



The better PHP API

(По-доброто PHP API)

От Васил Рангелов a.k.a. boen_robot
boen.robot@gmail.com

Какво е това „API“?

- ▶ В програмирането:
 - ▶ “**A**pplication **P**rogramming **I**nterface” – Набор от операции и техните входове и изходи, с едно общо предназначение и достъпни за програмиста.
- ▶ В MikroTik RouterOS:
 - ▶ Клиент/сървър протокол от слой 7 (ала HTTP, FTP, SSH и т.н.)
 - ▶ Предназначен за машини, не за хора
 - ▶ Строг, структуриран с липсващи „човешки“ удобства
 - ▶ За да се избегнат обърквания, протокола ще се води като “RouterOS API”.

За какво може да се използва?

- ▶ Създаване на собствени интерфейси за управление
- ▶ Интеграция на RouterOS с външни приложения, пр.
 - ▶ Billing системи
 - ▶ Приложения за самообслужване
 - ▶ Мониторинг инструменти
- ▶ С една дума: Автоматизация.

SSH vs. RouterOS API

4

- ▶ И двете биха могли да се използват за автоматизация
- ▶ SSH е предназначен за човешко ползване
 - ▶ „Козметични“ промени от една версия на RouterOS до друга могат да предизвикат проблеми с инструменти за автоматизация
 - ▶ Преименуване/премахване на команди/аргументи/свойства може да причини не само **трудни** за засичане грешки, но евентуално и щети
- ▶ RouterOS API е предназначен за машинно ползване
 - ▶ Единствено промени по самият протокол биха могли да причинят затруднения
 - ▶ **Лесни** за засичане грешки и евентуално възстановяване без щети.

Езици с RouterOS API клиенти за общо предназначение

- ▶ PHP (3)
- ▶ Java (2)
- ▶ .NET (C# (3), VB.NET (1))
- ▶ Python (2)
- ▶ Node.js (1)
- ▶ Perl (2)
- ▶ Ruby (1)
- ▶ Delphi (2)
- ▶ C/C++ (4)
- ▶ Erlang (1)
- ▶ ActionScript (1)

Сравнение на PHP RouterOS API клиенти

Клиент	API PHP class (Denis Basta)	RouterOS PHP class (ayufan)	API PHP package (boen_robot)
Изисква PHP версия	5.2+	5.2+	5.3+
Суров протоколен В/И	Да	Не	Да
Улеснения с общо предназначение	Един метод (comm())	Не	Набор от класове
CRUD и др. улеснения	Не	Да	Да (1 клас)
Асинхронни заявки	Не	Само с функции	Да
Скриптове	Не	Псевдо	Да
Постоянни връзки	Не	Не	Да
Емулация на команден ред	Не	Не	Псевдо

Подготвяне на рутера за RouterOS API

- ▶ За RouterOS по-стари от v6.0, протокола трябва да се ВКЛУЧИ С
 - ▶ `/ip service enable api`
- ▶ Протокола е включен по подразбиране от RouterOS v6.0
- ▶ RouterOS потребителят трябва да има "api" позволение
 - ▶ Началните потребителски групи го имат; внимавайте ако използвате собствени потребителски групи
- ▶ Ако се използват „input“ и „output“ веригите на защитната стена, трябва да се позволят **ВХОДЯЩИ** връзки **ДО** TCP порт 8728, както и **ИЗХОДЯЩИ ОТ** TCP порт 8728.

Подготвяне на клиентското устройство за RouterOS API

- ▶ Инсталирайте всичко необходимо за създаване и изпълнение на програми на избраният език
 - ▶ За PHP, това значи да инсталирате самият PHP интерпретатор и евентуално да го закачите към web сървър
 - ▶ Най-лесно с цялостни пакети като XAMPP: <http://apachefriends.org/>
 - ▶ За приложения с команден интерфейс, самият PHP от <http://php.net/> е достатъчен
- ▶ Позволете на съответните изпълними файлове ("php" и може би "httpd") да правят изходящи връзки към TCP порт 8728
 - ▶ За Windows, отивате в защитната стена с Win+R, и изпълнявате "wf.msc".

Първо използване на PHP RouterOS пакетът (бърз начин)

- ▶ Изтегляте PHAR файла от https://pear2.github.io/Net_RouterOS/
- ▶ Създавате файл в “htdocs” в директорията на Apache, наречен пр. “roctest.php”, със следното съдържание:

```
<?php
use PEAR2\Net\RouterOS;
require_once 'PEAR2_Net_RouterOS-1.0.0b5.phar';

try {
    $client = new RouterOS\Client('192.168.0.1', 'admin', 'password');
    echo 'OK';
} catch (Exception $e) {
    die($e);
}
```

- ▶ До-настройте пътеката до “.phar” файла, IP-то и идентификационните данни на RouterOS
- ▶ Отворете в web браузер <http://localhost/roctest.php>.

Алтернативни методи на инсталация

- ▶ Разархивиране на TGZ или ZIP файла от https://pear2.github.io/Net_RouterOS/
 - ▶ Включвате „src/PEAR2/Autoload.php”
- ▶ Чрез Composer:
 - ▶ `composer require pear2/net_routeros`
 - ▶ Включвате „vendor/autoload.php”
- ▶ Чрез PEAR:
 - ▶ `pear channel-discover pear2.php.net`
 - ▶ `pear install -a pear2/PEAR2_Net_RouterOS-alpha`
 - ▶ Включвате „PEAR2/Autoload.php”
- ▶ Чрез Pyrus:
 - ▶ `pyrus install PEAR2_Net_RouterOS-alpha`
 - ▶ Включвате „PEAR2/Autoload.php“.

Диагностика на проблеми (етап 1: среда)

- ▶ Отваряте команден ред
 - ▶ В Windows, Win+R, и пишете "cmd"
- ▶ Стартирате PHAR файла с PHP
 - ▶ Windows:
 - ▶ "D:\път\до\php.exe" "D:\път\до\PEAR2_Net_RouterOS.phar"
 - ▶ UNIX:
 - ▶ "/път/до/php" "/път/до/PEAR2_Net_RouterOS.phar"
- ▶ Оглеждате изхода за предупреждения и грешки
- ▶ Забележка: За краткост, горното ще се съкрати на "php PEAR2_Net_RouterOS.phar" в следващите слайдове.

Диагностика на проблеми (етап 2: връзка)

- ▶ Стартирате PHAR файла с IP-то на рутера като аргумент
 - ▶ `php PEAR2_Net_RouterOS.phar 192.168.0.1`
- ▶ Ако използвате различен порт за RouterOS API (пр. 443):
 - ▶ `php PEAR2_Net_RouterOS.phar -p 443 192.168.0.1`
- ▶ Оглеждате изхода за грешки и възможни решения
 - ▶ Всичко е наред ако не виждате нищо и можете да пишете
 - ▶ Въвеждате `"/quit"` (без кавичките), и натискате Enter два пъти за изход.

Диагностика на проблеми (етап 3: логин)

- ▶ Стартирате PHAR файла с потребителското име и парола за RouterOS в края, пр.
 - ▶ `php PEAR2_Net_RouterOS.phar 192.168.0.1 "admin" "password"`
- ▶ Оглеждате изхода за грешки и възможни решения
 - ▶ Всичко е наред ако не виждате нищо и можете да пишете
 - ▶ Въвеждате `"/quit"` (без кавичките), и натискате Enter два пъти за изход.

Пример: Torch за 4 секунди

```
<?php
use PEAR2\Net\RouterOS;
require_once 'PEAR2_Net_RouterOS-1.0.0b5.phar';

$client = new RouterOS\Client('192.168.0.1', 'admin', 'password');

header('Content-Type: text/plain');

$request = new RouterOS\Request('/tool torch duration=4');
$request->setArgument('interface', $_GET['nic']);
foreach ($client->sendSync($request) as $response) {
    foreach ($response as $name => $value) {
        echo "{$name}: {$value}\n";
    }
    echo "====\n";
}
```

Пример: Извеждане на логът на RouterOS

```
<?php
use PEAR2\Net\RouterOS;
require_once 'PEAR2_Net_RouterOS-1.0.0b5.phar';

$client = new RouterOS\Client('192.168.0.1', 'admin', 'password');

header('Content-Type: text/plain');

$util = new RouterOS\Util($client);
foreach ($util->setMenu('/log')->getAll() as $item) {
    echo $item('time') . ' ' .
        $item('topics') . ' ' .
        $item('message') . "\n";
}
```

Пример: Изпълнение на скрипт с параметри

```
<?php
use PEAR2\Net\RouterOS;
require_once 'PEAR2_Net_RouterOS-1.0.0b5.phar';

$client = new RouterOS\Client('192.168.0.1', 'admin', 'password');

header('Content-Type: text/plain');

$util = new RouterOS\Util($client);
$util->exec('
    /ip dhcp-client lease
        make-static [find address=$address]
        comment [find address=$address] $name
    /log info "User $name now has the static IP $address"
    ',
    array(
        'name'      => $_GET['user'],
        'address'   => $_GET['ip']
    )
);
```


Поддръжка

- ▶ Документация и още примери
 - ▶ GitHub wiki: https://github.com/pear2/Net_RouterOS/wiki
 - ▶ MikroTik wiki: http://wiki.mikrotik.com/wiki/API_PHP_package
- ▶ За въпроси
 - ▶ MikroTik форум (на английски): <http://forum.mikrotik.com/>
- ▶ Бъгове и предложения за клиента
 - ▶ GitHub: https://github.com/pear2/Net_RouterOS/issues
- ▶ Бъгове и предложения за протокола
 - ▶ MikroTik поддръжка (на английски): support@mikrotik.com.

Често задавани въпроси (технически)

- ▶ Може ли API приложение да работи директно от рутера?
 - ▶ Не.
- ▶ Мога ли да логвам hotspot потребители с API протокола?
 - ▶ Не. Не с API протокола, не с SSH, не дори чрез Winbox.
 - ▶ Hotspot потребители могат да се логват ЕДИНСТВЕНО когато ТЕХНИТЕ устройства направят HTTP(S) заявка до рутера с техните потребителско име и парола
- ▶ Ако рутера ми е скрит зад NAT?
 - ▶ Решенията са като за при всеки друг TCP протокол.
 - ▶ VPN?
- ▶ Ами криптиране?
 - ▶ API протоколът има TLS криптиран вариант на порт 8729
 - ▶ PEAR_Net_RouterOS го поддържа, но заради проблеми в PHP, за сега такива връзки са **МНОГО** нестабилни.

Често задавани въпроси (социални)

- ▶ Защо още един PHP клиент?
 - ▶ Другите PHP клиенти не ми харесват
 - ▶ Както и доста не-PHP клиенти, но по-добре нещо от колкото нищо, та...
 - ▶ Протоколът е лесен за реализация от бакалаври по Информатика като мен, та в крайна сметка „защо не?“
- ▶ Защо PEAR(2)?
 - ▶ PEAR не е framework, ами колекция пакети, следващи общ кодов стандарт
 - ▶ Нещо като „печат за качество“, макар вече не толкова широко приет
- ▶ Защо текущата версия е beta?
 - ▶ За PEAR2, „beta“ **НЕ** значи „възможни внезапни грешки“
 - ▶ Означава „възможни несъвместими промени в следващата версия; Прегледайте внимателно списъкът с промените при актуализации“
 - ▶ Подобно на това как Gmail дълго време беше beta.

Благодаря за вниманието

Въпроси



boen.robot@gmail.com