CURL COMMANDS

1. To find the location of the curl program in your system "where curl" is used.

2. To find the version of curl "curl --version" is used.

3. GET -> To send a get request to a URL we just write curl followed by the url, it generates the file html file of the given url.

4. -o -> This saves the output from the get request to the url.

5. -I -> This gives the header of the given url.

6. -O -> This saves the output from the get request to the url and save with the given name instead of using a custom name.

7. --limit-rate -10k -> This will limit the download rate.

8. -C - -> It is used to continue the download of the file.

```
C:\Users\user>curl -O https://mirrors.nxtgen.com/ubuntu-mirror/ubuntu-releases/22.04.3/ubuntu-22.04.3-desktop-amd64.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0 4804M    0 1279k    0     0    312k      0  4:22:47  0:00:04  4:22:43  312k^C
C:\Users\user>curl --limit-rate 100K -O https://mirrors.nxtgen.com/ubuntu-mirror/ubuntu-releases/22.04.3/ubuntu-22.04.3-desktop-amd64.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0 4804M    0 2927k    0     0    100k      0 13:36:45  0:00:29 13:36:16 62761^C
C:\Users\user>curl -C - -O https://mirrors.nxtgen.com/ubuntu-mirror/ubuntu-releases/22.04.3/ubuntu-22.04.3-desktop-amd64.iso
** Resuming transfer from byte position 3325693
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0 4801M    0  367k    0     0   53278      0 26:14:51  0:00:07 26:14:44 56899^C
C:\Users\user>
```

9. -L -> By default redirects are not followed in curl so if we want to follow the redirects we use this.

10. -a -> This is used to append the output of the file to an existing one.

11. -v -> Sometimes referred as verbose mode this allows to view request headers and connection information of the url. Request data, response headers, and other information about the request, such as the IP used and the SSL handshake process, are all included in the output.

```
C:\Users\user>curl -v https://www.google.com/
*   Trying [2404:6800:4009:825::2004]:443...
* Connected to www.google.com (2404:6800:4009:825::2004) port 443 (#0)
* schannel: disabled automatic use of client certificate
* ALPN: offers http/1.1
* ALPN: server accepted http/1.1
* using HTTP/1.1
> GET / HTTP/1.1
> Host: www.google.com
> User-Agent: curl/8.0.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Thu, 31 Aug 2023 06:44:08 GMT
< Expires: -1
< Cache-Control: private, max-age=0
< Content-Type: text/html; charset=ISO-8859-1
< Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-9TvV-C7i0Fl4YMZ1ktqj8g' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsaf
e-inline' https: http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp
< P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
< Server: gws
< X-XSS-Protection: 0
< X-Frame-Options: SAMEORIGIN
< Set-Cookie: 1P_JAR=2023-08-31-06; expires=Sat, 30-Sep-2023 06:44:08 GMT; path=/; domain=.google.com; Secure
< Set-Cookie: AEC=Ad49MVHmT9BuGa3V0kApQedrZGqyZqH9zdqANba00a1We7hdoTD-yS_cXgA; expires=Tue, 27-Feb-2024 06:44:08 GMT; path=/; domain=.google.com; Secure; HttpOnly; Same
Site=lax
< Set-Cookie: NID=511=psG7uyfI3TInXC8ZW_3ytEAPnQXaoHJhKknlyXHy5KwbZiR1qCU9isR4ztyYxoiX1C5P1GXSBA_Wx-LlvauWVpLUGcJfq4TjnziUJFpRhZSlhcdaj5wREAdJxIT-tuXGFa3fZzNF1ZIUMTowP1
68gU5SWyWGr1WKJpruEgDXK-w; expires=Fri, 01-Mar-2024 06:44:08 GMT; path=/; domain=.google.com; HttpOnly
< Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
< Accept-Ranges: none
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
<
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta conte
nt="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="9TvV-C7i0Fl4YMZ1ktqj8g">(function(){var _g={kEI:'
uDbwZPChIryn5OUP6teUmAQ',kEXPI:'0,1359409,6058,207,4804,2316,383,246,5,1129120,1197789,612,380090,16114,19397,9287,22431,1361,12319,17580,4998,17075,38444,887,1985,2891
,8349,3405,606,29843,30847,2614,13721,1014,1,16916,2652,4,3832,29062,13065,13658,27032,6642,7596,1,42154,2,16737,23024,5679,1020,31122,4568,6256,23421,1252,25135,8,7921
,2,2,1,10957,15675,8155,7381,2,15968,873,19635,5,1922,9779,22893,15785,3781,20198,20137,14,82,13332,6874,8377,3787,4264,10937,5375,2266,764,3034,3076,9706,1804,7734,562
,991,2,8487,10260,2172,5249,8199,7948,892,4652,949,545,3147,5227,3339,1991,5777,146,5769,1884,1743,3346,3,9,4129,1271,976,1437,1179,5210139,2,196,46,197,5994622,280311
,3311,141,795,19735,1,1,346,8263,111,31,32,10,23942037,579,4043528,16672,39283,1621,641,1123,1397272,23759270,887,2436,1852,2,2777,1,208,10336,630,2079,2878,45,2639,24
65,1708,1478,3245,241,1967,2949,941,1050,1374,1689,14,283,2389,671,68,1596,1,476,267,638,1160,382,2111,264,2,3,3226,482,1253,1087,3,284,3251,111,3108,891,354,9,191,482,
899,4,375,61,2,2,338,24,40,5,16,3,1423,393,493,2,498,101,194,49,1494,744,220,47,1506,155,3979,894,185,699,881,229,511,2,680,277,241,835,297,380,881,712,475,3,249,225,17
```

12. -vo -> When we don't want the body and just save the output of the headers and to a null device.

13. -H -> Used to setup the custom header for a url.

14. --data -> It is used to send a POST request to the server and send the data.

15. -X –H ->Sending JSON data to the server using POST.

```
Command Prompt

C:\Users\user>curl -X POST -H "Content-Type: application/json" -d "{"Name": "Ansu"}" https://httpbin.org/post
{
  "args": {},
  "data": "{Name: Ansu}",
  "files": {},
  "form": {},
  "headers": {
    "Accept": "*/*",
    "Content-Length": "12",
    "Content-Type": "application/json",
    "Host": "httpbin.org",
    "User-Agent": "curl/8.0.1",
    "X-Amzn-Trace-Id": "Root=1-64f03e60-5db0d9ce47e2a94d74f548b7"
  },
  "json": null,
  "origin": "122.163.56.204",
  "url": "https://httpbin.org/post"
}

C:\Users\user>
```

16. -sSo ->Manually verifying the Http requests.

```
file1 - Notepad

File  Edit  Format  View  Help
<!DOCTYPE html>
<html lang=en>
  <meta charset=utf-8>
  <meta name=viewport content="initial-scale=1, minimum-scale=1, width=device-width">
  <title>Error 404 (Not Found)!!1</title>
  <style>
    *{margin:0;padding:0}html,code{font:15px/22px arial,sans-serif}html{background:#fff;color:#222;padding:15px}body{margin
r_150x54dp.png) no-repeat;-webkit-background-size:100% 100%}}#logo{display:inline-block;height:54px;width:150px}
  </style>
  <a href=//www.google.com/><span id=logo aria-label=Google></span></a>
  <p><b>404.</b> <ins>That's an error.</ins>
  <p>The requested URL <code>/404</code> was not found on this server.  <ins>That's all we know.</ins>
```

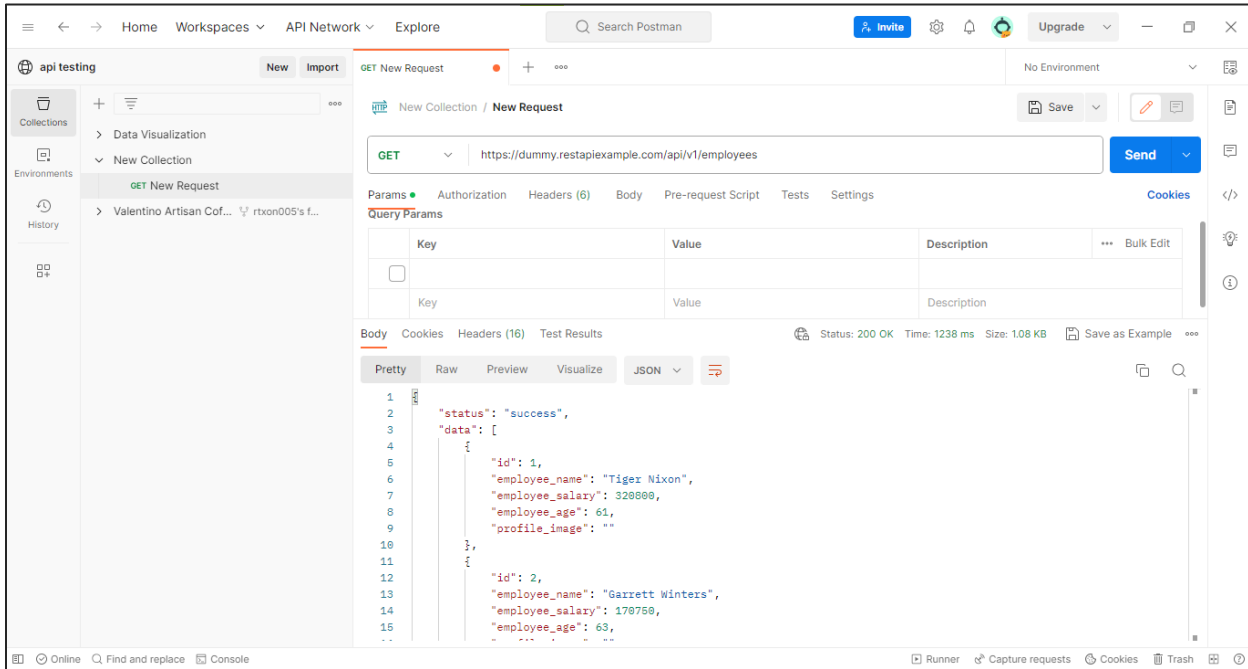## 17. -v --sslv3 ->Checking if the server supports a specific protocol.

```
Command Prompt

C:\Users\user>curl -v --sslv3 https://www.google.com/
Warning: Ignores instruction to use SSLv3
*   Trying [2404:6800:4009:80e::2004]:443...
* Connected to www.google.com (2404:6800:4009:80e::2004) port 443 (#0)
* schannel: disabled automatic use of client certificate
* ALPN: offers http/1.1
* ALPN: server accepted http/1.1
* using HTTP/1.1
> GET / HTTP/1.1
> Host: www.google.com
> User-Agent: curl/8.0.1
> Accept: */*
>
* schannel: failed to decrypt data, need more data
* schannel: failed to decrypt data, need more data
< HTTP/1.1 200 OK
< Date: Thu, 31 Aug 2023 09:45:37 GMT
< Expires: -1
< Cache-Control: private, max-age=0
< Content-Type: text/html; charset=ISO-8859-1
< Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-AZr1CnmnSQVhEIIZFI6KzQ' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsaf
e-inline' 'https: http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp
< P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
< Server: gws
< X-XSS-Protection: 0
< X-Frame-Options: SAMEORIGIN
< Set-Cookie: 1P_JAR=2023-08-31-09; expires=Sat, 30-Sep-2023 09:45:37 GMT; path=/; domain=.google.com; Secure
< Set-Cookie: AEC=Ad49MVFTKPpRwT3slw6GqMqd41dWyUxBI7g98PjSzmk-UzC82IPj_B3uHKc; expires=Tue, 27-Feb-2024 09:45:37 GMT; path=/; domain=.google.com; Secure; HttpOnly; Same
Site=lax
< Set-Cookie: NID=511=taDoUfLfNT2mbuOp5AwGs4PM3g-k30ifLM1eUCJie9x4-bX8IYVR_1mOuuu7toDIvLQ9P_FA6b8bajb2pf-G2zTy7PJA_f4Q2hKrzxt5Lalv37LOpKUDAbeLeZnvdH8qIOaxk5V0zb-tr4PJWG
gfD7J7lrjSrBZT21S136KJWGg; expires=Fri, 01-Mar-2024 09:45:37 GMT; path=/; domain=.google.com; HttpOnly
< Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
< Accept-Ranges: none
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
<
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta conte
nt="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="AZr1CnmnSQVhEIIZFI6KzQ">(function(){var _g={kEI:'
QwHwZLudC7yz1sQP1auL2Ag',kEXPI:'0,1359409,6058,207,4804,2316,383,246,5,1129120,1197743,380747,16115,28684,22431,1361,12311,17588,4998,17075,6085,31559,2872,2891,11754,6
06,29843,825,19391,10631,16105,230,1014,1,16916,2652,4,3832,29062,13065,13658,4437,22610,6627,7596,1,42154,2,16737,23024,5679,1020,31123,4567,6256,23421,1252,25135,8,79
21,2,2,1,10957,13669,2006,8155,7381,2,15967,873,19634,7,1922,9779,20640,21819,20199,20136,14,82,13333,6873,8377,3787,4261,10912,579,4824,3030,6110,9706,1804,7734,5622,4
50,2262,4943,12086,7421,8199,283,3,7661,893,868,4733,545,3147,8566,1992,5922,5769,3626,3347,3,9,3689,440,1271,976,1437,1179,5203197,6942,2,195,47,196,5994623,2803118,34
52,795,2,19733,1,1,346,7540,828,34,45,23942037,579,2737885,1322315,39283,2262,1754,1396641,23759270,886,7064,1,212,84,3337,6915,629,2079,2879,1595,1089,2465,3185,3244,2
```
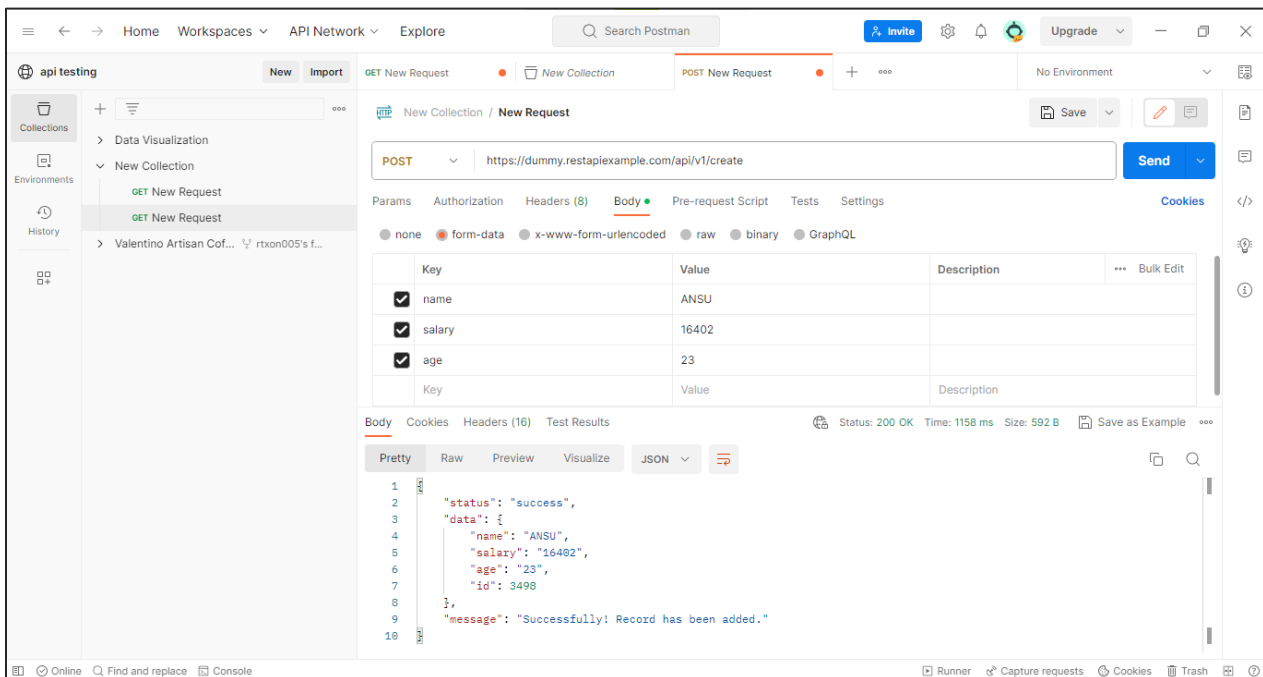
18. -k -> To avoid the certificate checking for a specific site.

19. -6 -> To check if a certain site is configured to use ipv6.

20. -k -filen_name -> It Provides different configuration for different url requests.

POSTMAN

1. GET

2. POST

3. PUT

4. PATCH

5. DELET

6. HEAD