HTTPie Friendly HTTP client

CONCEPTS

HTTPie sends JSON by default. The separator in each item decides its role: : header, == query, = string field, := raw JSON, @ file. The method is optional: GET without data, POST with.

Request item separators

Item	Becomes
`Name:value`	Header
`q==value`	Query parameter
`k=value`	String field
`k:=42`	Raw JSON (number, bool, list)
`k=@file`	Field from text file
`k@file`	File upload (with -f)

When stdout is redirected, colours and formatting turn off. In scripts and cron add --ignore-stdin so HTTPie does not wait on stdin.

REQUESTS

`http <url>`
GET a URL

`http POST <url> <key>=<value>`
POST a JSON object

`http PUT <url> <key>=<value>`
Use any other method

`https <url>`
Default to https://

`http :3000/path`
Request localhost:3000

`http <url> <key>:=<json>`
Raw JSON value

`http <url> < <file>`
Send a file as the body

HEADERS AND QUERY

`http <url> <Name>:<value>`
Set a header

`http <url> <key>==<value>`
Query parameter

`http --follow <url>`
Follow redirects

FORMS AND FILES

`http -f POST <url> <key>=<value>`
URL-encoded form

`http -f POST <url> <key>@<file>`
Upload a file (multipart)

OUTPUT

`http -v <url>`
Request and response

`http -h <url>`
Response headers only

`http -b <url>`
Response body only

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

`http --offline POST <url> <items>`
Print request, send nothing

`http --pretty none <url>`
No colours or formatting

AUTH

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

`http -a <user> <url>`
Prompt for password

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

`http --verify no <url>`
Skip TLS verification

SESSIONS

`http --session <name> -a <u>:<p> <url>`
Create or reuse a session

`http --session <name> <url>`
Replay stored auth/cookies

DOWNLOADS

`http -d <url>`
wget-style download

SCRIPTING AND NETWORK

`http --check-status <url>`
Non-zero exit on 3xx-5xx

`http --ignore-stdin <url>`
Do not read stdin