wget Download and mirror

START HERE

`wget <url>`
Download a file

`wget -q -O - <url>`
Print to stdout

`wget -O <file> <url>`
Choose the local name

`wget -r -np <url>`
Fetch a directory tree

`wget -c <url>`
Resume a partial download

`wget -m <url>`
Mirror a site

`wget -i <list.txt>`
Download many URLs

`wget --spider <url>`
Check without downloading

Fetch URL retries, redirects → saves → **Local file** -O, -P, -nd name it → -r parses HTML → **Follow links** -l depth, -np, -A/-R → -k rewrites → **Offline copy** -m -k -p

CONCEPTS

wget saves to a file by default and keeps going on its own: it retries, follows redirects and can recurse. curl prints to stdout and does not recurse; see the curl sheet for APIs.

An existing file is not overwritten: wget saves as name.1, name.2. Use -c to resume, -N to refresh only newer files, or -nc to skip.

DOWNLOAD

`wget <url>`
Download to the current directory

`wget -O <file> <url>`
Save under a chosen name
-O acts like shell redirection: it truncates the file. -N is ignored with -O; -c still resumes into it. -O - writes to stdout.

`wget -P <dir> <url>`
Save into a directory

`wget -c <url>`
Resume an interrupted download

`wget -i <file>`
Read URLs from a file

`wget -q <url>`
Quiet, no output

`wget -nv <url>`
Short one-line output

`wget -b <url>`
Run in the background
Progress goes to wget-log in the current directory.

`wget -O - <url>`
Write the body to stdout

`wget --show-progress -q <url>`
Quiet but keep the progress bar

`wget -nc <url>`
Skip if the file already exists

`wget -N <url>`
Only fetch if remote is newer

`wget --content-disposition <url>`
Use the server-suggested filename

`wget -E <url>`
Add .html when a page lacks it

`wget --start-pos=<offset> <url>`
Start at a byte offset

RELIABILITY AND SPEED

`wget -t <n> <url>`
Number of retries (0 is unlimited)

`wget --retry-connrefused <url>`
Retry on refused connections

`wget --retry-on-http-error=503,429 <url>`
Retry on listed HTTP codes

`wget -T <sec> <url>`
Set all timeouts

`wget --connect-timeout=<sec> <url>`
Limit connection setup

`wget --read-timeout=<sec> <url>`
Limit idle time while reading

`wget --waitretry=<sec> <url>`
Back off between retries

`wget --limit-rate=<rate> <url>`
Cap bandwidth (e.g. 500k, 2m)

`wget -w <sec> -i <file>`
Pause between downloads

`wget --random-wait -w <sec> -i <file>`
Vary the pause

`wget -Q <size> -i <file>`
Stop after a total size

`wget -4 <url>`
Use IPv4 only

RECURSIVE AND MIRROR

`wget -r <url>`
Follow links recursively (depth 5)

`wget -r -l <n> <url>`
Set recursion depth (0 or inf is unlimited)

`wget -np -r <url>`
Never ascend to the parent directory

`wget -m <url>`
Mirror: -r -N -l inf
Also adds --no-remove-listing, which only matters for FTP.

`wget -k <url>`
Convert links for offline viewing

`wget -p <url>`
Also fetch images, CSS, scripts

`wget -nd -r <url>`
Flatten into one directory

RECURSIVE AND MIRROR (cont.)

`wget -nH --cut-dirs=<n> -r <url>`
Drop host and N path levels

`wget -A <list> -r <url>`
Accept only these extensions

`wget -R <list> -r <url>`
Reject these extensions

`wget --accept-regex=<re> -r <url>`
Accept URLs matching a regex

`wget --reject-regex=<re> -r <url>`
Skip URLs matching a regex

`wget -H -D <domains> -r <url>`
Span hosts but only these domains

`wget -E -k -p <url>`
Save one page fully for offline reading

`wget -e robots=off -r <url>`
Ignore robots.txt
Use only where you are allowed to; be gentle with -w.

HTTP, AUTH AND HEADERS

`wget --header='<Name: value>' <url>`
Add a request header

`wget --user=<u> --ask-password <url>`
Basic auth, prompt for password
--password=<p> on the command line is visible in ps.

`wget --http-user=<u> --http-password=<p> <url>`
HTTP-only credentials

`wget --auth-no-challenge --user=<u> --ask-password <url>`
Send Basic auth without waiting for 401

`wget --load-cookies <file> <url>`
Send cookies from a file

`wget --save-cookies <file> --keep-session-cookies <url>`
Store cookies after a login

`wget -U '<agent>' <url>`
Set the User-Agent

`wget --referer=<url> <url>`
Set the Referer header

HTTP, AUTH AND HEADERS (cont.)

`wget --post-data='<k=v>' <url>`
Send a POST with form data

`wget --post-file=<file> <url>`
POST the contents of a file

`wget --method=PUT --body-file=<file> <url>`
Other methods with a body

`wget --content-on-error <url>`
Keep the body on 4xx/5xx

`wget -S <url>`
Show the server response headers

`wget --spider <url>`
Check a URL without saving it

`wget --no-proxy <url>`
Bypass configured proxies

TLS AND LOGGING

`wget --no-check-certificate <url>`
Skip certificate validation
Insecure; prefer --ca-certificate.

`wget --ca-certificate=<file> <url>`
Trust a custom CA bundle

`wget --https-only -r <url>`
Never follow plain-HTTP links

`wget -o <file> <url>`
Write the log to a file

wget vs curl

Need	wget	curl
Default output	`file`	`stdout`
Resume	`-c`	`-C - -O`
Recurse	`-r`	`no`
API calls	`--method`	`-X -d`

RECIPES

Resume with retries

```
wget -c -t 0 --retry-connrefused
-T 30 <url>
```

Polite site mirror

```
wget -m -k -p -E -np -w 1 --
random-wait \
  --limit-rate=500k <url>
```

Grab only PDFs from a page

```
wget -r -l 1 -nd -np -A pdf -P
pdfs <url>
```

Download list, quietly, to a log

```
wget -nv -c -i urls.txt -P
downloads -o fetch.log
```

Authenticated download

```
wget --header='Authorization:
Bearer <token>' \
  -O out.json <url>
```