

Curl Cheatsheet

By Dejan Panovski • Updated on Jan 12, 2026 • [Download PDF](#)

Quick reference for curl command-line HTTP client

Curl is a versatile command-line tool for transferring data using various protocols including HTTP, HTTPS, FTP, and more. This cheatsheet covers common use cases from simple downloads to complex API interactions with authentication and custom headers.

Basic Requests

Make simple HTTP requests.

curl url	GET request
curl -o file url	Save to file
curl -O url	Save with original name
curl -s url	Silent mode
curl -v url	Verbose output
curl -I url	Headers only (HEAD)
curl -L url	Follow redirects

HTTP Methods

Use different HTTP request methods.

curl -X GET url	GET request
curl -X POST url	POST request
curl -X PUT url	PUT request
curl -X DELETE url	DELETE request
curl -X PATCH url	PATCH request
curl -X OPTIONS url	OPTIONS request

POST Data

Send data with POST requests.

<code>curl -d "data" url</code>	POST form data
<code>curl -d @file url</code>	POST from file
<code>curl --data-urlencode "k=v" url</code>	URL encode data
<code>curl -F "file=@path" url</code>	Upload file
<code>curl -F "field=value" url</code>	Multipart form

JSON Data

Work with JSON APIs.

<code>curl --json '{"key":"value"}' url</code>	POST JSON
<code>curl --json @data.json url</code>	POST JSON from file
<code>echo '{}' curl --json @- url</code>	POST JSON from stdin
<code>curl -d '{"key":"value"}' -H "Content-Type: application/json" url</code>	POST JSON (older curl)
<code>curl -H "Accept: application/json" url</code>	Request JSON
<code>curl url jq</code>	Parse JSON response

Headers

Set custom HTTP headers.

<code>curl -H "Header: Value" url</code>	Add header
<code>curl -H "Authorization: Bearer token" url</code>	Auth header
<code>curl -H "Content-Type: text/xml" url</code>	Content type
<code>curl -A "User-Agent" url</code>	User agent
<code>curl -e "referer" url</code>	Referer header

Authentication

Handle various authentication methods.

<code>curl -u user:pass url</code>	Basic auth
<code>curl -u user url</code>	Prompt for password
<code>curl --digest -u user:pass url</code>	Digest auth
<code>curl --ntlm -u user:pass url</code>	NTLM auth
<code>curl -H "Authorization: Bearer token" url</code>	Bearer token
<code>curl --oauth2-bearer token url</code>	OAuth2 bearer

Cookies

Send and receive cookies.

<code>curl -b "name=value" url</code>	Send cookie
<code>curl -b cookies.txt url</code>	Send from file
<code>curl -c cookies.txt url</code>	Save cookies
<code>curl -b cookies.txt -c cookies.txt url</code>	Read and save

SSL/TLS

Handle HTTPS and certificates.

<code>curl -k url</code>	Ignore SSL errors
<code>curl --insecure url</code>	Ignore SSL errors
<code>curl --cacert ca.crt url</code>	Use CA cert
<code>curl --cert client.crt url</code>	Client cert
<code>curl --key client.key url</code>	Client key
<code>curl --cert client.pem:pass url</code>	Cert with password

Proxy

Route requests through proxies.

<code>curl -x proxy:port url</code>	HTTP proxy
<code>curl -x socks5://proxy:port url</code>	SOCKS5 proxy
<code>curl --proxy-user user:pass url</code>	Proxy auth
<code>curl --noproxy "*.local" url</code>	Bypass proxy

Download

Download files and resume transfers.

<code>curl -o file url</code>	Save to file
<code>curl -O url</code>	Use remote filename
<code>curl -O url1 -O url2</code>	Multiple files
<code>curl -C - -O url</code>	Resume download
<code>curl --limit-rate 1M url</code>	Limit speed
<code>curl -# url</code>	Progress bar

Upload

Upload files to servers.

<code>curl -T file url</code>	Upload file (PUT)
<code>curl -F "file=@path" url</code>	Form upload
<code>curl -F "file=@path;type=image/png" url</code>	With content type
<code>curl --upload-file file url</code>	Upload file

FTP

Work with FTP servers.

<code>curl ftp://server/file</code>	Download file
<code>curl -T file ftp://server/</code>	Upload file
<code>curl -u user:pass ftp://server/</code>	FTP with auth
<code>curl --ftp-ssl ftp://server/</code>	FTP over SSL
<code>curl ftp://server/ --list-only</code>	List directory

Timeouts

Control connection timing.

<code>curl --connect-timeout 10 url</code>	Connect timeout
<code>curl -m 30 url</code>	Max time
<code>curl --max-time 30 url</code>	Max time
<code>curl --retry 3 url</code>	Retry on failure
<code>curl --retry-delay 5 url</code>	Delay between retries

Output Control

Format and filter output.

<code>curl -s url</code>	Silent (no progress)
<code>curl -S url</code>	Show errors in silent
<code>curl -w "%{http_code}" url</code>	Print HTTP code
<code>curl -w "%{time_total}" url</code>	Print total time
<code>curl -o /dev/null -s -w "%{http_code}" url</code>	Status code only

Debug

Troubleshoot requests.

<code>curl -v url</code>	Verbose
<code>curl -vv url</code>	More verbose
<code>curl --trace file url</code>	Full trace to file
<code>curl --trace-ascii file url</code>	ASCII trace
<code>curl -D headers.txt url</code>	Dump headers

Common Patterns

Frequently used combinations.

<code>curl -sL url</code>	Silent, follow redirects
<code>curl -fsSL url</code>	Fail silently on errors
<code>curl --json '{"key":"value"}' url</code>	JSON POST
<code>curl -u user:pass -XPUT -d @file url</code>	Auth PUT from file