

Curl Command Cheat Sheet



Syntax

```
$ curl [parameters] [URL]
```

Display the command usage and lists most common options

```
$ curl --help
```

Display the command usage and list all available options

```
$ curl --help all
```

Basic Operations

COMMAND	DESCRIPTION
<pre>\$ curl http://example.com</pre>	Fetch a URL
<pre>\$ curl -O http://example.com/file.zip</pre>	Download a file
<pre>\$ curl -L http://example.com</pre>	Follow redirections

Data Transfer

COMMAND	DESCRIPTION
<pre>\$ curl -d "key1=value1&key2=value2" http://example.com/post_endpoint</pre>	Post Data
<pre>\$ curl -d '{"key1":"value1", "key2":"value2"}' -H "Content-Type: application/json" http://example.com/api</pre>	POST JSON Data
<pre>\$ curl -F "file=@path_to_file" http://example.com/upload</pre>	Upload a file

Authentication & Headers

COMMAND	DESCRIPTION
<pre>\$ curl -u username:password</pre>	HTTP Basic Authentication
<pre>\$ curl -H "Authorization: Bearer YOUR_TOKEN" http://example.com</pre>	Add Headers

Other Useful Options

COMMAND	DESCRIPTION
<pre>\$ curl --limit-rate 1M -O http://example.com/file.zip</pre>	Limit Rate (e.g., 1MB/s)
<pre>\$ curl -C - -O http://example.com/file.zip</pre>	Resume Broken Download
<pre>\$ curl -x http://proxyserver:port http://example.com</pre>	Use a Proxy

Debugging & Info

COMMAND	DESCRIPTION
<pre>\$ curl -v http://example.com</pre>	Verbose Mode
<pre>\$ curl -I http://example.com</pre>	Retrieve Only Headers
<pre>\$ curl --version</pre>	Curl Version & Protocols

SSL (Secure Socket Layer)

COMMAND	DESCRIPTION
<pre>\$ curl -k https://example.com</pre>	Skip SSL Certificate Verification
<pre>\$ curl --cert mycert.pem https://example.com</pre>	Use SSL Certificate

Common Options

Option	DESCRIPTION
<code>-d, --data <data></code>	HTTP POST data
<code>-f, --fail</code>	Fail fast with no output on HTTP errors
<code>-h, --help <category></code>	Get help for commands
<code>-I, --include</code>	Include protocol response headers in the output
<code>--output <file></code>	Write to file instead of stdout
<code>-O, --remote-name</code>	Write output to a file named as the remote file
<code>--silent</code>	Silent mode
<code>-T, --upload-file <file></code>	Transfer local FILE to destination
<code><user:password></code>	Server user and password
<code>-A, --user-agent <name></code>	Send User-Agent <name> to server

