

curl Quick Reference

HTTP requests, headers, auth, forms, debugging

Basic Usage

Simple Requests

```
curl https://example.com # GET request
curl -o file.html https://url # save to file
curl -O https://url/file.tar.gz # save with remote name
curl -L https://url # follow redirects
```

Common Flags

-s	Silent mode (no progress)
-S	Show errors in silent mode
-f	Fail silently on HTTP errors
-L	Follow redirects
-o file	Write output to file
-O	Save with remote filename
-C -	Resume interrupted download
--max-time 30	Timeout after 30 seconds

HTTP Methods

GET & HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (headers only)
curl -i https://example.com # include response headers
```

POST

```
curl -X POST https://api.example.com/users \
-H "Content-Type: application/json" \
-d '{"name":"Jo","email":"jo@ex.com"}
```

PUT & PATCH & DELETE

```
curl -X PUT https://api.example.com/users/1 \
-d '{"name":"Updated"}'
curl -X PATCH https://api.example.com/users/1 \
-d '{"email":"new@ex.com"}'
curl -X DELETE https://api.example.com/users/1
```

Headers

Setting Headers

```
curl -H "Content-Type: application/json" URL
curl -H "Accept: text/html" URL
curl -H "X-Custom: value" URL
curl -H "Header1: v1" -H "Header2: v2" URL
```

Response Headers

-i	Include response headers in output
-I	Fetch headers only (HEAD)
-D file	Dump response headers to file
-w '%{http_code}'	Print HTTP status code

Authentication

Basic & Token Auth

```
curl -u user:pass https://api.example.com
curl -H "Authorization: Bearer TOKEN" URL
curl -u user:pass --digest URL
curl --negotiate -u : URL # Kerberos/SPNEGO
```

Auth Methods

-u user:pass	Basic authentication
--digest	HTTP Digest auth
--negotiate	Kerberos/SPNEGO auth
--ntlm	NTLM authentication
-n	Use ~/.netrc credentials

Data & Forms

Sending Data

```
curl -d "key=val&key2=val2" URL # form urlencoded
curl -d @data.json URL # data from file
curl --data-raw '{"raw":"json"}' URL
curl --data-urlencode "q=hello world" URL
```

File Uploads

```
curl -F "file=@photo.jpg" URL
curl -F "file=@doc.pdf;type=application/pdf" URL
curl -F "field=value" -F "file=@img.png" URL
```

Multipart vs URL-Encoded

-d	application/x-www-form-urlencoded
-F	multipart/form-data
--json	Shorthand: sets Content-Type + Accept to JSON
-T file	Upload file via PUT

SSL/TLS

Certificate Options

```
curl --cacert ca.pem URL # custom CA bundle
curl --cert client.pem URL # client certificate
curl --cert client.pem --key key.pem URL
curl -k URL # skip TLS verify (dev only)
```

TLS Flags

-k / --insecure	Skip TLS certificate verification
--cacert file	Use custom CA certificate
--cert file	Client certificate
--key file	Client private key
--tlsv1.2	Force minimum TLS 1.2
--tlsv1.3	Force minimum TLS 1.3

Output & Debugging

Verbose & Trace

```
curl -v URL # verbose output
curl --trace dump.txt URL # full trace to file
curl --trace-ascii - URL # trace to stdout
curl -w "\n%{http_code}\n" URL # custom output format
```

Write-Out Variables

%{http_code}	HTTP response status code
%{time_total}	Total time in seconds
%{time_connect}	Time to establish connection
%{size_download}	Downloaded bytes
%{speed_download}	Average download speed
%{redirect_url}	Redirect URL (if any)
%{ssl_verify_result}	SSL verification result (0 = OK)

Write-Out Example

```
curl -s -o /dev/null -w \
"code: %{http_code}\ntime: %{time_total}s\n" \
https://example.com
```

Common Patterns

API Workflow

```
# GET JSON and pipe to jq
curl -s https://api.example.com/data | jq '.items[]'
# POST JSON with auth
curl -s -H "Authorization: Bearer $TOKEN" \
--json '{"key":"val"}' https://api.example.com
```

Download Patterns

```
# Download with progress bar
curl -# -O https://releases.example.com/v2.tar.gz
# Resume interrupted download
curl -C - -O https://releases.example.com/v2.tar.gz
# Download multiple files
curl -O https://url/file1 -O https://url/file2
```

Scripting Helpers

```
# Check if URL is reachable
curl -sf -o /dev/null https://example.com && echo OK
# Save cookies and reuse
curl -c cookies.txt -b cookies.txt URL
# Rate-limit request
curl --limit-rate 100k URL
```

Proxy & Network

Proxy Settings

```
curl -x http://proxy:8080 URL
curl -x socks5://proxy:1080 URL
curl --proxy-user user:pass -x http://proxy:8080 URL
curl --noproxy "*.local,localhost" URL
```

DNS & Resolve

--resolve host:port:addr	Force DNS resolution to addr
--dns-servers 8.8.8.8	Use custom DNS server
--interface eth0	Use specific network interface
-4 / -6	Force IPv4 / IPv6

Config & Advanced

Config Files

```
# ~/.curlrc - default options
--silent
--location
--max-time 30
```

```
# Use config file explicitly
curl -K myconfig.txt URL
```

Useful Flags

--retry 3	Retry on transient errors
--retry-delay 2	Delay between retries (seconds)
--compressed	Request and decompress gzip/br
--limit-rate 100k	Throttle transfer speed
-Z	Parallel transfers (curl 7.66+)
--create-dirs	Create path directories for -o