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 -I <url>`
HEAD request, headers only

HEADERS & AUTH

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

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

HEADERS & AUTH (cont.)

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

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

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

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

NETWORK & TLS

`curl -k <url>`
Skip certificate checks

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

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

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

DEBUGGING

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