

5 PHP cURL examples

Published September 06, 2015

Working on the server side does not necessarily imply that all the required information needs to be present on the database of the site on which we are working. As a matter of fact, growing parts of the information that is used on the server side comes from external sources, often via an API that allows some kind of service to provide information to the website without having access to the database on the remote site. To utilize this information, we can use the **cURL** built-in PHP extension.

cURL is a PHP extension, that allows us to receive and send information via the URL syntax. By doing so, cURL makes it easy to communicate between different websites and domains. This tutorial includes 5 common cases for the use of cURL, and they include:

1. [Downloading the content of a website](#)
2. [Downloading a file from a website](#)
3. [Auto form submission](#)
4. [Authentication](#)
5. [Use of cookies](#)



Joseph Benharosh is a full stack web developer and the author of the eBook The essentials of object oriented PHP.

How does the cURL extension work?

cURL works by sending a request to a web site, and this process includes the following four parts:

1. Initialization.

```
$handle = curl_init();
```

2. Setting the options. There are many options, for example, an option that defines the URL.

```
curl_setopt($handle, CURLOPT_URL, $url);
```

3. Execution with `curl_exec()`.

```
$data = curl_exec($handle);
```

4. Releasing the cURL handle.

```
curl_close($handle);
```

The second part is the most interesting because it allows us to define how cURL works in a highly accurate manner, by using the many options it has to offer.

1. How to download the contents of a remote website to a local file?

In order to download the contents of a remote web site, we need to define the following options:

CURLOPT_URL- Defines the remote URL.

CURLOPT_RETURNTRANSFER- Enables the assignment of the data that we download from the remote site to a variable. In this example, we assign the data into the variable `$output`.

```
<?php
$handle = curl_init();

$url = "http://www.ladygaga.com";

// Set the url
curl_setopt($handle, CURLOPT_URL, $url);
// Set the result output to be a string.
curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);

$output = curl_exec($handle);

curl_close($handle);

echo $output;
```

When we print the value of the variable `$output` to the screen, we will see the local version of the web site for the world's best singer.



The options can be written more compactly using `curl_setopt_array()`, which is a cURL function that convenes the options into an array.

```
curl_setopt_array($handle,
    array(
        CURLOPT_URL           => $url,
        CURLOPT_RETURNTRANSFER => true
    )
);
```

2. How to download a file from a remote site using cURL?

A remote file can be downloaded to our server, if the option `CURLOPT_FILE` is set. For example, the following code downloads the book "The Divine Comedy" from Project Gutenberg into a `the_divine_comedy.html` file on our server:

```
<?php
// The distant site url.
$url = "http://www.gutenberg.org/files/46852/46852-h/46852-h.htm";
// The file on our server.
$file = __DIR__ . DIRECTORY_SEPARATOR . "the_divine_comedy.html";
$handle = curl_init();

// Open the file on our server for writing.
$fileHandle = fopen($file, "w");

curl_setopt_array($handle,
    array(
        CURLOPT_URL           => $url,
        CURLOPT_FILE          => $fileHandle,
    )
);

$data = curl_exec($handle);

curl_close($handle);

fclose($fileHandle);
```

Handling the returned response

In order to get the parameters of the response for it to be monitored and debugged, we need to set the option `CURLOPT_HEADER`. For example:

```
<?php
$url = "http://www.gutenberg.org/files/41537/41537-h/41537-h.htm";

$file = __DIR__ . DIRECTORY_SEPARATOR . "the_divine_comedy.html";


$handle = curl_init();

$fileHandle = fopen($file, "w");

curl_setopt_array($handle,
    array(
        CURLOPT_URL => $url,
        CURLOPT_FILE => $fileHandle,
        CURLOPT_HEADER => true
    )
);

$data = curl_exec($handle);
```

To get additional information about the request, we use the `curl_getinfo` command that enables us to receive important technical information about the response, including the status code (200 for success) and the size of the downloaded file.



```
$statusCode =
    curl_getinfo($handle,
        CURLINFO_HTTP_CODE
    );

$downloadLength =
    curl_getinfo($handle,
        CURLINFO_CONTENT_LENGTH_DOWNLOAD
    );
```

In addition, we can also use the commands: `curl_error` and `curl_errno` to debug the response and receive informative error messages.

```
if(curl_errno($handle))
{
    print curl_error($handle);
}
```

Let's see the full code:

```
<?php
$url = "http://www.gutenberg.org/files/46852/46852-h/46852-h.htm";

$file = __DIR__ . DIRECTORY_SEPARATOR . "the_divine_comedy.html";

$handle = curl_init();
```

```
$fileHandle = fopen($file, "w");

curl_setopt_array($handle,
    array(
        CURLOPT_URL    => $url,
        CURLOPT_FILE    => $fileHandle,
        CURLOPT_HEADER => true
    )
);

$data = curl_exec($handle);

$responseCode = curl_getinfo($handle, CURLINFO_HTTP_CODE);

$downloadLength = curl_getinfo($handle, CURLINFO_CONTENT_LENGTH_DOWNLOAD);

if(curl_errno($handle))
{
    print curl_error($handle);
}
else
{
    if($responseCode == "200") echo "successful request";

    echo " # download length : " . $downloadLength;

    curl_close($handle);

    fclose($fileHandle);
}
```

3. How to submit forms with cURL?

Until this moment, we have demonstrated the use of the GET method of HTTP (which is generally used to watch and download content). cURL can also make use of the POST method of HTTP in order to submit forms.

In order to demonstrate form submission with cURL, we need to create the following two files:

1. [index.php](#) in which we put the cURL script.
2. [form.php](#) in which we put the form to be submitted.

The [form.php](#) will be in reality, found on a remote server (although, for the sake of the example, both files may be placed on the same server). Also, for the example, we will use a form with 3 fields: firstName, lastName and submit.

```
<?php
if(isset($_POST["submit"]))
{
    echo "Full name is " . $_POST["firstName"] .
        " " . $_POST["lastName"];
    exit;
}
?>

<html>
```

```
<body>

<form method = "POST" action = "" >
  <input name="firstName" type="text">
  <input name="lastName" type="text">
  <input type="submit" name="submit" value="שלח" >
</form>
</body>
</html>
```

To submit the form, the following options need to be set:

1. **CURLOPT_POST** – Sets the request to be in a post mode.
2. **CURLOPT_POSTFIELDS** – Receives the associative array of the fields that we want to post. The array keys are named after the name of the form fields.

```
<?php
$handle = curl_init();

$url = "http://localhost/curl/theForm.php";

// Array with the fields names and values.
// The field names should match the field names in the form.

$postData = array(
    'firstName' => 'Lady',
    'lastName'  => 'Gaga',
    'submit'    => 'ok'
);

curl_setopt_array($handle,
    array(
        CURLOPT_URL => $url,
        // Enable the post response.
        CURLOPT_POST      => true,
        // The data to transfer with the response.
        CURLOPT_POSTFIELDS => $postData,
        CURLOPT_RETURNTRANSFER => true,
    )
);

$data = curl_exec($handle);

curl_close($handle);

echo $data;
```

4. How to perform basic HTTP authentication with cURL?

In order to authenticate with cURL, the following 3 options need to be set:

1. **CURLOPT_HTTPAUTH**
2. **CURLOPT_USERPWD** – Through which we define the username and password.
3. **CURLOPT_RETURNTRANSFER**

Let's see the code:

```
curl_setopt_array($handle,
    array(
        CURLOPT_URL => $url,
        CURLOPT_HTTPAUTH => CURLAUTH_ANY,
        CURLOPT_USERPWD => "$username:$password",
        CURLOPT_RETURNTRANSFER => true,
    )
);
```

5. How to handle cookies with cURL?

The use of cookies allows a website to identify returning visitors and authenticated users. To this end, cURL provides us with a mechanism through which we can save cookies.

The two main options that allow us to handle cookies are:

1. **CURLOPT_COOKIEJAR**– Defines the file required to write the cookies.
2. **CURLOPT_COOKIEFILE**– Defines the file from which the cookies are to be read.

The following code example writes the cookies into a `cookie.txt` file on the first visit, and then reads the data in later visits.

```
<?php

$handle = curl_init();

$url = "http://www.ladygaga.com/artrave-the-artpop-ball";

$file = __DIR__ . DIRECTORY_SEPARATOR . "cookie.txt";

curl_setopt_array($handle,
    array(
        CURLOPT_URL => $url,
        // The file to which the cookies need to be written.
        CURLOPT_COOKIEFILE => $file,
        // The file from which the cookies need to be read.
        CURLOPT_COOKIEJAR => $file,
        CURLOPT_RETURNTRANSFER => true,
    )
);

$data = curl_exec($handle);

curl_close($handle);
```

Conclusion

Using PHP's cURL extension provides us with a convenient way to communicate with other web sites, particularly with APIs that are provided by a third party. In the next tutorial, we will learn how to request for private details in the name of users that sign in to our website with their GitHub account. It will be done by using Github's API, and with the help of cURL. The tutorial will be a good starting point for learning how to make a social login with any social network.

Was this tutorial helpful?

If so, the eBook

"The essentials of object oriented PHP"

can further help you.

"The essentials..." is an in-depth guide to help you quickly and easily increase your productivity and become a hot commodity in the job market.

Don't waste time!

Click on the green button to buy the eBook and start achieving your potential today!

BUY THE EBOOK!

0 Comments [phpenthusiast.com](#)

 Login ▾

 Recommend  Share

Sort by Best ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

Be the first to comment.

ALSO ON PHPENTHUSIAST.COM

how to use composer and packagist | PHPenthusiast

16 comments • 3 years ago

titi123 — thnx a lot

The factory pattern in PHP | PHPenthusiast

4 comments • 3 years ago

Atanas — Really explanatory. I suggest this tutorial to everybody who wants to learn about how to use the factory pattern.

Inheritance in object-oriented PHP




18 comments • 3 years ago

phpenthusiast — Thank you Norbert Barto, I have sent you a copy of my ebook to your email.

Tips for improving PHP code | PHPenthusiast

2 comments • 3 years ago

Happy coder — Of all the code smells I know of the one that I like the most is that of a function with too many arguments. My favorite solution is to pass the third argument as an array, like in

 Subscribe  Add Disqus to your siteAdd DisqusAdd  Privacy

Share Us



Only the latest tutorial-No spam

SUBSCRIBE

© 2015-2017 [Phpenthusiast.com](http://phpenthusiast.com) | All rights reserved.

Website Designed by [Austin Pen & Pixel](#)