httpbin(1): HTTP Request & Response Service

Freely hosted in [HTTP](#), [HTTPS](#), & [EU](#) flavors by [Kenneth Reitz](#) & [Heroku](#).

ENDPOINTS

[/](#) This page.
[/ip](#) Returns Origin IP.
[/user-agent](#) Returns user-agent.
[/headers](#) Returns header dict.
[/get](#) Returns GET data.
/post Returns POST data.
/patch Returns PATCH data.
/put Returns PUT data.
/delete Returns DELETE data
[/encoding/utf8](#) Returns page containing UTF-8 data.
[/gzip](#) Returns gzip-encoded data.
[/deflate](#) Returns deflate-encoded data.
[/brotli](#) Returns brotli-encoded data.
[/status/:code](#) Returns given HTTP Status code.
[/response-headers?key=val](#) Returns given response headers.
[/redirect/:n](#) 302 Redirects *n* times.
[/redirect-to?url=foo](#) 302 Redirects to the *foo* URL.
[/redirect-to?url=foo&status_code=307](#) 307 Redirects to the *foo* URL.
[/relative-redirect/:n](#) 302 Relative redirects *n* times.
[/absolute-redirect/:n](#) 302 Absolute redirects *n* times.
[/cookies](#) Returns cookie data.
[/cookies/set?name=value](#) Sets one or more simple cookies.
[/cookies/delete?name](#) Deletes one or more simple cookies.
[/basic-auth/:user/:passwd](#) Challenges HTTPBasic Auth.
[/hidden-basic-auth/:user/:passwd](#) 404'd BasicAuth.
[/digest-auth/:qop/:user/:passwd/:algorithm](#) Challenges HTTP Digest Auth.
[/digest-auth/:qop/:user/:passwd](#) Challenges HTTP Digest Auth.
[/stream/:n](#) Streams *min(n, 100)* lines.
[/delay/:n](#) Delays responding for *min(n, 10)* seconds.
[/drip?numbytes=n&duration=s&delay=s&code=code](#) Drips data over a duration after an optional initial delay, then (optionally) returns with the given status code.
[/range/1024?duration=s&chunk_size=code](#) Streams *n* bytes, and allows specifying a *Range* header to select a subset of the data. Accepts a *chunk_size* and request *duration* parameter.
[/html](#) Renders an HTML Page.
[/robots.txt](#) Returns some robots.txt rules.
[/deny](#) Denied by robots.txt file.
[/cache](#) Returns 200 unless an If-Modified-Since or If-None-Match header is provided, when it returns a 304.
[/etag/:etag](#) Assumes the resource has the given etag and responds to If-None-Match header with a 200 or 304 and If-Match with a 200 or 412 as appropriate.
[/cache/:n](#) Sets a Cache-Control header for *n* seconds.
[/bytes/:n](#) Generates *n* random bytes of binary data, accepts optional *seed* integer parameter.
[/stream-bytes/:n](#) Streams *n* random bytes of binary data, accepts optional *seed* and *chunk_size* integer parameters.
[/links/:n](#) Returns page containing *n* HTML links.
[/image](#) Returns page containing an image based on sent Accept header.
[/image/png](#) Returns a PNG image.
[/image/jpeg](#) Returns a JPEG image.
[/image/webp](#) Returns a WEBP image.

[/image/svg](#) Returns a SVG image.
[/forms/post](#) HTML form that submits to */post*
[/xml](#) Returns some XML

DESCRIPTION

Testing an HTTP Library can become difficult sometimes. [RequestBin](#) is fantastic for testing POST requests, but doesn't let you control the response. This exists to cover all kinds of HTTP scenarios. Additional endpoints are being considered.

All endpoint responses are JSON-encoded.

EXAMPLES

$ curl http://httpbin.org/ip

```
{"origin": "24.127.96.129"}
```

$ curl http://httpbin.org/user-agent

```
{"user-agent": "curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8l zlib/1.2.3"}
```

$ curl http://httpbin.org/get

```
{
    "args": {},
    "headers": {
        "Accept": "*/*",
        "Connection": "close",
        "Content-Length": "",
        "Content-Type": "",
        "Host": "httpbin.org",
        "User-Agent": "curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8l zlib/1.2.3"
    },
    "origin": "24.127.96.129",
    "url": "http://httpbin.org/get"
}
```

$ curl -I http://httpbin.org/status/418

```
HTTP/1.1 418 I'M A TEAPOT
Server: nginx/0.7.67
Date: Mon, 13 Jun 2011 04:25:38 GMT
Connection: close
x-more-info: http://tools.ietf.org/html/rfc2324
Content-Length: 135
```

$ curl https://httpbin.org/get?show_env=1

```
{
    "headers": {
        "Content-Length": "",
        "Accept-Language": "en-US,en;q=0.8",
        "Accept-Encoding": "gzip,deflate,sdch",
        "X-Forwarded-Port": "443",
        "X-Forwarded-For": "109.60.101.240",
        "Host": "httpbin.org",
        "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
        "User-Agent": "Mozilla/5.0 (X11; Linux i686) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.83 Safari/535.11"
        "X-Request-Start": "1350053933441",
        "Accept-Charset": "ISO-8859-1,utf-8;q=0.7,*;q=0.3",
        "Connection": "keep-alive",
        "X-Forwarded-Proto": "https",
        "Cookie": "_gauges_unique_day=1; _gauges_unique_month=1; _gauges_unique_year=1; _gauges_unique=1; _gauges_unique_hour
        "Content-Type": ""
    },
    "args": {
        "show_env": "1"
    },
    "origin": "109.60.101.240",
    "url": "http://httpbin.org/get?show_env=1"
}
```

Installing and running from PyPI

You can install httpbin as a library from PyPI and run
it as a WSGI app. For example, using Gunicorn:

```
$ pip install httpbin
$ gunicorn httpbin:app
```

AUTHOR

A Kenneth Reitz project.

SEE ALSO

Hurl.it - Make HTTP requests.

RequestBin - Inspect HTTP requests.

http://python-requests.org