curl Transfer data by URL

CONCEPTS

curl prints the response body to stdout and progress to stderr, so it composes with pipes. It does not fail on HTTP 404/500 unless you add -f.

URL scheme, host, path
↓ connect
Request method, headers, body
↓ send
Response status, headers, body
↓ output
Output stdout, -o, -O

Quote URLs containing & or ? so the shell does not split them.

REQUESTS

`curl <url>`
GET a URL

`curl -X <method> <url>`
Set the request method

`curl -d <data> <url>`
POST form-encoded data

`curl -d @<file> <url>`
POST data from a file

`curl --json '<json>' <url>`
POST JSON with proper headers

`curl -F <name>=<value> <url>`
Multipart form field

`curl -F <name>=@<file> <url>`
Upload a file as form field

`curl -G -d <k=v> <url>`
Send data as query string

`curl --url-query <k=v> <url>`
Add an encoded query param

`curl --data-urlencode <k=v> <url>`
URL-encode a POST value

`curl -T <file> <url>`
Upload with PUT

`curl -I <url>`
HEAD request, headers only

`curl --data-binary @<file> <url>`
POST file exactly as is

`curl -X PATCH -d <data> <url>`
Send a PATCH

HEADERS & AUTH

`curl -H "<name>: <value>" <url>`
Add a custom header

`curl -H "<name>:" <url>`
Remove a default header

`curl -u <user>:<pass> <url>`
Basic authentication

`curl -H "Authorization: Bearer <token>" <url>`
Send a bearer token

`curl --oauth2-bearer <token> <url>`
Bearer token shortcut

`curl -A "<agent>" <url>`
Set the User-Agent

`curl -b "<name>=<value>" <url>`
Send a cookie

`curl -b <jar> -c <jar> <url>`
Read and write a cookie file

`curl -n <url>`
Credentials from ~/.netrc

OUTPUT & DOWNLOAD

`curl -o <file> <url>`
Save body to a file

`curl -O <url>`
Save under the remote name

`curl -L <url>`
Follow redirects

`curl -i <url>`
Include response headers

`curl -s <url>`
Silent: no progress meter

`curl -sS <url>`
Silent, but show errors

`curl -f <url>`
Fail on HTTP errors (22)

`curl --fail-with-body <url>`
Fail on errors, keep body

`curl -C - -O <url>`
Resume a partial download

`curl -D <file> <url>`
Dump headers to a file

`curl -w "<format>" <url>`
Print info after transfer

`curl --compressed <url>`
Ask for and decode gzip/br

NETWORK & TLS

`curl -k <url>`
Skip certificate checks

`curl --cacert <file> <url>`
Trust a specific CA

`curl --cert <file> --key <key> <url>`
Client certificate

`curl -x <proxy> <url>`
Use a proxy

`curl --resolve <host>:<port>:<ip> <url>`
Pin a hostname to an IP

`curl --connect-timeout <s> <url>`
Limit connect time

`curl -m <s> <url>`
Limit total time

`curl --retry <n> <url>`
Retry transient failures

DEBUGGING

`curl -v <url>`
Verbose request and response

`curl --trace-ascii <file> <url>`
Full trace, readable

`curl -w "%{time_total}\n" -o /dev/null -s <url>`
Time a request

`curl -sI <url>`
Check headers quietly

`curl --help all`
List every option

`curl "<url>[1-5].txt"`
URL globbing ranges

RECIPES

POST JSON and check status

```
curl -sS --fail-with-body \
  --json '{"name":"x"}' \
  -H "Authorization: Bearer $TOKEN" \
  https://api.example.com/items
```

Robust download

```
curl -fL --retry 3 --connect-timeout 10 \
  -C - -o file.tgz https://example.com/file.tgz
```

Print only the HTTP status

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