xh Friendly HTTP client

CONCEPTS

> The separator in each argument picks its meaning: ':' header, '==' query, '=' string field, ':=' raw JSON value, '@' file upload. Data fields make it a POST, otherwise GET.

URL scheme optional
Items k:v k==v k=v
↓ parsed
Request JSON by default
↓ sent
Response colored on tty
↓ printed

REQUESTS

`xh <url>`
GET a URL and pretty-print

`xh post <url> <k>=<v>`
Explicit method

`xh put <url> <k>=<v>`
Replace a resource

`xh delete <url>`
Delete a resource

`xh <url> <k>=<v> <k2>=<v2>`
Build a JSON object

`xh <url> <k>:=<json>`
Send number, bool, array, object

`xh <url> <a>[<b>][0]=<v>`
Nested JSON path

`xh <url> <k>==<v>`
Add query parameter

`xh <url> <h>:<v>`
Set a header

`xh <url> <h>:`
Remove a default header

`xh <url> <h>:@<file>`
Read header value from file

`xh :3000/<path>`
Shorthand for localhost:3000

`xh --raw '<body>' <url>`
Send a raw body

`xh <url> < <file>`
Body from stdin

`xh <url> -I < <file>`
Ignore stdin (in scripts, cron)

FORMS AND FILES

`xh -f <url> <k>=<v>`
URL-encoded form body

`xh --multipart <url> <k>=<v>`
Multipart form body

`xh -f <url> <k>@<file>`
Upload a file field

`xh -f <url> <k>@<file>;type=<mime>`
Upload with MIME type

`xh -f <url> <k>@<file>;filename=<n>`
Upload with another filename

OUTPUT

`xh -v <url>`
Print request and response

`xh -h <url>`
Print response headers only

`xh -b <url>`
Print response body only

`xh -p <what> <url>`
Choose parts: H B h b m

`xh -q <url>`
Suppress output

`xh --pretty <mode> <url>`
all, colors, format or none

`xh -S <url>`
Stream the response as it arrives

`xh --curl <url>`
Print the equivalent curl command

`xh --offline <url> <k>=<v>`
Build and show request without sending

AUTH

`xh -a <user>:<pass> <url>`
Basic auth

`xh -a <user> <url>`
Basic auth, prompt for password

`xh -A bearer -a <token> <url>`
Bearer token

DOWNLOADS AND REDIRECTS

`xh -d <url>`
Download body to a file

DOWNLOADS AND REDIRECTS (cont.)

`xh -d -o <file> <url>`
Download to a chosen name

`xh -d -c -o <file> <url>`
Resume an interrupted download

`xh -F <url>`
Follow redirects

`xh --check-status <url>`
Non-zero exit on 4xx/5xx (default)

`xh --no-check-status <url>`
Exit 0 despite HTTP errors

SESSIONS

`xh --session <name> <url>`
Create or reuse a session

`xh --session <file> <url>`
Session stored in a path

`xh --session-read-only <name> <url>`
Use session, do not update it

> A session stores headers, cookies and auth for one host. Log in once with --session, then later calls reuse the cookies without repeating -a.

NETWORK AND TLS

`xh --timeout <sec> <url>`
Give up after N seconds

`xh --proxy <proto>:<url> <url>`
Route through a proxy

`xh --verify no <url>`
Skip certificate verification

`xh --verify <ca.pem> <url>`
Trust a custom CA bundle

`xh --https <host>`
Default to HTTPS

CONCEPTS AND SETUP

`xhs <url>`
HTTPS by default alias

`xh --help`
List all options

`xh --version`
Print version