

PRESS REVIEW ARCHIVE

Digital Media Monitoring & Documentation Service

Source URL: <https://packagist.org/packages/aglavas/shodan-php-rest-api>

Archived Date: August 15, 2025 at 15:09

Document Type: Web Page Archive

Wayback Machine: https://web.archive.org/web/*/https://packagist.org/packages/aglavas/shodan-php-rest-api

Page Screenshot

The screenshot shows the Packagist website interface. At the top, there's a navigation bar with the Packagist logo, a search bar, and links for 'Composer v1 support is coming to an end', 'Browse', 'Submit', 'Create account', and 'Sign in'. Below the navigation bar, the package name 'aglavas/shodan-php-rest-api' is displayed. A message indicates that there is no license information available for the latest version (dev-master) of this package. The package description is 'PHP sdk (API wrapper) for Shodan'. On the right side, there's a 'Maintainers' section showing a list of maintainers and a 'Details' section with statistics: Installs: 4, Dependents: 0, Suggesters: 0, Security: 0, Stars: 0, Watchers: 0, and Forks: 58. Below the package description, there's a table showing the package's requirements and provides. The table has three columns: 'requires', 'requires (dev)', and 'suggests'. The 'requires' column shows 'None', 'requires (dev)' shows 'None', and 'suggests' shows 'None'. Below the table, there's a section for the package's license and a link to the README file.

Packagist The PHP Package Repository

Composer v1 support is coming to an end [Browse](#) [Submit](#) [Create account](#) [Sign in](#)

Search packages...

aglavas/shodan-php-rest-api

`composer require aglavas/shodan-php-rest-api`

There is no license information available for the latest version (dev-master) of this package.

PHP sdk (API wrapper) for Shodan

Maintainers

Details

github.com/aglavas/Shodan-PHP-RES...

Source

Installs: 4
Dependents: 0
Suggesters: 0
Security: 0
Stars: 0
Watchers: 0
Forks: 58

dev-master 2024-05-05 18:35 UTC

requires	requires (dev)	suggests
None	None	None

provides	conflicts	replaces
None	None	None

© Unknown License [5f99b9ace00894369ceeabadd364fb962d1bdd4](#)

[Andrija Glavaš](#) <aglavas11@gmail.com>


[README](#)

dev-master
dev-feature_composer_psr4

This package is auto-updated.
Last update: 2025-08-05 21:32:51 UTC

Search packages...

aglavas/shodan-php-rest-api

 composer require aglavas/shodan-php-rest-api

There is no license information available for the latest version (dev-master) of this package.

PHP sdk (API wrapper) for Shodan

Maintainers



Details

github.com/aglavas/Shodan-PHP-REST-API


Source

Installs: 4
Dependents: 0
Suggesters: 0
Security: 0
Stars: 0
Watchers: 0
Forks: 58

dev-master

2024-05-05 18:35 UTC

 Unknown License  5ff9b9ace00894369ceeabadcd364fb962d1bdb4

 Andrija Glavaš

dev-master

dev-feature_composer_psr4

This package is auto-updated.

Last update: 2025-08-05 21:32:51 UTC

README



Lean and easily extendible PHP API for Shodan.io, supporting both the free API and the paid one. Requests are auto-generated by structure that defines the API protocol.

Authors and License



Shodan-PHP-REST-API is licensed under the GNU GPL v3 and is a project sponsored by ISGroup SRL and authored by Alex Salvetti and Francesco [asci11](#) Ongaro. This software is currently used by ScadaExposure, a permanent observatory on the exposure of ICS and SCADA devices on the Internet, to generate it's datasets.

Features

- Search Shodan.
- Streaming API support for real-time consumption of Shodan data.
- Exploit search API fully implemented.

Notes

- Shodan.php is the API class: constants, shodan methods and the generation of the HTTP requests are defined here.
- The script uses PHP magic methods (<http://php.net/manual/en/language.oop5.magic.php>).
- shodan-api.php is the CLI interface, allowing to run differents commands; it also provides an how-to function.
- Our API implementation uses 3 different base URLs: Shodan API, Streaming API and Exploits API.
- Tests folder provides some examples on how to write your own search query, use the CLI `-r` flag for running them all or call one with the `-t` flag.
- If you're in search of better and more thorough documentation, please refer to Shodan's REST API documentation (<https://developer.shodan.io/api>).
- For Shodan EXPLOITS API refer to the documentation (<https://developer.shodan.io/api/exploits/rest>).
- For Shodan STREAM API refer to the documentation (<https://developer.shodan.io/api/stream>).

Usage

You can implement the class API directly in your code or experiment with the CLI. In both cases you'll need to change your API KEY in shodan-api.php or anywhere you instantiate the API object:

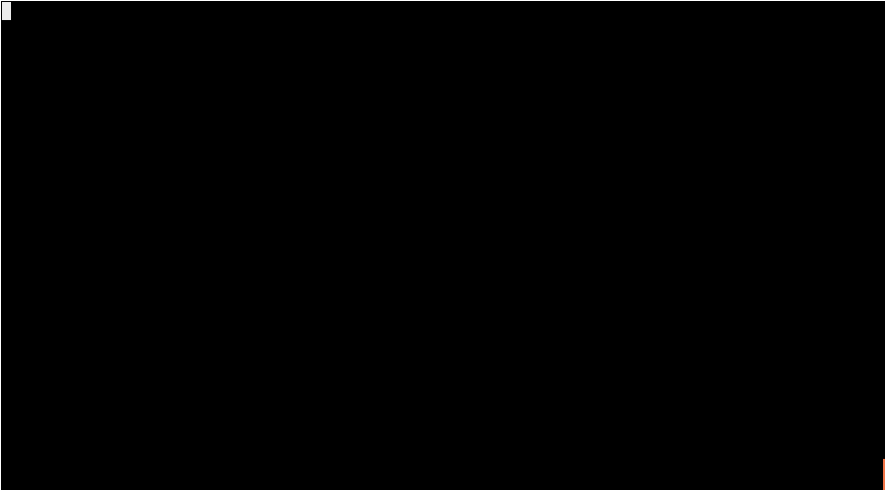
```
$key = 'Insert your API key here';
```

Following are the options:

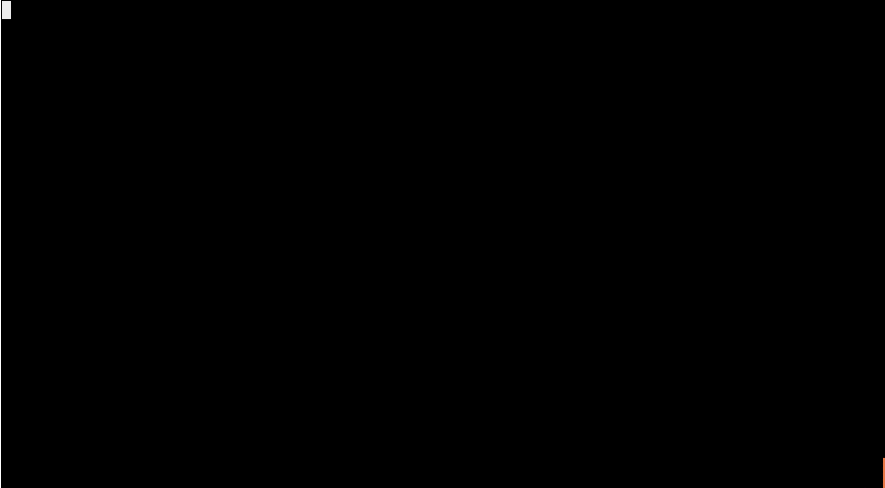
Short form	Long form	Variables
-r	--run-tests	
-t	--run-test	STRING
-m	--method	ShodanHost --ip STRING [--history BOOLEAN] [--minify BOOLEAN]
-m	--method	ShodanHostCount --query STRING [--facets STRING]
-m	--method	ShodanHostSearch --query STRING [--facets STRING]
-m	--method	ShodanHostSearchTokens --query STRING
-m	--method	ShodanPorts
-m	--method	ShodanProtocols
-m	--method	ShodanScan --ips STRING
-m	--method	ShodanScanInternet --port INTEGER --protocol STRING
-m	--method	ShodanScan_Id --id STRING
-m	--method	ShodanServices
-m	--method	ShodanQuery [--page INTEGER] [--sort STRING] [--order STRING]
-m	--method	ShodanQuerySearch --query STRING [--page INTEGER]
-m	--method	ShodanQueryTags [--size INTEGER]
-m	--method	LabsHoneyscore --ip STRING
-m	--method	Search --query STRING [--facets STRING] [--page INTEGER]
-m	--method	Count --query STRING [--facets STRING]
-m	--method	ShodanBanners
-m	--method	ShodanAsn --asn STRING
-m	--method	ShodanCountries --countries STRING
-m	--method	ShodanPorts_Stream --ports STRING

Some CLI Run Examples

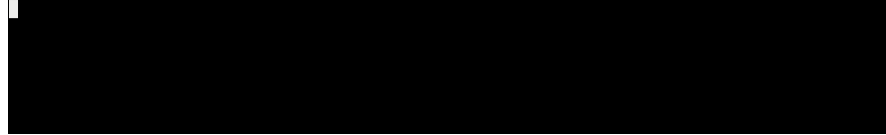
Showing usage options:



Shodan Host method on Facebook ip:



Shodan Scan request on some ips:



Shodan Scan request status:



Handle overlapping methods

Using PHP magic methods we call the method by its name and use it to generate the URL for the request. For doing that we use `preg_replace` inserting a `/` when an uppercase character is found and appending that character in lowercase.

But we found that two methods in Shodan API were overlapping with other two methods, that are: "ShodanScan" and "ShodanPorts". So we renamed "ShodanScan" given with "id" parameter in "ShodanScan_Id", and "ShodanPorts" for the stream API in "ShodanPorts_Stream".

But the URL must not have those renaming, so we eliminate the `_` and all it comes next of it for getting the job done.

You can find it at: <https://github.com/ScadaExposure/Shodan-PHP-REST-API/blob/master/src/Shodan.php#L471>

Tests class - REST API

Shodan Host (/tests/ip.php):

Return all services that have been found on the given host IP.

```
var_dump($client->ShodanHost(array(
    'ip' => '69.171.230.68', // https://www.facebook.com/
)));
```

Shodan Host Count (/tests/count.php):

Returns the total number of results that matched the query and any facet information that was requested.

```
var_dump($client->ShodanHostCount(array(
    'query' => 'Niagara Web Server',
)));
```

Shodan Host Search (/tests/search.php):

Search Shodan using the same query syntax as the website and use facets to get summary information for different properties. - This method may use API query credits depending on usage.

```
var_dump($client->ShodanHostSearch(array(
    'query' => 'Niagara Web Server',
)));
```

Shodan Host Search Tokens (/tests/search.php):

This method lets you determine which filters are being used by the query string and what parameters were provided to the filters.

```
var_dump($client->ShodanHostSearchTokens(array(
    'query' => 'Niagara Web Server country:"IT"',
)));
```

Shodan Ports (/tests/ports.php):

This method returns a list of port numbers that the crawlers are looking for.

```
var_dump($client->ShodanPorts());
```

Shodan Protocols (/tests/protocols.php):

This method returns an object containing all the protocols that can be used when launching an Internet scan.

```
var_dump($client->ShodanProtocols());
```

Shodan Scan (/tests/crawl.php):

Use this method to request Shodan to crawl a network. - POST METHOD REQUIRE PAID API KEY.

```
var_dump($client->ShodanScan(array(
    'ips' => '69.171.230.0/24',
)));
```

Shodan Scan Internet (/tests/crawl.php):

Use this method to request Shodan to crawl the Internet for a specific port. - POST METHOD REQUIRE PAID API KEY AND SHODAN PERMISSION.

```
var_dump($client->ShodanScanInternet(array(
    'port' => '80',
    'protocol' => 'dns-tcp',
)));
```

Shodan Scan Id (/tests/crawl.php):

Check the progress of a previously submitted scan request.

```
var_dump($client->ShodanScan_Id(array(
    'id' => 'R2XRT5HH6X67PFAB',
)));
```

Composer v1 support is coming to an end ✖

Shodan Services (/tests/crawl.php):

This method returns an object containing all the services that the Shodan crawlers look at. It can also be used as a quick and practical way to resolve a port number to the name of a service.

```
var_dump($client->ShodanServices());
```

Shodan Query (/tests/saved_query.php):

Use this method to obtain a list of search queries that users have saved in Shodan.

```
var_dump($client->ShodanQuery(array(
    'page' => '1',
)));
```

Shodan Query (/tests/saved_query.php):

Use this method to search the directory of search queries that users have saved in Shodan.

```
var_dump($client->ShodanQuery(array(
    'query' => 'fax',
)));
```

Shodan Query Tags (/tests/query_tags.php):

Use this method to obtain a list of popular tags for the saved search queries in Shodan.

```
var_dump($client->ShodanQueryTags(array(
    'size' => '30',
)));
```

Tests class - Experimental method

Labs Honeyscore (/tests/honeypot.php):

Calculates a honeypot probability score ranging from 0 (not a honeypot) to 1.0 (is a honeypot).

```
var_dump($client->LabsHoneyscore(array(
    'ip' => '54.231.184.227', // http://mushmush.org/
)));
```

Tests class - Exploits REST API

Search Exploits (/tests/exploits.php):

Search across a variety of data sources for exploits and use facets to get summary information.

```
var_dump($client->Search(array(
    'query' => 'cve',
)));
```

Count Exploits (/tests/exploits.php):

This method behaves identical to the "/search" method with the difference that it doesn't return any results.

```
var_dump($client->Count(array(
    'query' => 'cve',
)));
```

Account Informations

This method allows you to get your Shodan's account information results.

```
var_dump($client->AccountProfile());

{
    "member": true,
    "credits": 0,
    "display_name": null,
    "created": "2020-06-15T10:44:43.148000"
}
```

DNS Resolve

Look up the IP address for the provided list of hostnames, takes a comma-separated list of hostnames.

```
var_dump($client->DnsResolve(array(
    'hostnames' => 'google.com,bing.com',
)));

{
    "google.com": "172.217.6.46",
    "facebook.com": "157.240.22.35"
}
```

DNS Reverse

Look up the hostnames that have been defined for the given list of comma-separated IP addresses.

```
var_dump($client->DnsReverse(array(
    'ips' => '8.8.8.8,1.1.1.1',
)));

{
    "8.8.8.8": [
        "dns.google"
    ],
    "1.1.1.1": [
        "one.one.one.one"
    ]
}
```


Composer v1 support is coming to an end ✖


[About Packagist](#)
[Atom/RSS Feeds](#)

[Statistics](#)
[Browse Packages](#)

[API](#)
[Mirrors](#)

[Status](#)
[Dashboard](#)

Maintenance and hosting provided by 

Bandwidth and CDN provided by  [.net](#)

Monitoring provided by  **DATADOG**

