

The PHP Client URL (cURL) library

Fulvio Corno

e-lite Research Group
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino - Italy
<http://elite.polito.it>



v. 1.0, 2009-03-23

Outline

1 The cURL library



The problem

- From a PHP page, being able to retrieve information served by another web server
- The information must be *processed* by the PHP page, not simply shown on the resulting page
- We need an **synchronous remote call** mechanism



Ingredients

- 1 The **URL** address of a remote web page
- 2 The **parameters** that we should pass to such remote page
 - How many parameters? Names? Data types? Values?
 - In what form? GET? POST? Encoding?
- 3 The **format of the results** returned by the remote page
 - Format: text? xml?
 - DTD or schema?
 - Error codes: in http response header or encoded as part of the result?



The cURL library

- `cURL` is one of the most powerful PHP extensions. It stands for Client URL, and allows you to communicate with other servers using a wide range of protocols.
- `libcurl` (the library behind the PHP cURL extension) currently supports a wide range of protocols, including HTTP, HTTPS, FTP, TELNET, FILE, LDAP, DICT, GOPHER and HTTPS, as well as HTTPS certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, and user:password authentication.



Main functions

- `curl_init` Initialize a cURL session – returns a “handle” object (`$ch`)
- `curl_setopt` Set an option for a cURL transfer – common options: URL, POST data, result disposition
- `curl_exec` Perform a cURL session – actual data transfer and http request
- `curl_close` Close a cURL session – end of transaction



Basic example

```
$ch = curl_init( ) ;

curl_setopt($ch, CURLOPT_URL, "http://abc.com/page.php") ;

// do a POST
curl_setopt($ch, CURLOPT_POST, true) ;
curl_setopt($ch, CURLOPT_POSTFIELDS, "id=333") ;

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true) ;
// return the result of curl_exec, instead
// of outputting it directly

$result = curl_exec($ch) ;

curl_close($ch) ;
```



Sending the request

- set the URL with `CURLOPT_URL`
- parameters in GET
 - encoded in the URL
 - "...page.php?id=333&user=pippo"
- parameters in POST
 - specify POST instead of GET with option `CURLOPT_POST` set to **true**
 - specify all parameters in `CURLOPT_POSTFIELDS`
 - encoded as a string
 - as an associative array in PHP



Getting the result

- call `curl_exec($ch)`
- usually, the result is directly *interpolated* in the page (implicit `echo`)
- to avoid interpolation, set `CURLOPT_RETURNTRANSFER` to `true`
 - result is returned as a string by the `curl_exec` call



Other functions

- `curl_copy_handle` Copy a cURL handle along with all of its preferences
- `curl_errno` Return the last error number
- `curl_error` Return a string containing the last error for the current session
- `curl_getinfo` Get information regarding a specific transfer
- `curl_setopt_array` Set multiple options for a cURL transfer
- `curl_version` Gets cURL version information



Other options

CURLOPT_HEADER TRUE to include the header in the output

CURLOPT_PORT An alternative port number to connect to.

CURLOPT_HTTPHEADER An array of HTTP header fields to set.

CURLOPT_FILE The file that the transfer should be written to. The default is STDOUT (the browser window).



Further information

- <http://www.php.net/manual/en/book.curl.php>



License

This document is licensed under the Creative Commons
Attribution-Noncommercial-Share Alike 3.0 Unported License.

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

