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

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 --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

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

`xh -F <url>`
Follow redirects

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

SESSIONS

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

NETWORK AND TLS

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

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