

# Guide to API Authentication - Username and Password Authentication

## Guide to API Authentication

[Access Hash Authentication](#)

[API Tokens in WHM](#)

[Browser-Based Authentication](#)

[Secure Remote Logins](#)

[Single Sign On](#)

[Two-Factor Authentication](#)

[Username and Password Authentication](#)

## Introduction

Scripts can authenticate via a username and password in an HTTP header. The script sends an HTTP header to the server during API functions. This allows the script to effectively log in as the desired user before the function.

### Important:

- We recommend that you use a secure remote login when possible. For more information, read our [Secure Remote Logins](#) documentation.
- **Only** use this method with a secure SSL connection over port 2083 (cPanel), port 2096 (Webmail), or port 2087 (WHM). Do **not** use this method to authenticate over an unsecured connection (port 2086, 2095, or 2082).
- API calls that use a method that includes a URL **must** use the correct port:
  - 2082 — Unsecure calls to cPanel's APIs.
  - 2083 — Secure calls to cPanel's APIs.
  - 2086 — Unsecure calls to WHM's APIs, or to cPanel's APIs via the WHM API.
  - 2087 — Secure calls to WHM's APIs, or to cPanel's APIs via the WHM API.
  - 2095 — Unsecure calls to cPanel's APIs via a Webmail session.
  - 2096 — Secure calls to cPanel's APIs via a Webmail session.

Otherwise-correct calls will return `Permission denied` or `Function not found` errors if they use an incorrect port number.

- This document **only** includes cPanel & WHM authentication methods. For Manage2 authentication information, read our [Guide to the Manage2 API](#) documentation.

## Username and password authentication

When you use the username and password method to authenticate, your script sends an HTTP header to the server during API function calls. This allows the script to effectively log in as the desired user before the function.

## Example Perl script

[Click to view...](#)

### Notes:

- This script runs as the `root` user.
- This script requires the `LWP::Protocol::https` module. If you attempt to run this script, you **must** first run the `/scripts/perlinstaller LWP::Protocol::https` command to install the module.
- This script calls WHM API 1's `listaccts` function. Make **cert**ain that you update this code for the correct API version, port, and other function-specific call information.

## Related documentation

- [Manage2 API Functions - Register Key-Based Authentication](#) — This function registers a server to use keyed authentication.

```
#!/usr/bin/perl
use strict;
use LWP::UserAgent;
use LWP::Protocol::https;
use MIME::Base64;

my $user = "root";
my $pass = "12345luggage";

my $auth = "Basic " .
MIME::Base64::encode( $user . ":" .
$pass );

my $ua = LWP::UserAgent->new(
    ssl_opts => { verify_hostname
=> 0, SSL_verify_mode =>
'SSL_VERIFY_NONE', SSL_use_cert => 0
},
);
my $request = HTTP::Request->new( GET
=>
"https://127.0.0.1:2087/json-api/list
accts?api.version=1" );
$request->header( Authorization =>
$auth );
my $response =
$ua->request($request);
print $response->content;
```

- Line 6 declares the `$user` variable and assigns it a value of `root`.
- Line 7 declares the `$pass` variable and assigns it the `root` account's password, `12345luggage`.
- Line 9 declares the `$auth` variable, and assigns it a value of `Basic root:12345luggage`.
- Line 12 declares the `$request` variable, which stores information about the call. To set its value, the `HTTP::Request` module's `new()` method creates a function to the WHM API 1 `listaccts` function.
  - This call uses the GET method.
  - When you construct URLs to use this method, use the same methods as for a browser-based call.
- Line 13 uses the `header()` method to use the `$auth` value as the call's authentication information.
- Line 14 uses the `LWP::UserAgent` module to run the function.
- Line 15 prints the function's output.

## Example PHP script

▼ [Click to view...](#)

### Note:

This script calls WHM API 1's `listaccts` function. Make **certain** that you update this code for the correct API version, port, and other function-specific call information.

- Guide to External Authentication - OpenID Connect - External authentication allows your server's users to log in to WHM, cPanel, and Webmail through OpenID Connect-compliant identity providers.

```

<?
$whmusername = "root";
$whmpassword = "12345luggage";

$query =
"https://127.0.0.1:2087/json-api/list
accts?api.version=1";

$curl = curl_init();
// Create Curl Object
curl_setopt($curl,
CURLOPT_SSL_VERIFYPEER,0); //
Allow self-signed certs
curl_setopt($curl,
CURLOPT_SSL_VERIFYHOST,0); //
Allow certs that do not match the
hostname
curl_setopt($curl, CURLOPT_HEADER,0);
// Do not include header in output
curl_setopt($curl,
CURLOPT_RETURNTRANSFER,1); //
Return contents of transfer on
curl_exec
$header[0] = "Authorization: Basic "
.
base64_encode($whmusername.":".$whmpa
ssword) . "\n\r";
curl_setopt($curl,
CURLOPT_HTTPHEADER, $header); //
set the username and password
curl_setopt($curl, CURLOPT_URL,
$query); // execute the query
$result = curl_exec($curl);
if ($result == false) {
    error_log("curl_exec threw error \"
    . curl_error($curl) . "\" for
    $query");

// log error if curl exec fails
}
curl_close($curl);

print $result;

?>

```

- Line 3 sets the `$whmusername` value as the `root` user.
- Line 4 sets the `$whmpassword` value as the `root` account's password, `12345luggage`.
- Line 6 assigns a WHM API 1 `listaccts` function to the `$query` value
  - When you construct URLs to use this method, use the same

- [Guide to External Authentication](#) — External Authentication modules allow users to log in through OpenID Connect-compliant identity providers.
- [Guide to Testing Custom Code - API Authentication](#) — This guide explains the basics of how to test your custom code's authentication with the cPanel & WHM server.

methods as for a browser-based call.

- Line 13 assigns the `$header[0]` variable a value of `Authorization: Basic $whmusername: $whmpassword`.
  - The `$whmusername` variable contains the account's username.
  - The `$whmpassword` variable contains the account's password.
- Line 14 uses the `$header` hash to properly configure the HTTP header for the function.
- Line 15 uses the `$query` variable to pass in the function itself.
- Lines 17 through 22 execute the function.
- Line 24 prints the function's output.

- [Guide to API Authentication - Browser-based Authentication](#)  
—  
Browser-based authentication accesses cPanel & WHM's functionality via the browser.