



CLOUD CLIENT PHP

Support

The [ScientiaMobile Enterprise Support Portal](#) is open to all WURFL users, both commercial license holders and evaluation users. It represents the combined knowledge base for the WURFL community. Commercial licensees are invited to post questions in the forum using the account to which their licenses are associated. This may mean faster handling of those posts by ScientiaMobile's personnel.

For commercial license holders, there are tiered support levels to address a variety of business support needs. After logging into your account, commercial licensees with support options can access the [Enterprise Support](#) portal to post tickets. These tickets will receive expedited attention.

To inquire about support plans, use our [License Inquiry](#) or our [General Inquiry form](#).

Update Notifications

If you would like to be notified of our API updates, major data updates, and other technical changes, please [subscribe](#) to our ScientiaMobile Announcements list

scientiamobile

www.scientiamobile.com
Tel +1.703.310.6650
E-mail: sales@scientiamobile.com

Copyright © 2025 ScientiaMobile, all rights reserved. WURFL Cloud, WURFL OnSite, WURFL and, InFuze WURFL InSight and respective logos are trademarks of ScientiaMobile. Apache is the trademark of the Apache Software Foundation. NGINX is the trademark of Nginx Software Inc. Varnish is the trademark of Varnish Software AB

ScientiaMobile WURFL Cloud Client for PHP

Introduction

The WURFL Cloud Service by ScientiaMobile, Inc., is a cloud-based mobile device detection service that can quickly and accurately detect over 500 capabilities of visiting devices. It can differentiate between portable mobile devices, desktop devices, SmartTVs and any other types of devices that have a web browser.

In order to use the WURFL Cloud client you need to obtain an API key from Scientiamobile. The WURFL Cloud API is offered in two main flavors - **Base** and **Premium**. The Base version is limited to returning only a small number of capabilities; the Premium version doesn't have an upper limit to the number of returned capabilities and also offers more caching options for improving the overall performance. Once you've properly set up an account, you can download the cloud client and receive an API key to successfully use the library. You can create an account by visiting ScientiaMobile.com.

Upgrading from v1.x

*IMPORTANT: Since version 2.0.0, the WURFL Cloud Client for PHP uses namespaces. In addition, things were moved around a bit to allow for easier configuration. With the introduction of namespaces, **PHP 5.3+ is required**.*

To upgrade your existing v1 code, you will need to replace calls to the following classes:

Old:

```
require_once '../Client/Client.php';
```

```
WurflCloud_Client_Config  
WurflCloud_Client_Client  
WurflCloud_Cache_Null  
WurflCloud_Cache_Cookie
```

New:

```
require_once '../src/autoload.php';
```

```
ScientiaMobile\WurflCloud\Config  
ScientiaMobile\WurflCloud\Client  
ScientiaMobile\WurflCloud\Cache\Null  
ScientiaMobile\WurflCloud\Cache\Cookie
```

Every other class was also renamed/namespaced, so if you are using them directly, you can follow the mapping above to figure out the new name. If you were using the MyWurfl.php singleton, you can use the new one and no code changes will be required in your application.

Note that in v2, support for proxy servers were added, as well as better support for timeouts. More on these options is found below.

Installation

Requirements

- PHP 5.3+.
- JSON extension (almost always included).
- cURL extension is recommended.

Copy Files

If you're reading this, you've probably already extracted the Cloud Client files. You should put them

somewhere on your server that is accessible from your web application.

Test the WURFL Cloud Client

From your web browser, you should go to the WURFL Cloud Client's `examples/` folder. You will see the Compatibility Test Script, which will verify that your configuration is compatible with the WURFL Cloud Client.

You should test your API Key from this page by pasting it in the input box, then clicking `Test API Key`. If successful, you will see `Your server is able to access WURFL Cloud and your API Key was accepted`. If there was a problem, the error message will be displayed instead. Please note that it may take a few minutes from the time that you sign up for your WURFL Cloud API Key to become active.

Integration

You should review the included examples (`example.php`, `MyWurfl.php`, `show_capabilities.php`) to get a feel for the Client API, and how best to use it in your application.

Here is a quick example of how to get up and running quickly:

```
// Include the autoloader - edit this path!
require_once '../src/autoload.php';

// Create a configuration object
$config = new ScientiaMobile\WurflCloud\Config();

// Set your WURFL Cloud API Key
$config->api_key = 'xxxxxx:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx';

// Create the WURFL Cloud Client
$client = new ScientiaMobile\WurflCloud\Client($config);

// Detect your device
$client->detectDevice();

// Use the capabilities
if ($client->getDeviceCapability('is_wireless_device')) {
    echo "This is a mobile device";
} else {
    echo "This is a desktop device";
}
```

Configuration

The WURFL Cloud Client object `ScientiaMobile\WurflCloud\Client` takes three arguments: `Config`, `Cache`, `HttpClient`:

Config

`ScientiaMobile\WurflCloud\Config`: Used for setting the WURFL Cloud API Key `api_key` and adding `addCloudServer()` / removing `clearServers()` WURFL Cloud Servers.

Cache (optional)

Standard caching classes:

- `ScientiaMobile\WurflCloud\Cache\Null`: Disables caching completely
- `ScientiaMobile\WurflCloud\Cache\Cookie`: Cookie-based caching.

Premium WURFL Cloud accounts have access to additional, high-performance caching layers. Note

that these files are only included in the WURFL Cloud Client package, downloaded from a Premium account. Please refer to the included examples for specific cache adapter configuration.

- ScientiaMobile\WurflCloud\Cache\File: Filesystem-based caching.
- ScientiaMobile\WurflCloud\Cache\APC: APC memory-based caching.
- ScientiaMobile\WurflCloud\Cache\Memcache: Memcached distributed memory-based. caching using the PHP memcache extension.
- ScientiaMobile\WurflCloud\Cache\Memcached: Memcached distributed memory-based caching using the PHP memcached extension.

HttpClient (optional)

- ScientiaMobile\WurflCloud\HttpClient\Fsock: Uses native PHP fsock calls.
- ScientiaMobile\WurflCloud\HttpClient\Curl: Uses the PHP extension curl.
- ScientiaMobile\WurflCloud\HttpClient\Guzzle: Uses the [Guzzle HTTP client](#).

Note: to use Guzzle, you must have the Guzzle library loaded. To load it locally, you can run composer update from the root of the WURFL Cloud Client folder.

Then make sure to use the composer autoloader vendor/autoload.php instead of the built in one (src/sutoload.php).

Proxy Server Configuration

The Curl and Guzzle HTTP Clients support a proxy server configuration via the `setProxy()` method:

```
// Common proxy examples
$http_client = new ScientiaMobile\WurflCloud\HttpClient\Curl();

// Socks 4 Proxy
$http_client->setProxy("socks4://192.168.1.1:1080");

// Socks 4a Proxy
$http_client->setProxy("socks4a://192.168.1.1:1080");

// Socks 5 Proxy
$http_client->setProxy("socks5://192.168.1.1:1080");

// Socks 5 Proxy + Authentication
$http_client->setProxy("socks5://someuser:somepass@192.168.1.1:1080");

// HTTP Proxy
$http_client->setProxy("http://192.168.1.1:8080");

// HTTP Proxy + Authentication
$http_client->setProxy("http://someuser:somepass@192.168.1.1:8080");

// Pass $http_client in to the Client to use it
$client = new ScientiaMobile\WurflCloud\Client($config, $cache, $http_client);
```

HTTP Timeout

The WURFL Cloud Client is set to forcefully terminate the WURFL Cloud request after 1 second if a response has not been received. In some cases you may want to increase this timeout to account for high latency. All of the HTTP Clients support the `setTimeout()` method:

```
$http_client = new ScientiaMobile\WurflCloud\HttpClient\Curl();

// Timeout is in milliseconds (10000 == 10 seconds)
$http_client->setTimeout(10000);

// Pass $http_client in to the Client to use it
$client = new ScientiaMobile\WurflCloud\Client($config, $cache, $http_client);
```

Unit Testing

Unit tests are included with the client and can be run with PHPUnit. Before you can run the unit tests, you must install the dependencies via Composer from the root directory:

```
cd WurflCloudClient-PHP*
curl -sS https://getcomposer.org/installer | php
php composer.phar install
```

Before you run the PHPUnit tests, you must set your WURFL Cloud API Key in an environment variable so it can be used in the tests. To run the tests, run phpunit from the root directory (where phpunit.xml.dist is located):

```
export WURFL_CLOUD_API_KEY="123456:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
php vendor/phpunit/phpunit/phpunit.php -v
```

Note that in order to get all the tests to pass, you will need to use the API Key from a Premium WURFL Cloud account with all capabilities enabled. This is because small responses from the WURFL Cloud system are never compressed if they fit within one network packet, and the unit tests that cover compression see this as a failure. If this is the case, you will see failures like this:

There were 3 failures:

- 1) ScientiaMobile\WurflCloud\HttpClient\CurlTest::testCallCompression
Failed asserting that an array contains 'Content-Encoding: gzip'.
- 2) ScientiaMobile\WurflCloud\HttpClient\FsockTest::testCallCompression
Failed asserting that an array contains 'Content-Encoding: gzip'.
- 3) ScientiaMobile\WurflCloud\HttpClient\GuzzleTest::testCallCompression
Failed asserting that an array contains 'Content-Encoding: gzip'.

You will also need to have the extensions apc, memcache, memcached, json, and curl enabled. By default, apc is not enabled in CLI mode, so you may need to launch phpunit like this to force it on:

```
php -d apc.enable_cli=1 vendor/phpunit/phpunit/phpunit.php -v
```

© 2025 ScientiaMobile Inc.

All Rights Reserved.

NOTICE: All information contained herein is, and remains the property of ScientiaMobile Incorporated and its suppliers, if any. The intellectual and technical concepts contained herein are proprietary to ScientiaMobile Incorporated and its suppliers and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from ScientiaMobile Incorporated.