Install HTTPie

Please make sure you have Python 3.7 or newer (`python --version`).

```
python3 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
pip install httpie
```

`python3 -m venv .venv` :
This command creates a new virtual environment in the current directory. The virtual environment is named `.venv` . Virtual environments are a way to keep the dependencies required by different projects separate.

`source .venv/bin/activate` :
This command activates the virtual environment you just created. When a virtual environment is activated, the Python interpreter and installed packages associated with the virtual environment are used instead of the system-wide ones.

`pip install --upgrade pip` :
This command upgrades `pip` to their latest versions within the activated virtual environment. `pip` is the package installer for Python.

`pip install httpie` :
This command installs the `httpie` package within the activated virtual environment.

Usage

```
http -v https://reqres.in/api/users?page=2
```

The command `http -v https://reqres.in/api/users?page=2` is a command-line HTTP client request made using HTTPie, a user-friendly command-line HTTP client for the API era.

Here's what each part of the command does:

- `http` : This is the HTTPie command itself. It's used to make an HTTP request.
- `-v` : This is a command option that stands for "verbose". When you use this option, HTTPie will display the entire HTTP request (including headers and body data) and the entire HTTP response (including headers and body data).

- `https://reqres.in/api/users?page=2` : This is the URL that you're sending the HTTP request to. In this case, it's an API endpoint that returns user data.

Request

```
GET /api/users?page=2 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: reqres.in
User-Agent: HTTPie/3.2.2
```

This is an HTTP GET request. Here's what each part of the request does:

`GET /api/users?page=2 HTTP/1.1`
This line is the request line, which is made up of the HTTP method (`GET`), the request-target (`/api/users?page=2`), and the HTTP version (`HTTP/1.1`). The `GET` method is used to request data from a specified resource. In this case, it's requesting data from the `/api/users?page=2` endpoint.

`Accept: */*`
The `Accept` header field can be used by user agents to specify response media types that are acceptable. `*/*` means that any media type is acceptable.

`Accept-Encoding: gzip, deflate`
The `Accept-Encoding` header field can be used by user agents to indicate what response content-codings are acceptable. In this case, it's saying that it can accept responses that are encoded with `gzip` and `deflate` .

`Connection: keep-alive`
The `Connection` header field allows the sender to specify options that are desired for that particular connection. In this case, it's specifying that the network connection should stay open after the current transaction finishes for potential subsequent requests.

`Host: reqres.in`
The `Host` request header specifies the domain name of the server and port number (if applicable) to which the request is being sent. It's required in all HTTP/1.1 request messages.

`User-Agent: HTTPie/3.2.2`
The `User-Agent` header field contains information about the user agent originating the request. This information is often used by servers for analytics, debugging, logging, or showing customized content. In this case, it's indicating that the user agent software is HTTPie version 3.2.2.

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Cache-Control: max-age=14400
Connection: keep-alive
Content-Encoding: gzip
Content-Type: application/json; charset=utf-8
Date: Sun, 29 Oct 2023 08:40:07 GMT
Etag: W/"406-ut0vzoCuidvyMf8arZpMpJ6ZRDw"
X-Powered-By: Express

{
    "data": [
        {
            "avatar": "https://reqres.in/img/faces/7-image.jpg",
            "email": "michael.lawson@reqres.in",
            "first_name": "Michael",
            "id": 7,
            "last_name": "Lawson"
        },
        {
            "avatar": "https://reqres.in/img/faces/8-image.jpg",
            "email": "lindsay.ferguson@reqres.in",
            "first_name": "Lindsay",
            "id": 8,
            "last_name": "Ferguson"
        },
        {
            "avatar": "https://reqres.in/img/faces/9-image.jpg",
            "email": "tobias.funke@reqres.in",
            "first_name": "Tobias",
            "id": 9,
            "last_name": "Funke"
        },
        {
            "avatar": "https://reqres.in/img/faces/10-image.jpg",
            "email": "byron.fields@reqres.in",
            "first_name": "Byron",
            "id": 10,
            "last_name": "Fields"
        },
        {
            "avatar": "https://reqres.in/img/faces/11-image.jpg",
            "email": "george.edwards@reqres.in",
            "first_name": "George",
            "id": 11,
            "last_name": "Edwards"
        },
        {
            "avatar": "https://reqres.in/img/faces/12-image.jpg",
            "email": "rachel.howell@reqres.in",
            "first_name": "Rachel",
            "id": 12,
            "last_name": "Howell"
        }
    ],
    "page": 2,
    "per_page": 6,
    "total": 12,
    "total_pages": 2
}
```

`HTTP/1.1 200 OK:`
This is the status line of the response. `HTTP/1.1` is the HTTP version used by the server, `200` is the HTTP status code indicating success, and `OK` is the reason phrase associated with the 200 status code.

`Access-Control-Allow-Origin:*`
This header indicates which origins are allowed to read the response. The `*` means that any origin may read the response. A response that tells the browser to allow requesting code from the origin https://developer.mozilla.org to access a resource will include the following:

```
Access-Control-Allow-Origin: https://developer.mozilla.org
```

`Cache-Control: max-age=14400`
This header defines caching policies. `max-age=14400` means that the resource is valid for 14400 seconds (4 hours) from when it was generated.

`Connection: keep-alive`
This header indicates whether the network connection should remain open after the current transaction finishes.

`Content-Encoding: gzip`
This header specifies that gzip encoding was used to compress the body of the response.

`Content-Type: application/json; charset=utf-8`
This header indicates that the media type of the resource is JSON and its character encoding is UTF-8.

`Date: Sun, 29 Oct 2023 08:40:07 GMT`
The `Date` header contains the date and time at which the message originated.

`Etag: W/"406-ut0vzoCuidvyMf8arZpMpJ6ZRDw"`
The term ‘W/’ (case-sensitive) indicates the use of a weak validator. Weak ETags are easy to generate, but they are less useful for comparisons. On the other hand, strong validators are ideal for comparisons, but they can be challenging to generate efficiently. It’s important to note that the weak ETag values of two representations of the same resources might be semantically equivalent, but not byte-for-byte identical. This means that weak ETags prevent caching when byte range requests are used. However, strong ETags allow range requests to be cached.

An entity tag, or ETag, uniquely represents the requested resource. It is a string of ASCII characters enclosed in double quotes, like ‘675af34563dc-tr34’. The method for generating ETag values is not specified. Typically, an ETag value could be a hash of the content, a hash of the last modification timestamp, or simply a revision number. For instance, a wiki engine might use a hexadecimal hash of the article content as the ETag.

`X-Powered-By: Express`
This header indicates what technology powers the server.

The body of your HTTP response contains JSON data, which includes user information, page details, and support text.

Other Examples

GET SINGLE USER

```
http -v https://reqres.in/api/users/2
```

GET SINGLE USER NOT FOUND

```
http -v https://reqres.in/api/users/23
```

POST CREATE

```
http -v POST https://reqres.in/api/users name="Alireza Aghamohammadi" job="Software Engineer"
```

PUT Replace

```
http -v PUT https://jsonplaceholder.typicode.com/posts/1 title="SAD"
```

PATCH Modify

```
http -v PATCH https://jsonplaceholder.typicode.com/posts/1 title="SAD"
```

DELETE REMOVE

```
http -v DELETE https://reqres.in/api/users/2
```

POST SUCCESSFUL REGISTER

```
http -v POST https://reqres.in/api/register email='eve.holt@reqres.in' password='pistol'
```

POST UNSUCCESSFUL REGISTER

```
http -v POST https://reqres.in/api/register email='al.aghamohammadi@gmail.com'
```