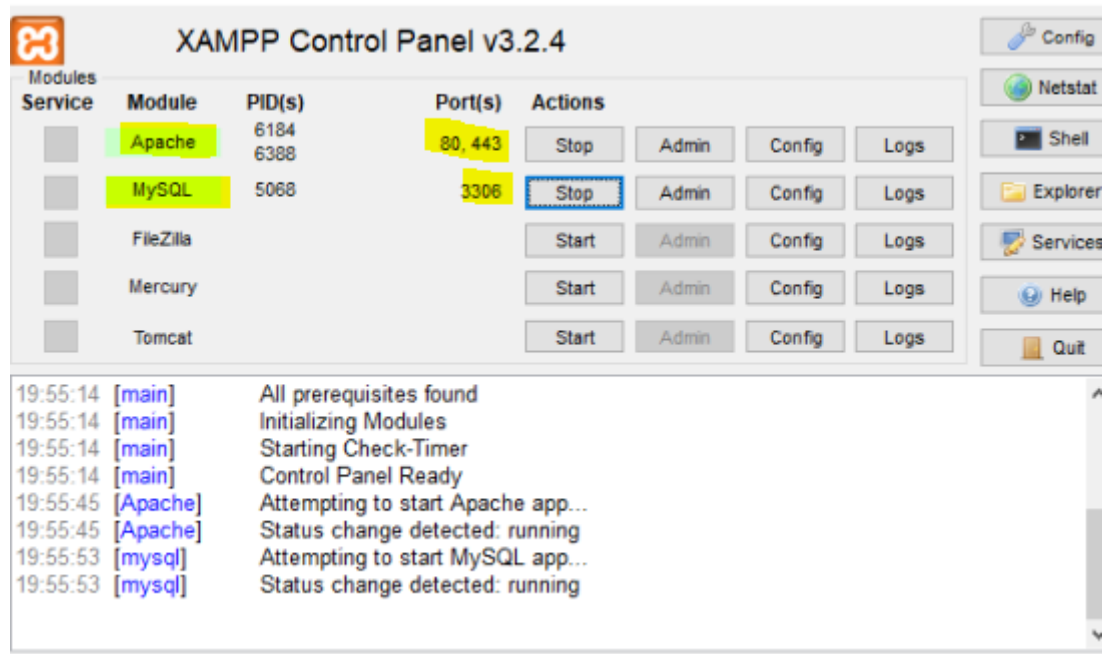
INSTALACIÓN

Descargamos el archivo [phpmixbill](#)

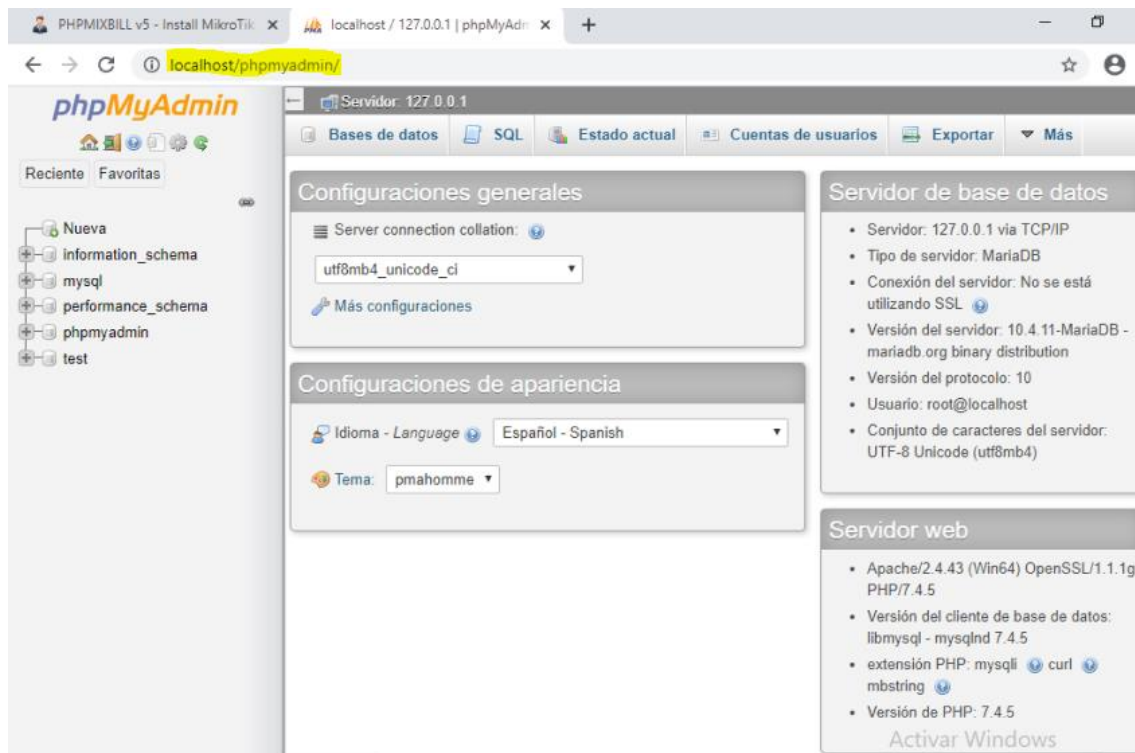
Extraemos los archivos y copiamos la carpeta extraída en C:/xampp/htdocs y le cambiamos el nombre a phpmixbill



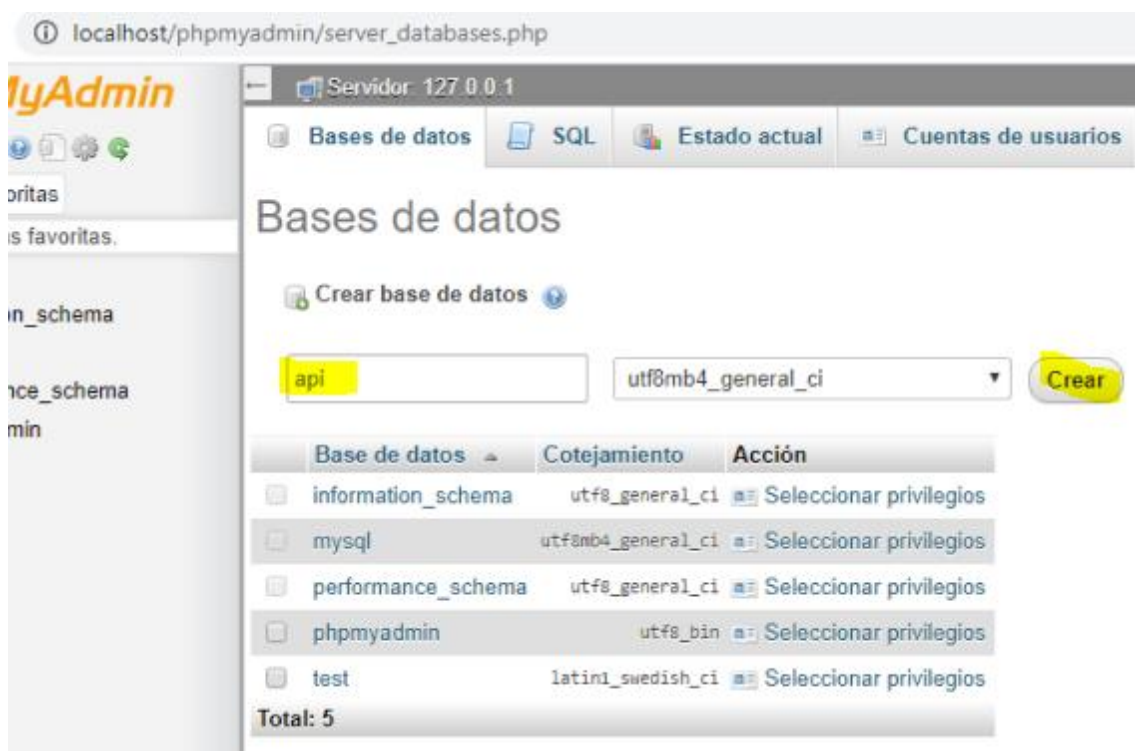
Iniciamos apache y MySQL en XAMPP



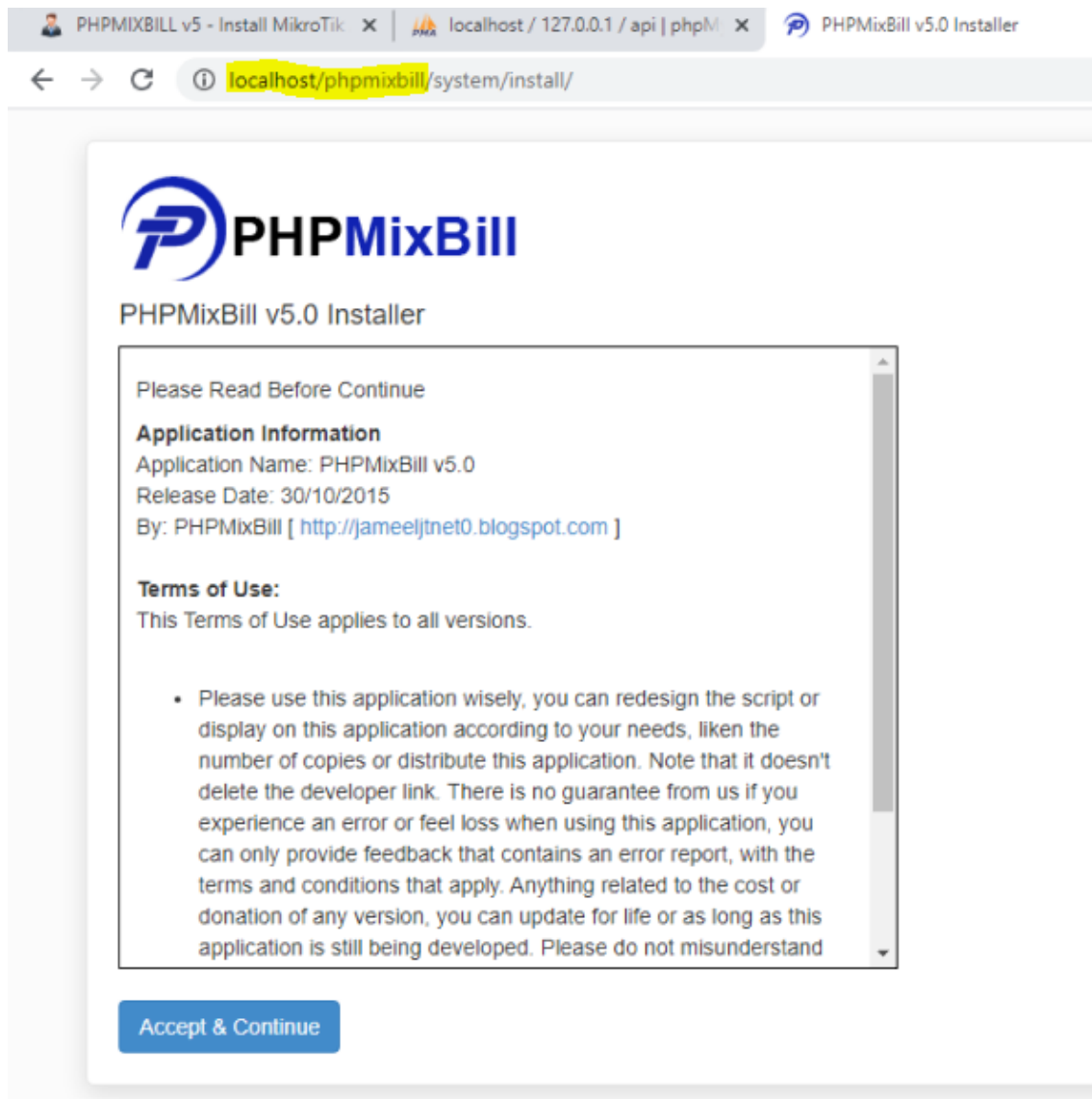
Entramos con Chrome en phpMyAdmin



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 v5.0 Installer

Database Connection & Site config

Application URL

Application url without trailing slash at the end of url (e.g. http://172.16.10.10). Plea

Database Host

Database Username

Database Password

Database Name

GitHub - ibnux/phpmixbill: PHP x localhost / 127.0.0.1 / api | php x PHPMixBill v5.0 Installer

localhost/phpmixbill/system/install/step4.php



PHPMixBill v5.0 Installer

Config File Created and Database Imported.

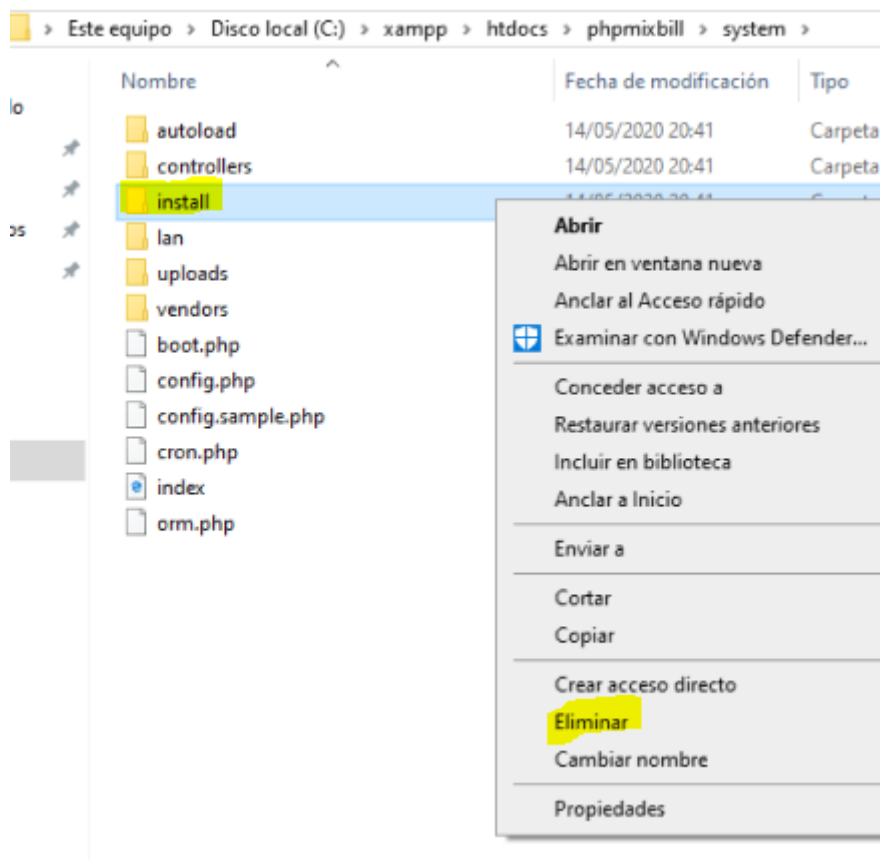
Click Continue

Copyright © 2015 PHPMixBill. All Rights Reserved

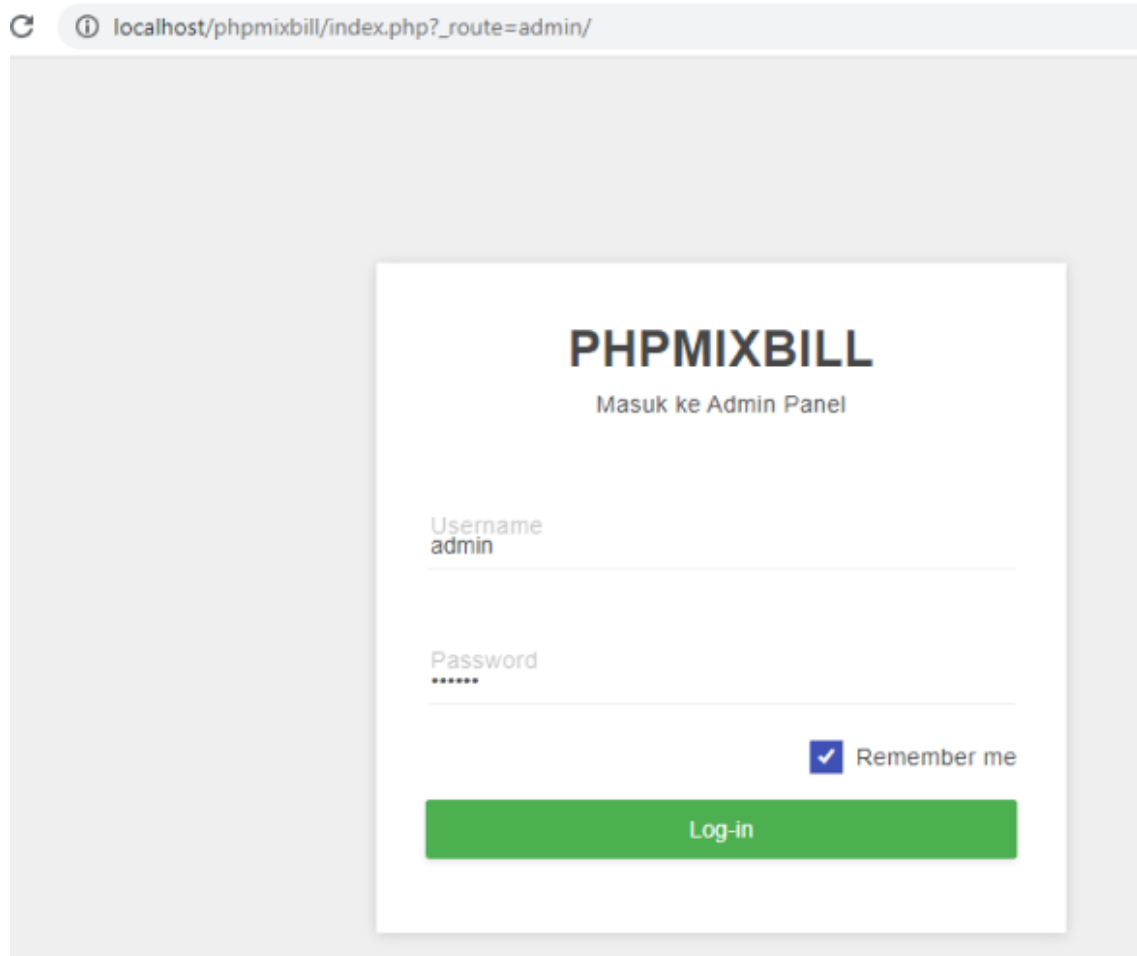
Estas son nuestras credenciales, debemos guardarlas o echarle captura



A continuación, borramos el archivo de instalación.



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



← → ↻ ⓘ localhost/phpmixbill/index.php?_route=admin/post

Fatal error: Uncaught PDOException: SQLSTATE[42S02]: Base table or view not found: 1146 Table 'api.tbl_users' doesn't exist in C:\xampp\htdocs\phpmixbill\system\orm.php:435 Stack trace: #0 C:\xampp\htdocs\phpmixbill\system\orm.php(435): PDOStatement->execute() #1 C:\xampp\htdocs\phpmixbill\system\orm.php(1834): ORM->_execute('SELECT * FROM '...', Array, 'default') #2 C:\xampp\htdocs\phpmixbill\system\orm.php(613): ORM->_run() #3 C:\xampp\htdocs\phpmixbill\system\controllers\admin.php(21): ORM->find_one() #4 C:\xampp\htdocs\phpmixbill\system\boot.php(207): include('C:\xampp\htdocs\...') #5 C:\xampp\htdocs\phpmixbill\index.php(13): require('C:\xampp\htdocs\...') #6 {main} thrown in C:\xampp\htdocs\phpmixbill\system\orm.php on line 435

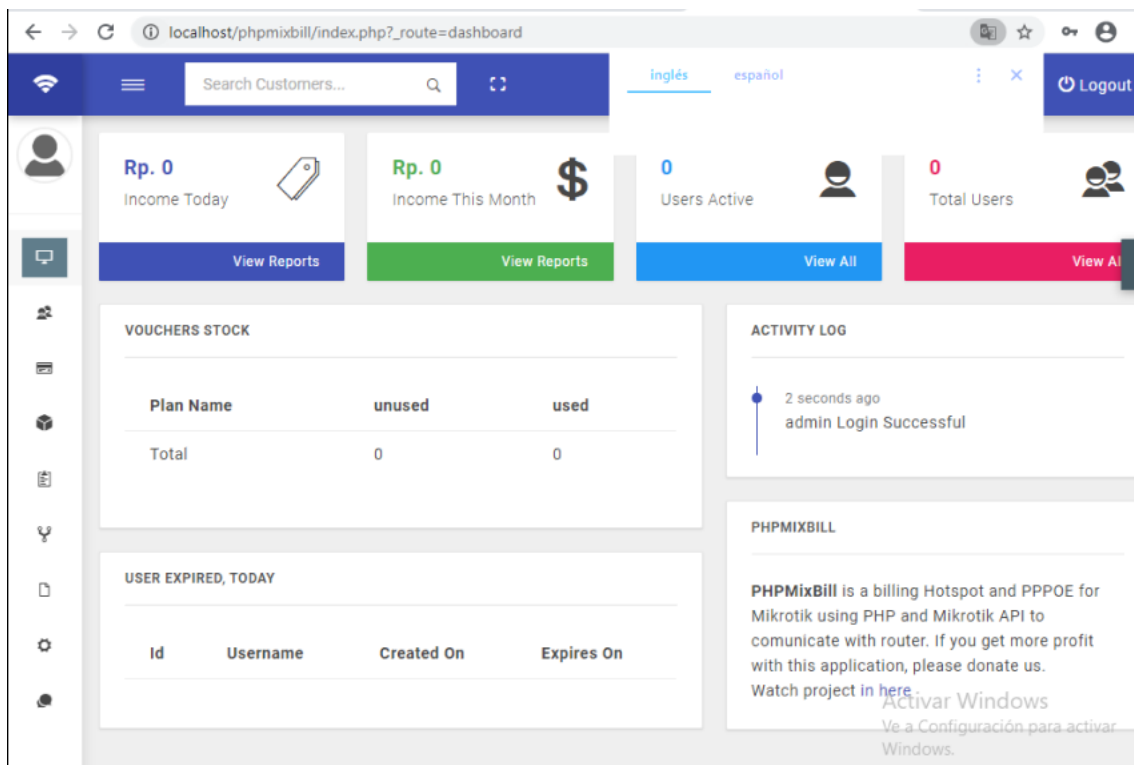
Como vemos al iniciar sesión da error, para arreglarlo en el archivo
C:\xampp\htdocs\phpmixbill\system\install\phpmixbill.sql, en la línea 92 cambiamos
'0000-00-00 00:00:00' por CURRENT_TIMESTAMP

```
phpmixbill.sql X
C: > xampp > htdocs > phpmixbill > system > install > phpmixbill.sql

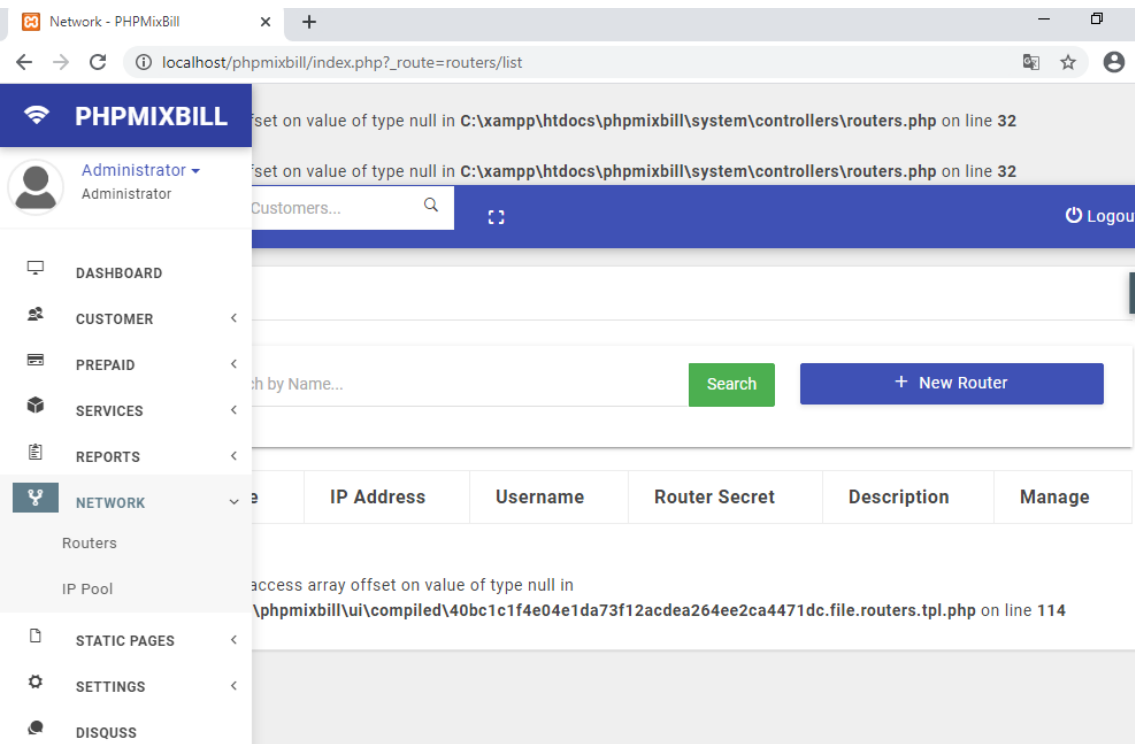
75 `author` varchar(60) DEFAULT NULL,
76 PRIMARY KEY (`id`)
77 ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
78
79 -----
80 -- Dump Data for `tbl_language`
81 --
82
83 INSERT INTO `tbl_language` (`id`,`name`,`folder`,`author`) VALUES (
84 INSERT INTO `tbl_language` (`id`,`name`,`folder`,`author`) VALUES (
85
86 -----
87 -- Structure for `tbl_logs`
88 --
89
90 CREATE TABLE `tbl_logs` (
91 `id` int(10) NOT NULL AUTO_INCREMENT,
92 `date` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
93 `type` varchar(50) NOT NULL,
94 `description` text NOT NULL,
95 `userid` int(10) NOT NULL,
96 `ip` text NOT NULL
```

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



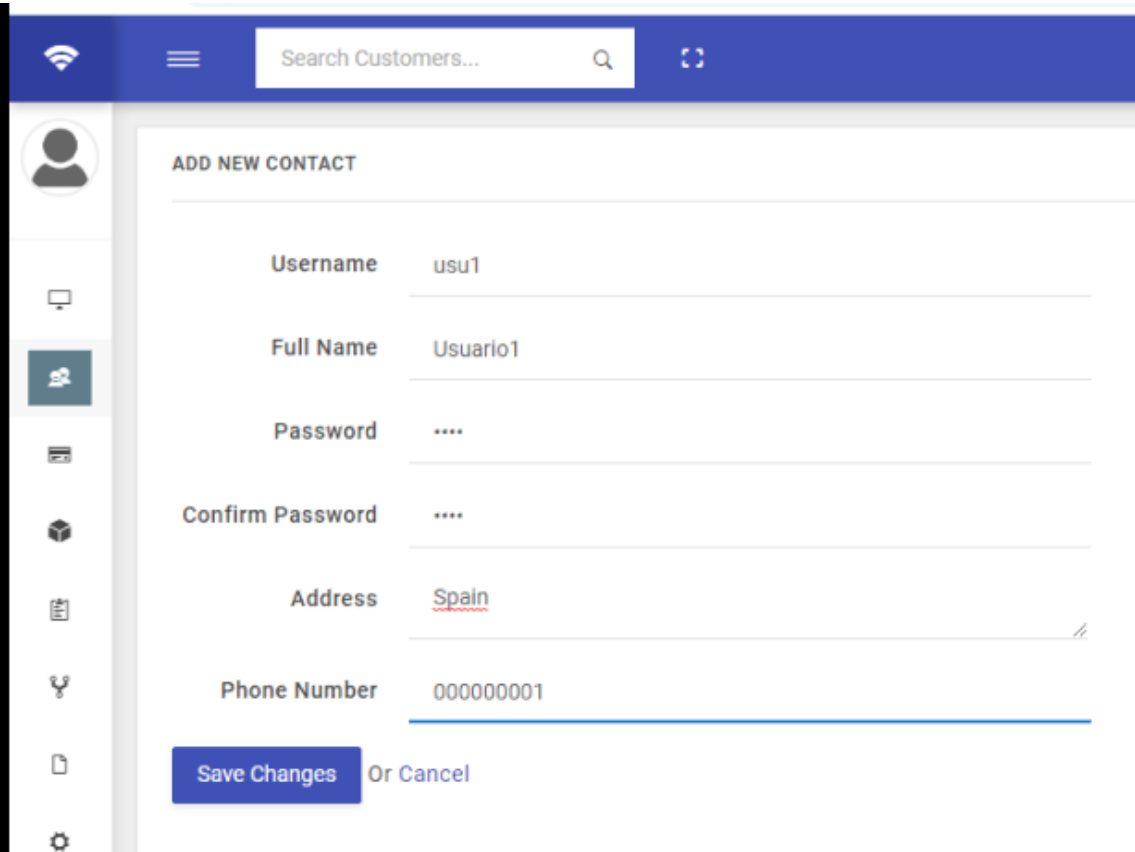
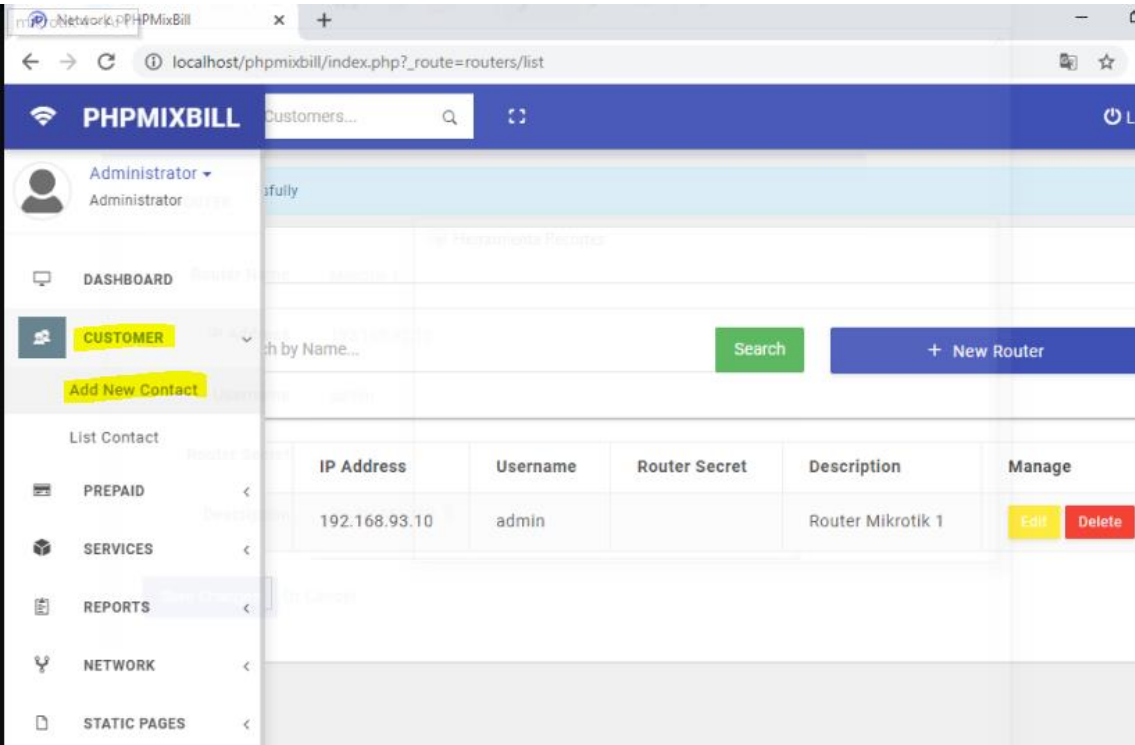
En network pinchamos en routers, luego en new router

The screenshot shows the 'ADD ROUTER' form. The form fields are:

- Router Name: Mikrotik 1
- IP Address: 192.168.93.10
- Username: admin
- Router Secret: (empty)
- Description: Router Mikrotik 1

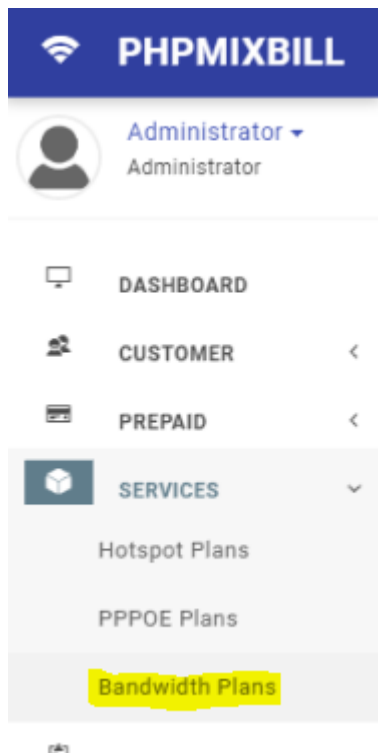
At the bottom of the form, there are two buttons: 'Save Changes' and 'Or Cancel'.

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


The image shows the 'ADD NEW BANDWIDTH' form in the PHPMIXBILL application. The form is titled 'ADD NEW BANDWIDTH' and has a search bar at the top with the placeholder text 'Search Customers...'. The form contains three input fields: 'Bandwidth Name' with the value '2-Mbps', 'Rate Download' with the value '2', and 'Rate Upload' with the value '1'. The units for the rates are set to 'Mbps' for both. At the bottom of the form, there are two buttons: 'Submit' and 'Or Cancel'.


Field	Value	Unit
Bandwidth Name	2-Mbps	
Rate Download	2	Mbps
Rate Upload	1	Mbps


 **PHPMIXBILL**

 **Administrator** ▼
Administrator

 **DASHBOARD**

 **CUSTOMER** <


 **PREPAID** <





 **SERVICES** ▼


Hotspot Plans








PPPOE Plans

Bandwidth Plans

 **REPORTS** <



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

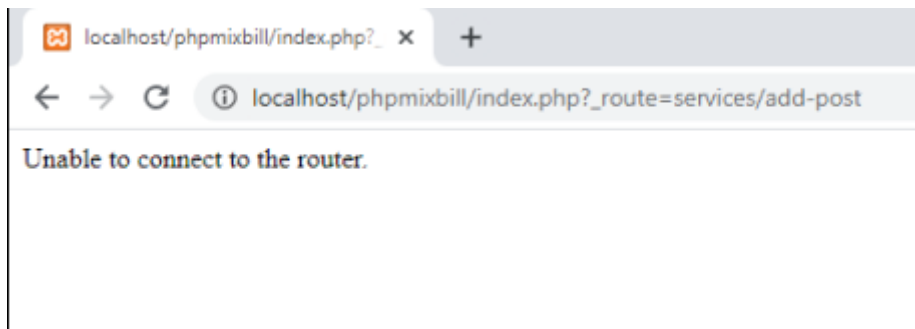
Days ▼

Router Name

Mikrotik 1 ▼

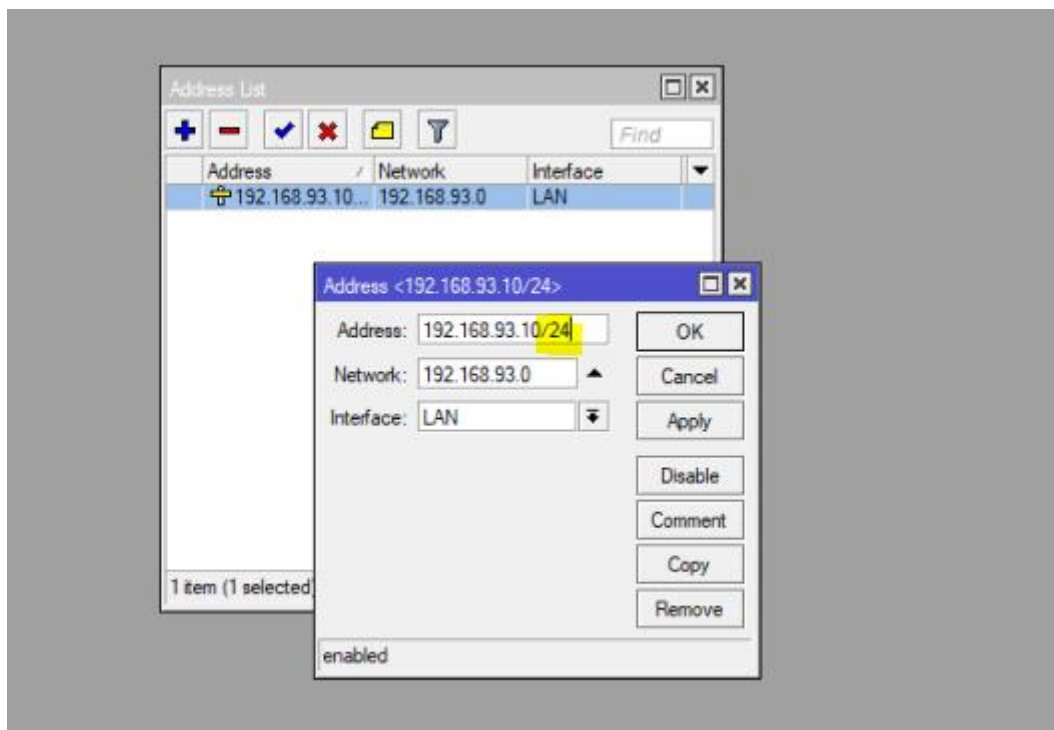
Save Changes

 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 red de Mikrotik

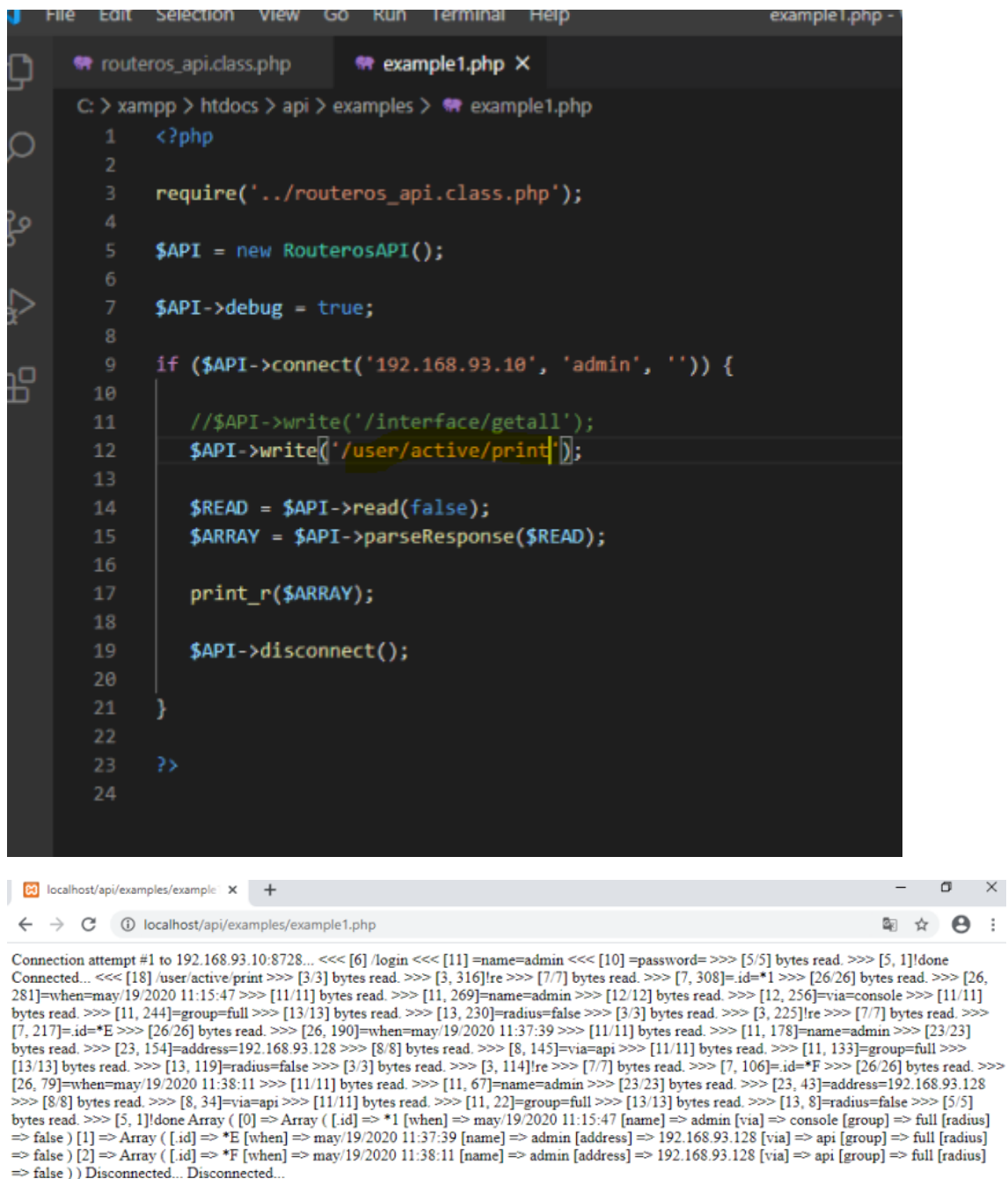
```
routeros_api.class.php example1.php X
C: > xampp > htdocs > api > examples > example1.php
1  <?php
2
3  require('../routeros_api.class.php');
4
5  $API = new RouterosAPI();
6
7  $API->debug = true;
8
9  if ($API->connect('192.168.93.10', 'admin', '')) {
10
11      $API->write('/interface/getall');
12
13
14      $READ = $API->read(false);
15      $ARRAY = $API->parseResponse($READ);
16
17      print_r($ARRAY);
18
19      $API->disconnect();
20
21  }
22
23  ?>
24
```

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



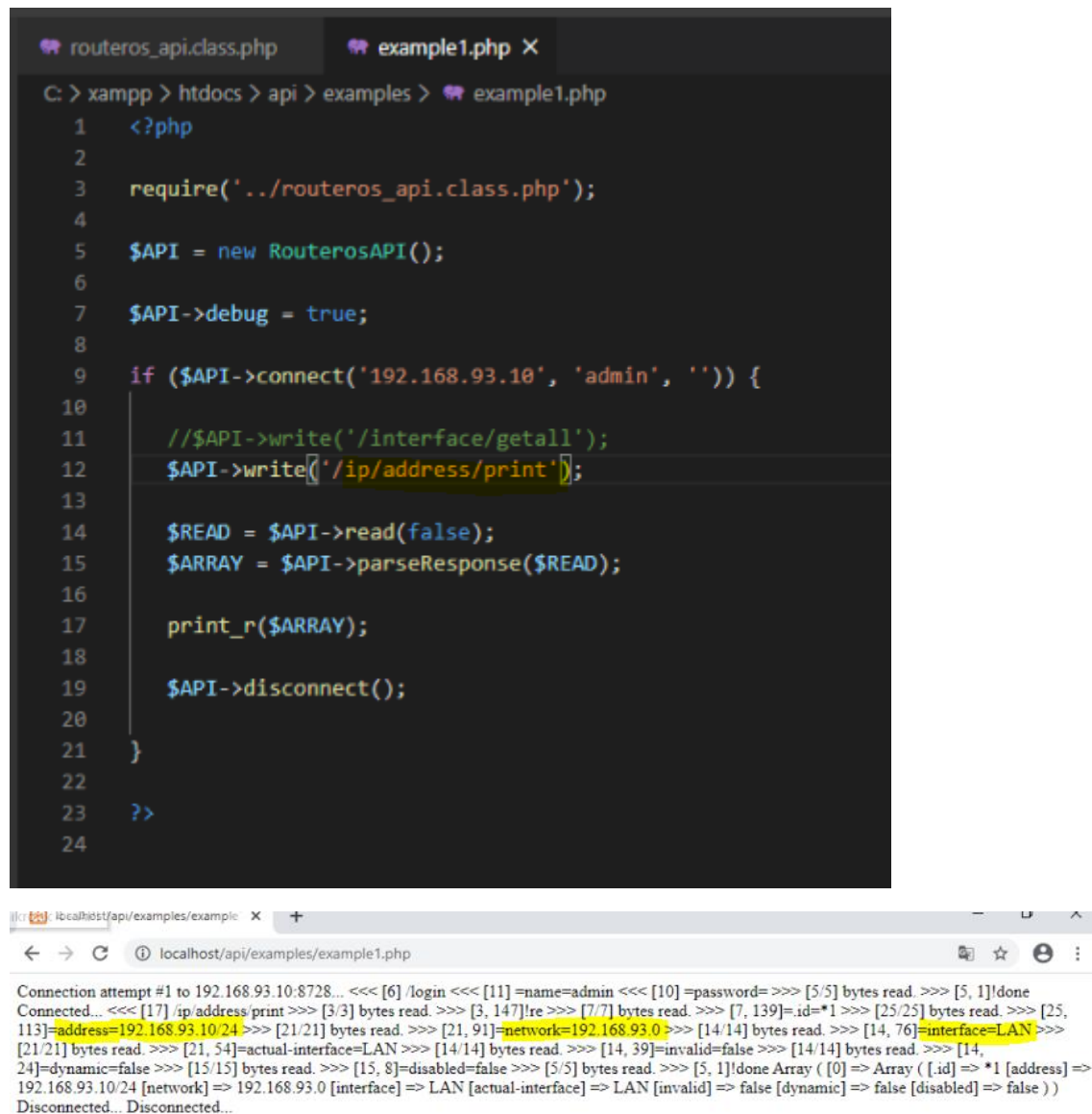
The image shows a code editor window with a file named `example1.php` open. The code is a PHP script that interacts with a RouterOS API. It includes a `require` statement for `routeros_api.class.php`, initializes a `RouterosAPI` object, sets `$API->debug = true;`, and attempts to connect to the API at `192.168.93.10` with the username `admin`. After a successful connection, it sends a `/user/active/print` command and prints the response using `print_r($ARRAY);`.

Below the code editor, a web browser window shows the output of the script. The output is a detailed log of the API connection attempt and the response received. It shows the connection attempt to `192.168.93.10:8728`, the successful login, and the response to the `/user/active/print` command. The response is an array containing two active users: `admin` and `api`.

```
Connection attempt #1 to 192.168.93.10:8728... <<< [6] /login <<< [11] =name=admin <<< [10] =password=>>> [5/5] bytes read. >>> [5, 1]!done
Connected... <<< [18] /user/active/print >>> [3/3] bytes read. >>> [3, 316]!re >>> [7/7] bytes read. >>> [7, 308]=id=*1 >>> [26/26] bytes read. >>> [26,
281]=when=may/19/2020 11:15:47 >>> [11/11] bytes read. >>> [11, 269]=name=admin >>> [12/12] bytes read. >>> [12, 256]=via=console >>> [11/11]
bytes read. >>> [11, 244]=group=full >>> [13/13] bytes read. >>> [13, 230]=radius=false >>> [3/3] bytes read. >>> [3, 225]!re >>> [7/7] bytes read. >>>
[7, 217]=id=*E >>> [26/26] bytes read. >>> [26, 190]=when=may/19/2020 11:37:39 >>> [11/11] bytes read. >>> [11, 178]=name=admin >>> [23/23]
bytes read. >>> [23, 154]=address=192.168.93.128 >>> [8/8] bytes read. >>> [8, 145]=via=api >>> [11/11] bytes read. >>> [11, 133]=group=full >>>
[13/13] bytes read. >>> [13, 119]=radius=false >>> [3/3] bytes read. >>> [3, 114]!re >>> [7/7] bytes read. >>> [7, 106]=id=*F >>> [26/26] bytes read. >>>
[26, 79]=when=may/19/2020 11:38:11 >>> [11/11] bytes read. >>> [11, 67]=name=admin >>> [23/23] bytes read. >>> [23, 43]=address=192.168.93.128
>>> [8/8] bytes read. >>> [8, 34]=via=api >>> [11/11] bytes read. >>> [11, 22]=group=full >>> [13/13] bytes read. >>> [13, 8]=radius=false >>> [5/5]
bytes read. >>> [5, 1]!done Array ( [0] => Array ( [id] => *1 [when] => may/19/2020 11:15:47 [name] => admin [via] => console [group] => full [radius]
=> false ) [1] => Array ( [id] => *E [when] => may/19/2020 11:37:39 [name] => admin [address] => 192.168.93.128 [via] => api [group] => full [radius]
=> false ) [2] => Array ( [id] => *F [when] => may/19/2020 11:38:11 [name] => admin [address] => 192.168.93.128 [via] => api [group] => full [radius]
=> false ) ) Disconnected... Disconnected...
```

Nos muestra que solo esta admin conectado

Mostrar direcciones ip asignadas



```
routeros_api.class.php example1.php X
C: > xampp > htdocs > api > examples > example1.php
1  <?php
2
3  require('../routeros_api.class.php');
4
5  $API = new RouterosAPI();
6
7  $API->debug = true;
8
9  if ($API->connect('192.168.93.10', 'admin', '')) {
10
11     //$API->write('/interface/getall');
12     $API->write('/ip/address/print');
13
14     $READ = $API->read(false);
15     $ARRAY = $API->parseResponse($READ);
16
17     print_r($ARRAY);
18
19     $API->disconnect();
20
21 }
22
23 ?>
24
```

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] /ip/address/print >>> [3/3] bytes read. >>> [3, 147]!re >>> [7/7] bytes read. >>> [7, 139]=id=*1 >>> [25/25] bytes read. >>> [25,
113]=address=192.168.93.10/24 >>> [21/21] bytes read. >>> [21, 91]=network=192.168.93.0 >>> [14/14] bytes read. >>> [14, 76]=interface=LAN >>>
[21/21] bytes read. >>> [21, 54]=actual-interface=LAN >>> [14/14] bytes read. >>> [14, 39]=invalid=false >>> [14/14] bytes read. >>> [14,
24]=dynamic=false >>> [15/15] bytes read. >>> [15, 8]=disabled=false >>> [5/5] bytes read. >>> [5, 1]!done Array ([0] => Array ([.id] => *1 [address] =>
192.168.93.10/24 [network] => 192.168.93.0 [interface] => LAN [actual-interface] => LAN [invalid] => false [dynamic] => false [disabled] => false))
Disconnected... Disconnected...

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

```
File Edit Selection View Go Run Terminal Help
otik 4 - API

routeros_api.class.php  example1.php X

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

1  <?php
2
3  require('../routeros_api.class.php');
4
5  $API = new RouterosAPI();
6
7  $API->debug = true;
8
9  if ($API->connect('192.168.93.10', 'admin', '')) {
10
11
12      $API->write('/ip/service/getall');
13
14      $READ = $API->read(false);
15      $ARRAY = $API->parseResponse($READ);
16
17      print_r($ARRAY);
18
19      $API->disconnect();
20
21  }
22
23  ?>
24
```

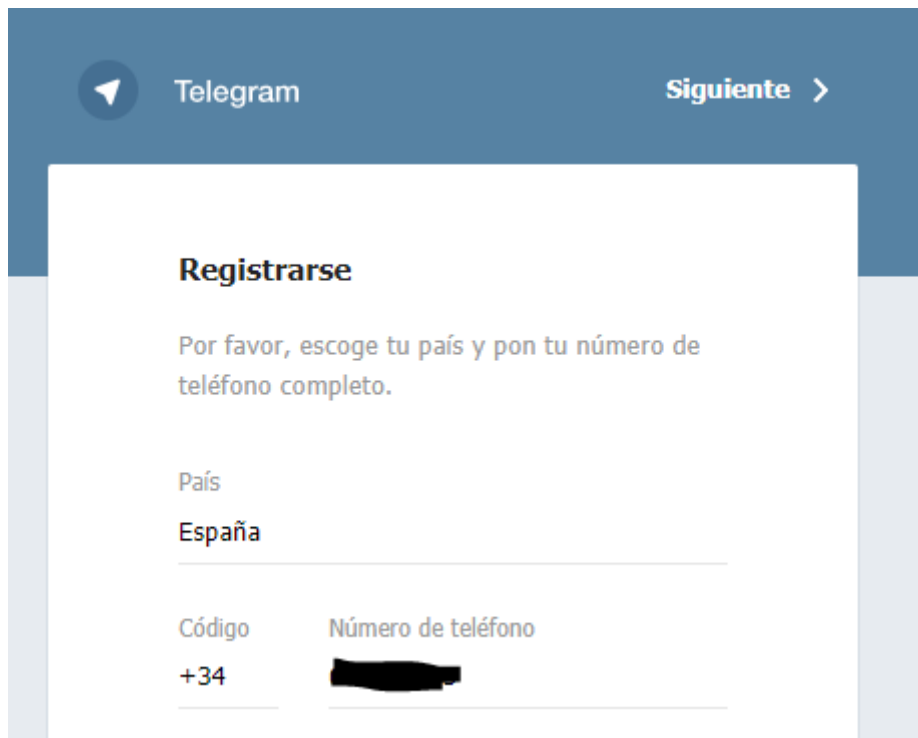
```
localhost/api/examples/example1.php
localhost/api/examples/example1.php

Connection attempt #1 to 192.168.93.10:8728... <<< [6] /login <<< [11] =name=admin <<< [10] =password=>>> [5/5] bytes read. >>> [5, 1]!done
Connected... <<< [18] /ip/service/getall >>> [3/3] bytes read. >>> [3, 291]!re >>> [7/7] bytes read. >>> [7, 283]=id=*0 >>> [12/12] bytes read. >>> [12,
270]=name=telnet >>> [8/8] bytes read. >>> [8, 261]=port=23 >>> [9/9] bytes read. >>> [9, 251]=address=>>> [14/14] bytes read. >>> [14,
236]=invalid=false >>> [15/15] bytes read. >>> [15, 220]=disabled=false >>> [3/3] bytes read. >>> [3, 215]!re >>> [7/7] bytes read. >>> [7, 207]=id=*1
>>> [9/9] bytes read. >>> [9, 197]=name=ftp >>> [8/8] bytes read. >>> [8, 188]=port=21 >>> [9/9] bytes read. >>> [9, 178]=address=>>> [14/14] bytes
read. >>> [14, 163]=invalid=false >>> [15/15] bytes read. >>> [15, 147]=disabled=false >>> [3/3] bytes read. >>> [3, 142]!re >>> [7/7] bytes read. >>> [7,
134]=id=*2 >>> [9/9] bytes read. >>> [9, 124]=name=www >>> [8/8] bytes read. >>> [8, 115]=port=80 >>> [9/9] bytes read. >>> [9, 105]=address=>>>
[14/14] bytes read. >>> [14, 90]=invalid=false >>> [15/15] bytes read. >>> [15, 74]=disabled=false >>> [3/3] bytes read. >>> [3, 69]!re >>> [7/7] bytes
read. >>> [7, 61]=id=*4 >>> [9/9] bytes read. >>> [9, 51]=name=ssh >>> [8/8] bytes read. >>> [8, 42]=port=22 >>> [9/9] bytes read. >>> [9,
32]=address=>>> [14/14] bytes read. >>> [14, 17]=invalid=false >>> [15/15] bytes read. >>> [15, 1]=disabled=false >>> [3/3] bytes read. >>> [3, 347]!re
>>> [7/7] bytes read. >>> [7, 339]=id=*6 >>> [13/13] bytes read. >>> [13, 325]=name=www-ssl >>> [9/9] bytes read. >>> [9, 315]=port=443 >>> [9/9]
bytes read. >>> [9, 305]=address=>>> [17/17] bytes read. >>> [17, 287]=certificate=none >>> [13/13] bytes read. >>> [13, 273]=invalid=true >>> [14/14]
bytes read. >>> [14, 258]=disabled=true >>> [3/3] bytes read. >>> [3, 253]!re >>> [7/7] bytes read. >>> [7, 245]=id=*7 >>> [9/9] bytes read. >>> [9,
235]=name=api >>> [10/10] bytes read. >>> [10, 224]=port=8728 >>> [9/9] bytes read. >>> [9, 214]=address=>>> [14/14] bytes read. >>> [14,
199]=invalid=false >>> [15/15] bytes read. >>> [15, 183]=disabled=false >>> [3/3] bytes read. >>> [3, 178]!re >>> [7/7] bytes read. >>> [7, 170]=id=*8
>>> [12/12] bytes read. >>> [12, 157]=name=winbox >>> [10/10] bytes read. >>> [10, 146]=port=8291 >>> [9/9] bytes read. >>> [9, 136]=address=>>>
[14/14] bytes read. >>> [14, 121]=invalid=false >>> [15/15] bytes read. >>> [15, 105]=disabled=false >>> [3/3] bytes read. >>> [3, 100]!re >>> [7/7] bytes
read. >>> [7, 92]=id=*9 >>> [13/13] bytes read. >>> [13, 78]=name=api-ssl >>> [10/10] bytes read. >>> [10, 67]=port=8729 >>> [9/9] bytes read. >>> [9,
57]=address=>>> [17/17] bytes read. >>> [17, 39]=certificate=none >>> [14/14] bytes read. >>> [14, 24]=invalid=false >>> [15/15] bytes read. >>> [15,
8]=disabled=false >>> [5/5] bytes read. >>> [5, 1]!done Array ( [0] => Array ( [.id] => *0 [name] => telnet [port] => 23 [address] => [invalid] => false
[disabled] => false ) [1] => Array ( [.id] => *1 [name] => ftp [port] => 21 [address] => [invalid] => false [disabled] => false ) [2] => Array ( [.id] => *2
[name] => www [port] => 80 [address] => [invalid] => false [disabled] => false ) [3] => Array ( [.id] => *4 [name] => ssh [port] => 22 [address] =>
[invalid] => false [disabled] => false ) [4] => Array ( [.id] => *6 [name] => www-ssl [port] => 443 [address] => [certificate] => none [invalid] => true
[disabled] => true ) [5] => Array ( [.id] => *7 [name] => api [port] => 8728 [address] => [invalid] => false [disabled] => false ) [6] => Array ( [.id] => *8
[name] => winbox [port] => 8291 [address] => [invalid] => false [disabled] => false ) [7] => Array ( [.id] => *9 [name] => api-ssl [port] => 8729 [address]
=> [certificate] => none [invalid] => false [disabled] => false ) ) Disconnected... Disconnected...
```

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

BOT DE TELEGRAM

Primero debemos registrarnos en Telegram



The image shows the Telegram registration interface. At the top, there's a blue header with the Telegram logo and a 'Siguiente >' button. Below this, a white card contains the title 'Registrarse' and the instruction 'Por favor, escoge tu país y pon tu número de teléfono completo.' There are two input fields: 'País' with 'España' selected, and 'Número de teléfono' with a masked number starting with '+34'.

Telegram **Siguiente >**

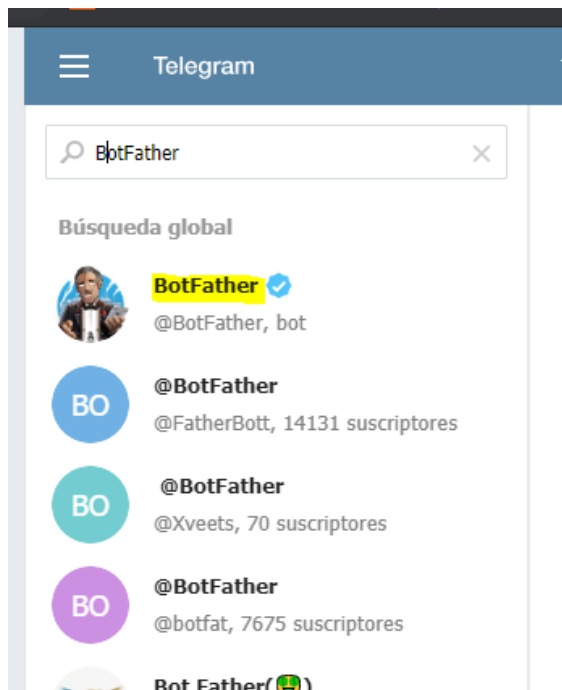
Registrarse

Por favor, escoge tu país y pon tu número de teléfono completo.

País
España

Código Número de teléfono
+34 [Redacted]

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




Crearemos un nuevo bot

BotFather bot

JO

Jose
/newbot

18:05:47




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

18:05:47

JO

Jose
Mikrotik API - Bot 1

18:06:03




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

18:06:03

JO

Jose
mikrotikBot

18:06:41



BotFather
Sorry, this username is already taken. Please try something different.

18:06:41



JO


Jose
Mikrotik_Jose_Bot










18:07:00

JO

Escribe un mensaje...







ENVIAR

Buscamos al bot

Telegram

Mikrotik_Jose_Bot

M1

Mikrotik API - Bot 1

Tú: /start

19:00

Mensajes

BotFather

Done! Congratulations on your...

18:07

BotFather

Tú: Mikrotik_Jose_Bot

18:07

JO

Jose

Mikrotik_Jose_Bot

18:07:00

BotFather

Done! Congratulations on your new bot. You will find it at t.me/Mikrotik_Jose_Bot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

110780718: [REDACTED]

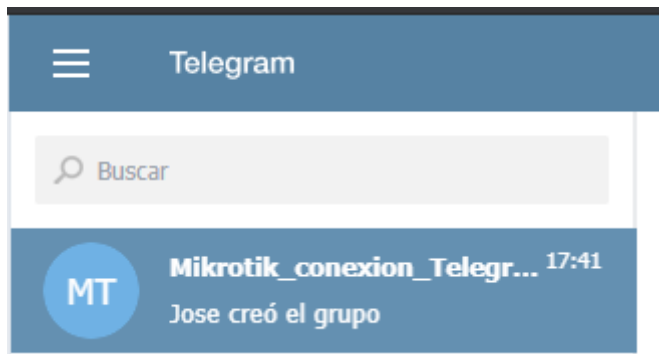
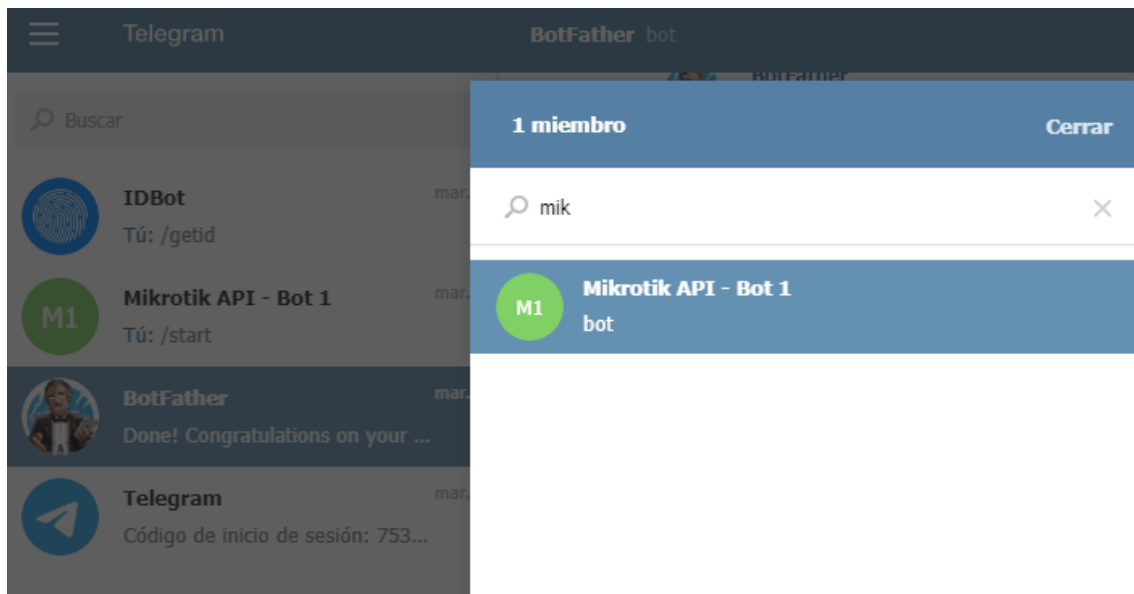
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: <https://core.telegram.org/bots/api>

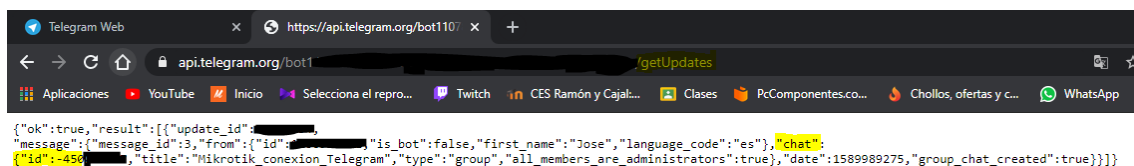
18:07:00

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](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               TTTTTTTTTT   KKK
MMMM  MMM   KKK               TTTTTTTTTT   KKK
MMM MMMM MMM III KKK KKK RRRRRR 000000 TTT III KKK KKK
MMM MM  MMM III KKKKK RRR RRR 000 000 TTT III KKKKK
MMM   MMM III KKK KKK RRRRRR 000 000 TTT III KKK KKK
MMM   MMM III KKK KKK RRR RRR 000000 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.

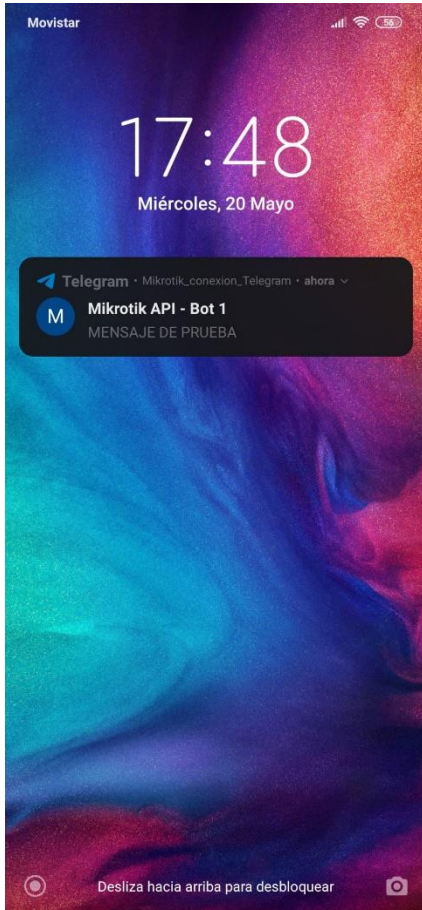
Current installation "software ID": OXYZ-CFZR
Please press "Enter" to continue!

[admin@MikroTik] > /tool fetch url="https://api.telegram.org/bot10[REDACTED]
sendMessage?chat_id=-4500[REDACTED]&text=MENSAJE DE PRUEBA"
status: finished
downloaded: 0KiB-C-z pause]
total: 0KiB
duration: 1s

[admin@MikroTik] > █
```

if para buscar

17:48
20/05/2020



Vemos que ha llegado el mensaje correctamente

Programado de alertas

Ahora mediante un script programaremos el Mikrotik para que envíe 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: 2KiB[-z pause]
total: 2KiB
duration: 0s

[admin@MikroTik] >
```

Abrimos el script usando ' import IXF_script.clien '

The screenshot shows the Mikrotik WinBox interface. On the left, the 'Terminal' window displays the output of the script import command, showing a ASCII art logo for 'www.internetporfichas.com' and a list of services. Below the terminal, the 'Log' window shows a list of system events, including the successful import of the script. On the right, the 'Netwatch' configuration window is visible, showing a single host '8.8.8.8' with an interval of '00:00:40' and a timeout of '1000'. The status is 'up'.

Terminal

```
#####
#          #          #
#          #          #
#          #          #
#####

www.internetporfichas.com

CONFIGURACIONES (HOTSPOT, BALANCEOS QOS, ETC
SCRIPTS, SOFTWARE, PLANTILLAS Y MAS
VENTA DE EQUIPOS UBIQUITI, MIKROTIK, MIMOSA, RF

Para soporte Escribenos en nuestro chat
de nuestra pagina web
o llamenos al 9531064743

[admin@MikroTik] > import IXF_script.clien
```

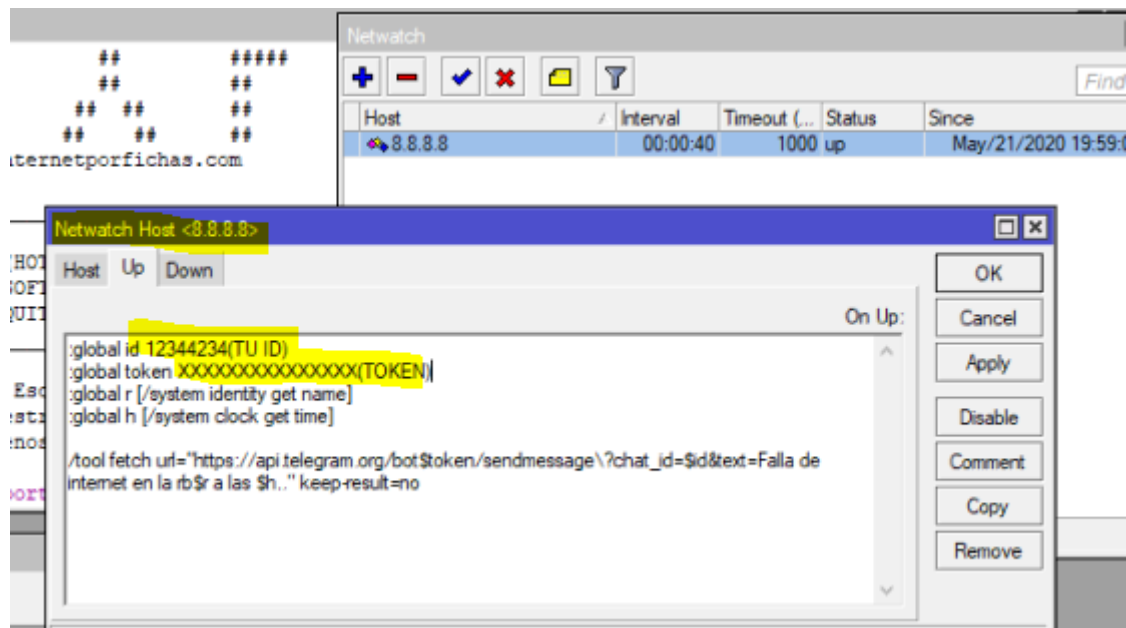
Netwatch

Host	Interval	Timeout (...)	Status
8.8.8.8	00:00:40	1000	up

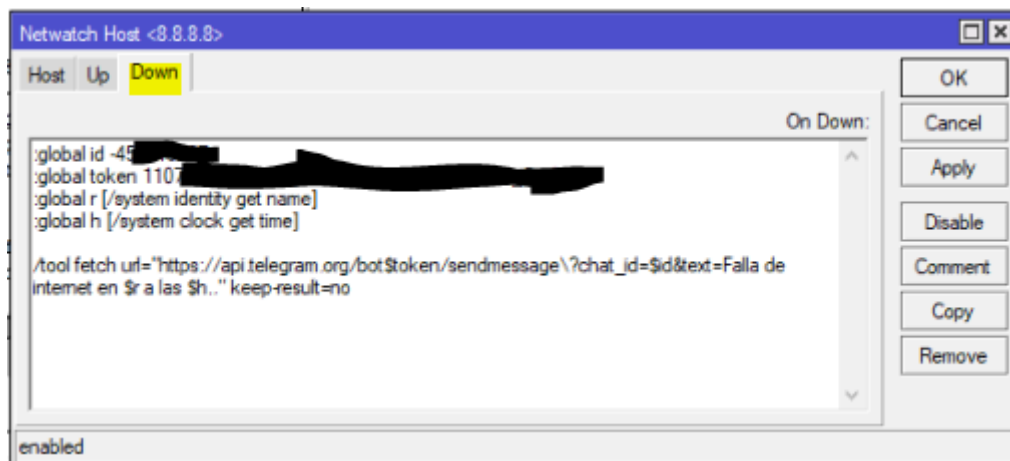
Log

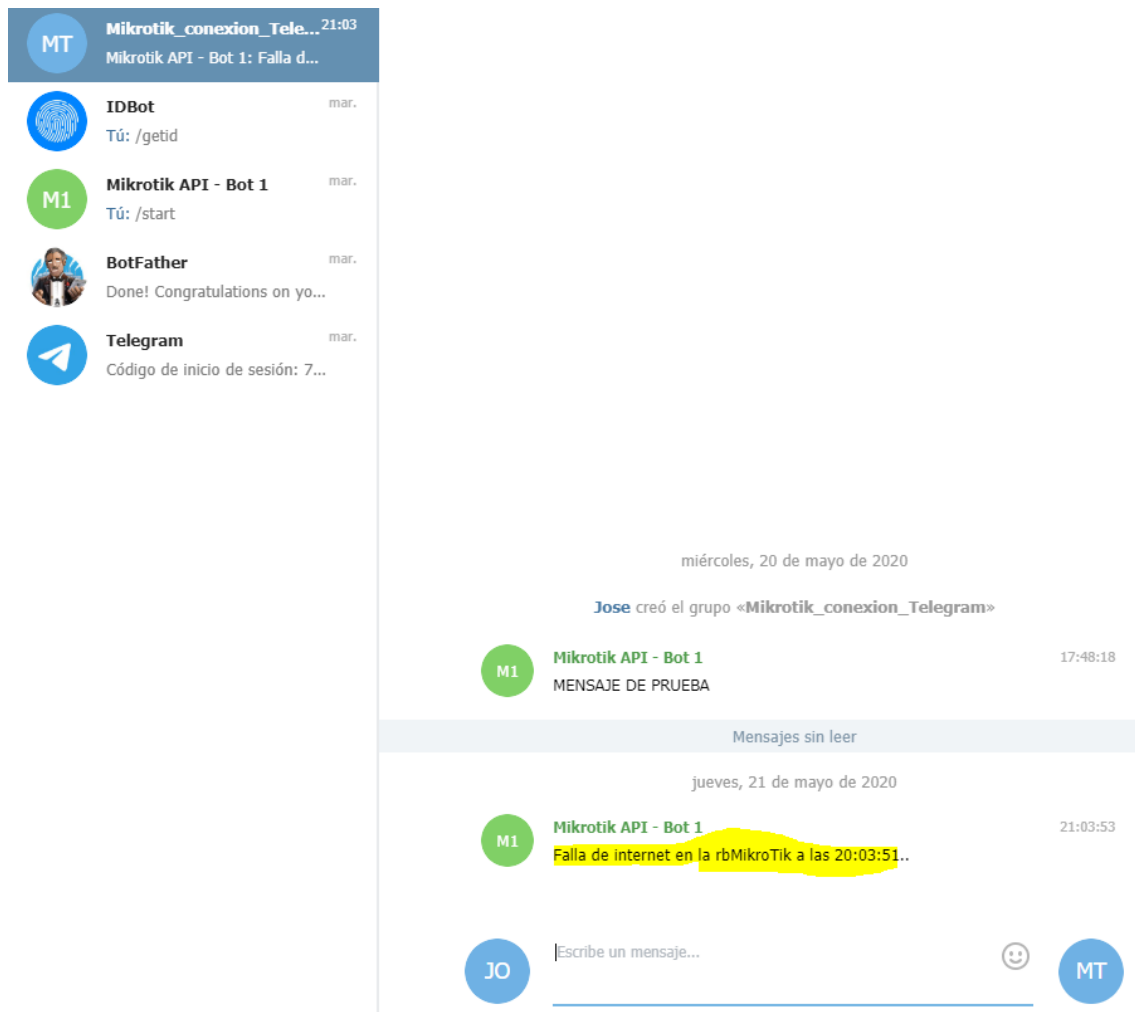
Time	Level	Category	Message
May/21/2020 19:58:56	memory	script, warning	VISITARNOS EN WWW.INTERNETPORFICHAS.COM Y REGALARNOS UN LIKE SI QUIERES MAS SCRIPT GRATIS
May/21/2020 19:58:58	memory	script, warning	ING ROBERTH BARRERA
May/21/2020 19:59:00	memory	system, info	changed system note settings by admin
May/21/2020 19:59:02	memory	system, info	monitoring new host by admin
May/21/2020 19:59:02	memory	system, info, account	user admin logged out from 192.168.93.128 via winbox
May/21/2020 19:59:04	memory	script, warning	LISTO FELICIDADES YA SE HA CARGADO EL ESCRIPT CON EXI
May/21/2020 19:59:04	memory	script, error	RECUERDA QUE DEBES CARGAR TU TOKEN Y ID
May/21/2020 19:59:06	memory	system, info, account	user admin logged in from 192.168.93.128 via winbox
May/21/2020 19:59:06	memory	system, info, account	user admin logged in from 192.168.93.128 via telnet
May/21/2020 19:59:12	memory	smb, info	created new share: pub

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.

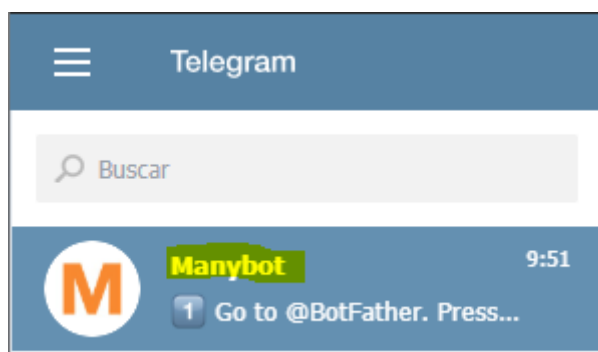




Vemos que funciona correctamente

CREAR COMANDOS PERSONALIZADOS EN EL BOT DE INSTAGRAM

Lo primero es buscar a ManyBot



Ahora añadimos el token del bot

lunes, 25 de mayo de 2020

¿Qué puede hacer este bot?

🔥 Manybot lets you create your own bots.

- ✓ Send messages to your subscribers.
- ✓ Create custom commands.
- ✓ Create custom menus and submenus.
- ✓ Autopost from RSS, Twitter, VK and YouTube.
- ✓ Custom forms for orders and feedback.
- ✓ 2'000'000'000+ messages sent.
- ✓ 100'000 bots created.

Press Start 🖱️ to create your first bot!



Jose

9:50:07

/start



Manybot

9:50:09

Welcome! Please choose your language.



Jose

9:50:22

/addbot

/addbot

9:51:57



Manybot

9:51:58

- 1 Go to [@BotFather](#). Press his name to do that and then press 'Send Message' if it's needed.
- 2 Create a new bot with him. To do this use the 'newbot' command inside [@BotFather](#).
- 3 Copy the API token that [@BotFather](#) will give you.
- 4 Come back to [@Manybot](#) and send the copied API token here.



11075 [redacted]



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


M1


Mikrotik API - Bot 1


10:08:04

 Posting
[/newpost](#) - send a post to your subscribers
[/schedule](#) - view your scheduled posts
[/subscribers](#) - see how many subscribers you have

 Commands and Bot Menu
[/commands](#) - manage custom commands and user menues

 Settings
[/botlang](#) - set up your bot language
[/setdescription](#) - change the description of your bot
[/autoposting](#) - set up autoposting from Twitter, YouTube, RSS, VK
[/channels](#) - automate posting from your bot to a Telegram channel
[/admins](#) - manage admins of your bots



 Tutorials
[/tutorials](#) - discover how to create and manage your bots

 Other
[/help](#) - this menu










Have further questions? You can always reach us at
hello@manybot.io

JO

Escribe un mensaje...

M1

ENVIAR

Send New Post to Subscribers

Custom Commands

Form Replies

Settings

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

JO Jose 10:22:30
Save


M1 Mikrotik API - Bot 1 10:22:32
Custom command /saludo was successfully created.

10:22:32
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 Hola!   **M1**

Ahor vamos al botfather para especificarle los comandos

JO Jose 10:30:37
/setcommands

 **BotFather** 10:30:39
Choose a bot to change the list of commands.

JO Jose 10:32:40
@Mikrotik_Jose_Bot


 **BotFather** 10:32:42
OK. Send me a list of commands for your bot. Please use this format:

command1 - Description
command2 - Another description

Send /empty to keep the list empty.

JO Jose 10:33:02
prueba1 - mensaje de prueba
saludo - mensaje de saludo

Mensajes sin leer

 **BotFather** 10:33:04
Success! Command list updated. /help

Ahora probamos en el bot



También funciona en el grupo



CONTROLAR MIKROTIK DESDE TELEGRAM POR COMANDOS

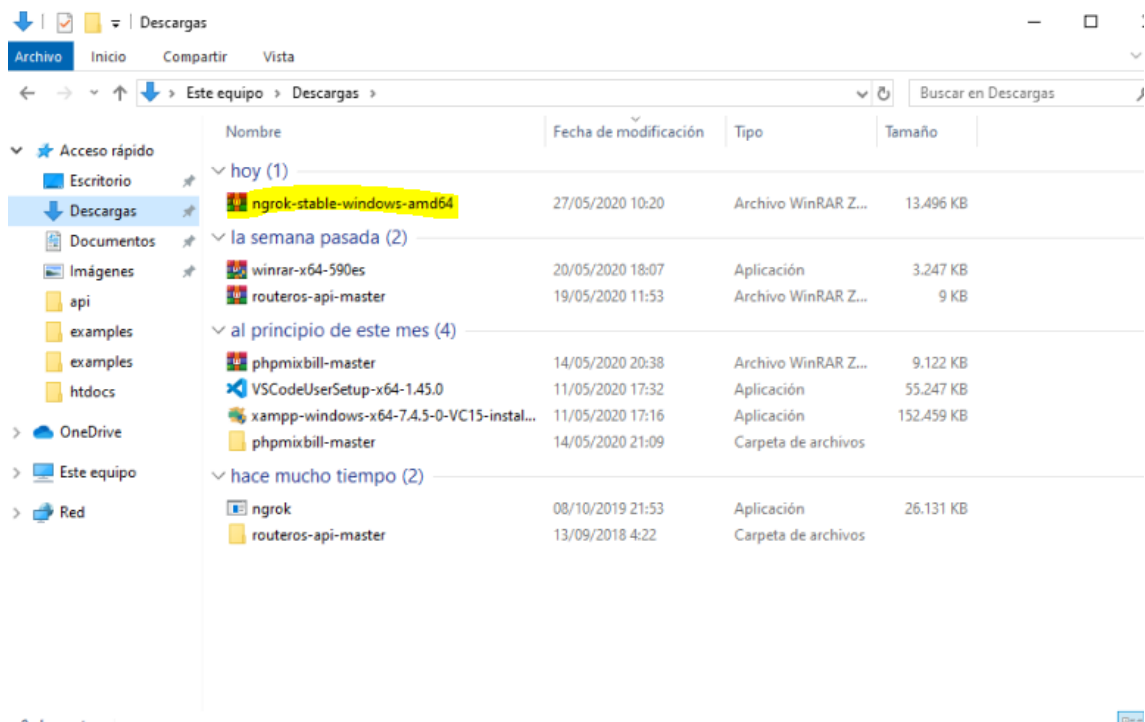
Lo primero que debemos hacer es usar crear un programa con php usando una API de Mikrotik para resetear el Mikrotik.

```
example1.php X
C: > xampp > htdocs > api > examples > example1.php
1  <?php
2
3  require('../routeros_api.class.php');
4
5  $API = new RouterosAPI();
6
7  $API->debug = true;
8
9  if ($API->connect('192.168.93.10', 'admin', '')) {
10
11      $API->write('/system/reboot');
12
13      $READ = $API->read(false);
14      $ARRAY = $API->parseResponse($READ);
15
16      print_r($ARRAY);
17
18      $API->disconnect();
19
20  }
21
22  ?>
23
```

Usando esta [API](#) creamos un programa para que reinicie el Mikrotik.

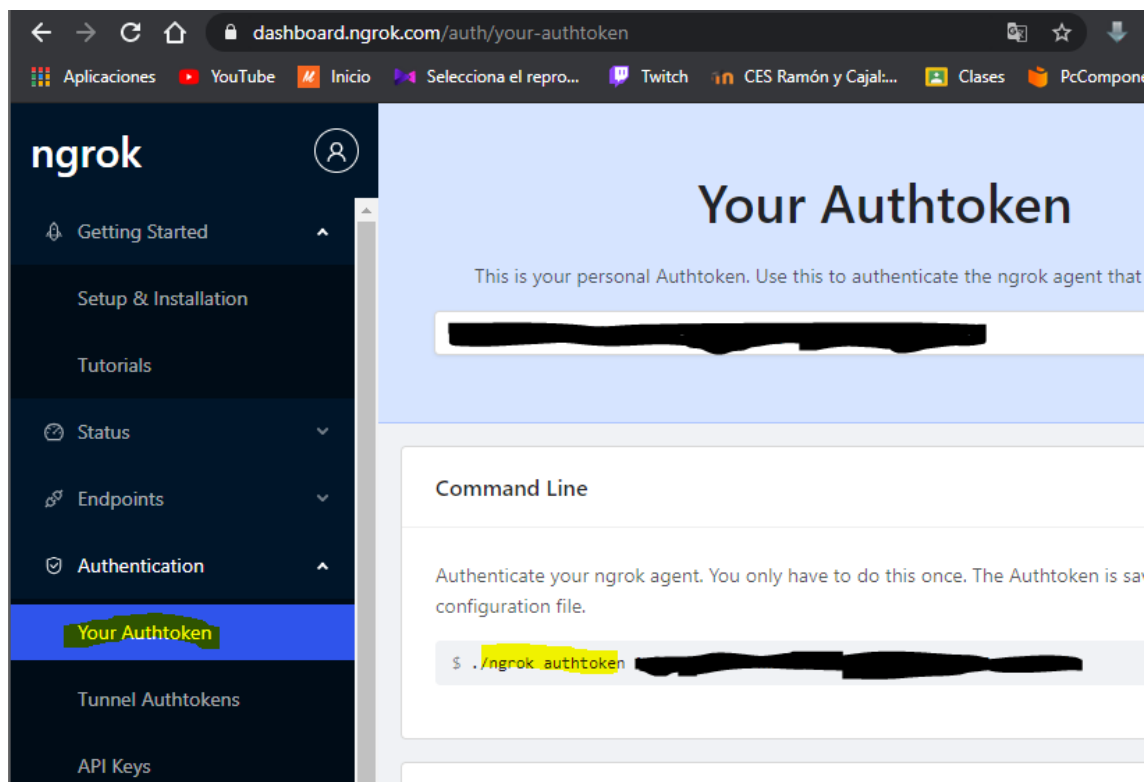
Ahora debemos subir nuestra pagina en localhost a internet para que se pueda acceder desde telegram, para ello usaremos [ngrok](#) :

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`

```
C:\Users\w10\Downloads\ngrok.exe
ngrok http -subdomain=baz 8080 # port 8080 available at baz.ngrok.io
ngrok http foo.dev:80 # tunnel to host:port instead of localhost
ngrok http https://localhost # expose a local https server
ngrok tcp 22 # tunnel arbitrary TCP traffic to port 22
ngrok tls -hostname=foo.com 443 # TLS traffic for foo.com to port 443
ngrok start foo bar baz # start tunnels from the configuration file

VERSION:
  2.3.35

AUTHOR:
  inconshreveable - <alan@ngrok.com>

COMMANDS:
  authtoken  save authtoken to configuration file
  credits    prints author and licensing information
  http       start an HTTP tunnel
  start      start tunnels by name from the configuration file
  tcp        start a TCP tunnel
  tls        start a TLS tunnel
  update     update ngrok to the latest version
  version    print the version string
  help       Shows a list of commands or help for one command

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\w10\Downloads>ngrok authtoken 1cl'
Authtoken saved to configuration file: C:\Users\w10\.ngrok2\ngrok.yml
C:\Users\w10\Downloads>
```

Ahora abriremos una conexión a nuestro server

ngrok http 80

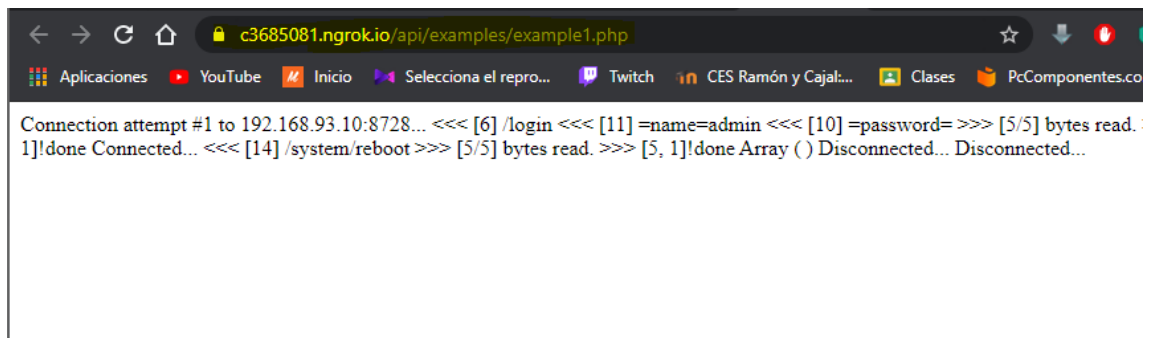
```
C:\Users\w10\Downloads>ngrok http 80
```

```
C:\Users\w10\Downloads\ngrok.exe - ngrok http 80
ngrok by @inconshreveable

Session Status      online
Account             [REDACTED].com (Plan: Free)
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://fe7a7b1b.ngrok.io -> http://localhost:80
Forwarding           https://fe7a7b1b.ngrok.io -> http://localhost:80

Connections
  ttl    opn    rt1    rt5    p50    p90
    0      0     0.00   0.00   0.00   0.00
```

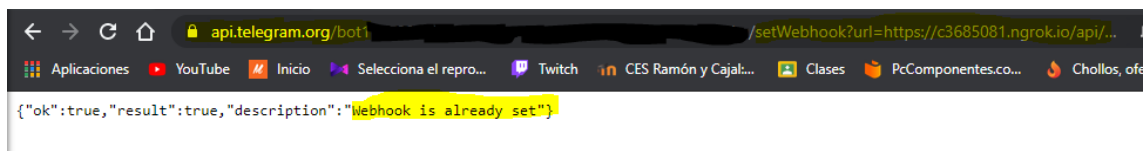
Deberemos usar la dirección https para que acceda al Mikrotik y lo reinicie



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:







-  **Jose** 10:37:36
/setcommands
-  **BotFather** 10:37:38
Choose a bot to change the list of commands.
-  **Jose** 10:37:39
@Mikrotik_Jose_Bot
-  **BotFather** 10:37:40
OK. Send me a list of commands for your bot. Please use this format:

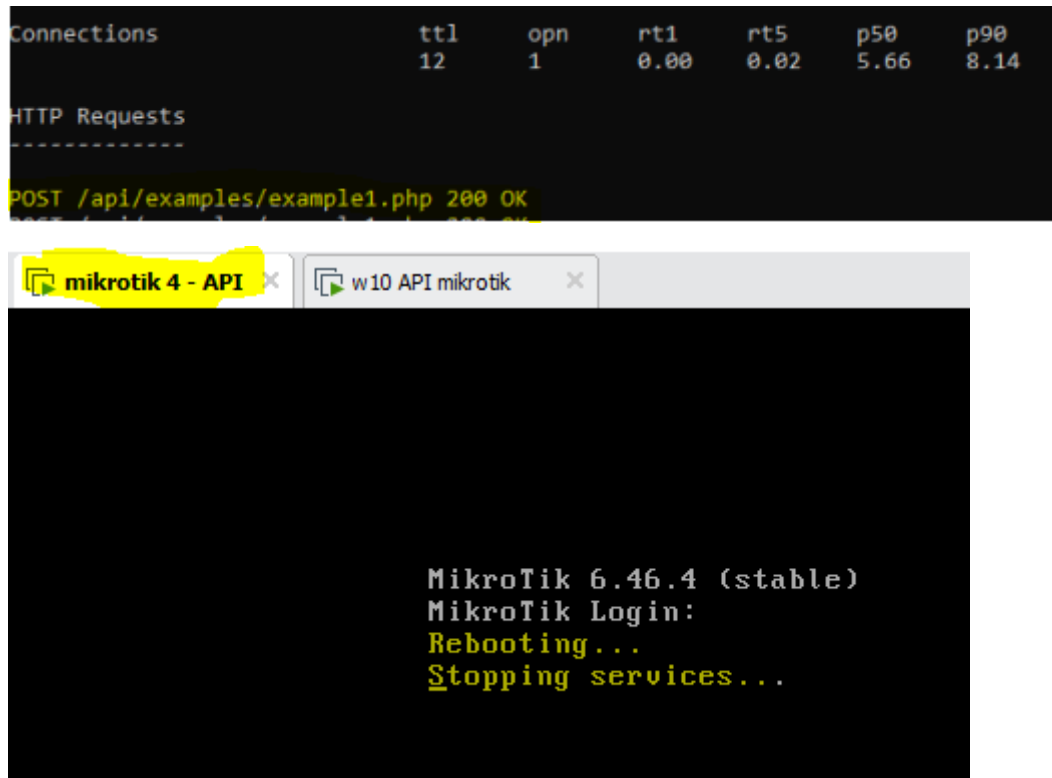
command1 - Description
command2 - Another description

Send [/empty](#) to keep the list empty.
-  **Jose** 10:43:47
[resetear - resetea](#)
-  **BotFather** 10:43:49
Success! Command list updated. [/help](#)

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

miércoles, 27 de mayo de 2020

-  **Jose** 10:32:59
/start
-  **Jose** 11:37:18
[/resetear](#)
-  Escribe un mensaje...    M1



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:AAGfyusehJynVBjszCIRFtQGeQi_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/>