



FREE eBook

LEARNING

curl

Free unaffiliated eBook created from
Stack Overflow contributors.

#curl

Table of Contents

About	1
Chapter 1: Getting started with curl	2
Remarks.....	2
Examples.....	2
Transfer data using curl.....	2
Using curl in PHP to fetch data.....	2
Using curl through the command line.....	2
Use the libcurl easy C API to get a remote resource.....	3
Chapter 2: Curl Installation	4
Examples.....	4
From packages.....	4
Chapter 3: Name resolve curl tricks	5
Examples.....	5
Editing the hosts file.....	5
Providing custom IP address for a name.....	5
Change the "Host:" header.....	5
Credits	7

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [curl](#)

It is an unofficial and free curl ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official curl.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with curl

Remarks

This section provides an overview of what curl is, and why a developer might want to use it.

It should also mention any large subjects within curl, and link out to the related topics. Since the Documentation for curl is new, you may need to create initial versions of those related topics.

Examples

Transfer data using curl

cURL is the name of the project which depicts: 'Client for URLs' and also be called as **Client URL Request Library**

it combines two separate packages: curl and libcurl.

1. `curl` is a command line tool used to get documents/files from or send documents to a server, using any of the supported protocols: DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMTP, SMTPS, Telnet and TFTP.
2. `libcurl` is the underlying library curl uses to do the actual networking and transfer work. `libcurl` is used by thousands of services, applications and devices and very often through one of the "language bindings" that allows programmers of higher level languages to access its powers.

Using curl in PHP to fetch data

```
<?php

    $ch = curl_init(); //curl handler init

    curl_setopt($ch,CURLOPT_URL,"http://www.google.com/search?q=curl");
    curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);// set optional params
    curl_setopt($ch,CURLOPT_HEADER, false);

    $result=curl_exec($ch);

    curl_close($ch);

    echo $result;

?>
```

Using curl through the command line

Show curl version:

```
curl --version
```

GET a remote resource and have it displayed in the terminal:

```
curl http://stackoverflow.com
```

GET a remote resource and save it in a local file:

```
curl -o file https://stackoverflow.com
```

Add headers to response:

```
curl -i http://stackoverflow.com
```

Output only headers:

```
curl -I http://stackoverflow.com
```

Use the libcurl easy C API to get a remote resource

```
#include <stdio.h>
#include <curl/curl.h>

int main(void)
{
    CURL *curl;
    CURLcode res;

    curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");

        /* example.com is redirected, so we tell libcurl to follow redirection */
        curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);

        /* Perform the request, res will get the return code */
        res = curl_easy_perform(curl);
        /* Check for errors */
        if(res != CURLE_OK)
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(res));

        /* always cleanup */
        curl_easy_cleanup(curl);
    }
    return 0;
}
```

Read Getting started with curl online: <https://riptutorial.com/curl/topic/4246/getting-started-with-curl>

Chapter 2: Curl Installation

Examples

From packages

The curl source packages can be downloaded from the following page:

<https://curl.haxx.se/download.html>

However, the best way to install it is to use your package repository.

For Linux distros, you can use apt-get, yum or rpm depending on the distribution you are using, so the exact command will be:

```
apt-get install curl
```

Or:

```
yum install curl
```

For Mac users, you can install curl via Homebrew. More details here:

<http://brewformulas.org/Curl>

Read Curl Installation online: <https://riptutorial.com/curl/topic/10591/curl-installation>

Chapter 3: Name resolve curl tricks

Examples

Editing the hosts file

The easiest way to connect via curl to a different server is to alter the hosts file on your machine.

On Linux and Unix systems, the hosts file is located in **/etc/hosts**, while on Windows systems it will be located in **c:\windows\system32\drivers\etc\hosts**.

Once you open the file with a text editor of your choice, add

```
1.2.3.4 domain.tld www.domain.tld
```

This is basically the IP of the server you would like to resolve the domain to followed by the domain and a www version of the domain.

Curl will then resolve to this domain until the added line in the hosts file is removed.

The limitation in this example is that editing the hosts file often requires admin access and also, it affects all applications connected to the domain at the same time.

Providing custom IP address for a name

The most effective way to resolve curl to a different server is to use the `--resolve` option. This option will insert the address into curl's DNS cache, so it will effectively make curl believe that's the address it got when it resolved the name. Like so:

```
curl --resolve example.com:80:1.2.3.4 http://example.com/
```

In the above example, firstly we specify the domain (example.com), then we ask it to connect on port 80 to the IP 1.2.3.4. Depending on the protocol used and the server's configuration, the port can vary. For HTTP the port is 80 and for HTTPS, the port is 443.

It is important to note here that the `--resolve` option will send SNI for the name in the URL. This means that when connecting to the server via HTTPS, curl will verify the server's response to make sure it servers for the name in the URL. In other words, it will ensure there is an SSL on the server installed for the domain.

Change the "Host:" header

The "Host:" header is a normal way an HTTP client tells the HTTP server which server it speaks to. By passing custom modified "Host:" header you can have the server respond with the content of the site, even if you didn't actually connect to the host name.

For example, if you have a site on your localhost and you wish to have curl ask for its index page, the command is:

```
curl -H "Host: example.com" http://localhost/
```

The main disadvantage of modifying the "Host:" header is that curl will only extract the SNI name to send from the given URL. In other words, the "Host:" header modification is not enough when communication with a server via HTTPS.

Read Name resolve curl tricks online: <https://riptutorial.com/curl/topic/10565/name-resolve-curl-tricks>

Credits

S. No	Chapters	Contributors
1	Getting started with curl	Alexander Prokofyev , Altaf Hussain , Community , Daniel Stenberg , Danilo Terra , Riad
2	Curl Installation	Zdravko B
3	Name resolve curl tricks	Zdravko B