HTTPie Command-Line HTTP Client Cheatsheet

A comprehensive guide to using HTTPie, a command-line HTTP client, for making API requests, testing endpoints, and inspecting HTTP traffic with examples and usage scenarios.

Basic Usage and Parameters

Making Basic Requests

`http [METHOD] URL [item [item]]` - Basic syntax for making HTTP requests.

Example:

`http GET example.com` - Sends a GET request to example.com

`http POST example.com` - Sends a POST request.

Specifying HTTP Method:

HTTPie defaults to GET if no method is specified. Other methods like POST, PUT, DELETE, etc., can be explicitly used.

Examples:

`http PUT example.com/api/resource/1`

`http DELETE example.com/api/resource/1`

Including Headers:

Custom headers can be included directly in the command.

Example:

`http GET example.com 'Content-Type: application/json'`

Passing Request Data

`item` - Request data (query parameters, body). `field=value`	String parameters in the request body or query string. **Example:** `http POST example.com name='John Doe'` - Sends a POST request with `name=John Doe` in the body.
`field==value`	URL parameters. **Example:** `http GET example.com search=='my search'` - Sends a GET request to `example.com?search=my+search`.
`field:=value`	Non-string parameters. Useful for JSON payloads. **Example:** `http POST example.com age:=29` - Sends a POST request with `age: 29` as a JSON field.
`field:='[1,2,3]'`	JSON Data. Sends a JSON array. **Example:** `http POST example.com list:='[1,2,3]'` - sends `list` equals to `[1, 2, 3]`
`file@path`	Attach files. **Example:** `http POST example.com file@/path/to/file.txt` - Sends a POST request with the contents of `file.txt`.
`field=@file`	Read field value from a file (text). **Example:** `http POST example.com token=@token.txt` - Sets the `token` field to the content of `token.txt`.
`field:=@file.json`	Read field value from a file (JSON). **Example:** `http POST example.com user:=@user.json` - Sets the `user` field to the JSON content of `user.json`.

Working with Forms and Raw JSON

Submitting Forms

`--form` - To submit forms, use the `--form` option. This sets the Content-Type to `application/x-www-form-urlencoded`.

Example:

`http --form POST example.com name='John Smith' cv@document.txt`

Ensure correct Content-Type:

When submitting forms, HTTPie automatically sets the `Content-Type` header. Manually setting this header might be necessary in some cases.

Example:

`http --form POST example.com Content-Type:application/x-www-form-urlencoded name='value'`

Sending Raw JSON Data

Piping Raw JSON:

You can pipe raw JSON data to HTTPie for POST or PUT requests. This is useful when you have a JSON payload from another command or file.

Example:

`echo '{"hello": "world"}' | http POST example.com/post`

Specifying Content-Type for JSON:

When sending raw JSON data, ensure that the `Content-Type` header is set to `application/json`.

Example:

`echo '{"key": "value"}' | http POST example.com Content-Type:application/json`

Handling Different Content Types

HTTPie supports various content types, and you can specify them using the `Content-Type` header. This ensures that the server correctly interprets the data you send.

Example:

`http POST example.com Content-Type:application/xml < data.xml`

Options for Output and Authentication

Printing Options

`-v`, `--verbose`	Verbose mode, same as `--print=HhBb --all`. **Example:** `http -v example.com` - Shows full request and response details.
`-h`, `--headers`	Print only headers, same as `--print=h`. **Example:** `http -h example.com` - Displays only the response headers.
`-b`, `--body`	Print only body, same as `--print=b`. **Example:** `http -b example.com` - Shows only the response body.
`--all`	Print intermediate requests. **Example:** `http --all example.com` - Shows all requests in case of redirects.
`--print=FORMAT`	Specify which parts of the request/response to print. **Example:** `http --print=HhBb example.com` - Shows request headers, response headers, request body, and response body.
`--pretty=STYLE`	Style for output formatting (none, all, colors, format). **Example:** `http --pretty=colors example.com` - Formats output with colors.
`--json`, `-j`	Response is serialized as a JSON object. **Example:** `http -j example.com` - Ensures the response is treated as JSON.

Authentication Options

`--auth USER:PASS`	Provide HTTP Basic authentication credentials. **Example:** `http --auth user:password example.com`
`--auth-type TYPE`	Specify the authentication type (basic, digest). **Example:** `http --auth-type digest --auth user:password example.com`

Session Management and Downloading

Session Handling

`--session NAME`	Create or use a session to store authentication and cookies. **Example:** `http --session mysession example.com` - Creates a session named 'mysession'.
`--session-read-only NAME`	Use a read-only session. **Example:** `http --session-read-only mysession example.com` - Uses 'mysession' in read-only mode.
Session Use Cases	Sessions are useful for maintaining authentication and cookies across multiple requests. They help simulate a user's interaction with a website. **Example:** `http --session logintest POST example.com/login username=myuser password=mypass` `http --session logintest GET example.com/protected` - Accesses a protected resource using the stored session.

Downloading Content

`--download`, `-d`	Download the response body like wget. **Example:** `http -d example.com/file.zip` - Downloads `file.zip`.
`--continue`, `-c`	Continue an interrupted download. **Example:** `http -c -d example.com/largefile.iso` - Continues downloading `largefile.iso`.
`--output FILE`, `-o FILE`	Specify the output file for the downloaded content. **Example:** `http -d -o output.html example.com` - Downloads the content and saves it to `output.html`.

Advanced Options

Redirection and Timeouts

`--follow`, `-F`	Follow HTTP redirects. **Example:** `http -F example.com` - Follows any redirects from example.com.
`--max-redirects N`	Set the maximum number of redirects to follow. **Example:** `http --max-redirects 3 -F example.com` - Follows up to 3 redirects.
`--timeout SECONDS`	Set a timeout for the request. **Example:** `http --timeout 10 example.com` - Sets a 10-second timeout.

SSL Verification and Proxies

`--verify no`	Skip SSL verification. **Example:** `http --verify no example.com` - Skips SSL verification (not recommended for production).
`--proxy PROTO:URL`	Specify a proxy to use for the request. **Example:** `http --proxy http:http://proxy.example.com:8080 example.com` - Uses the specified HTTP proxy.

Request Headers Examples

`Content-Type`	Sets the media type of the body of the request. **Example:** `http POST example.com Content-Type:application/json data='{"key": "value"}'`
`Authorization`	Provides credentials to authenticate with a server. **Example:** `http GET example.com Authorization:'Bearer YOUR_API_TOKEN'`
`User-Agent`	Identifies the client making the request. **Example:** `http GET example.com User-Agent:'MyCustomApp/1.0'`