

1. Status codes are issued by a server in response to a client's request made to the server. All HTTP response status codes are separated into five classes or categories. *1xx informational response, 2xx successful, 3xx redirection, 4xx client error, 5xx server error.*

1. 101 switching protocols: The requester has asked the server to switch protocols and the server has agreed to do so.

2. 103 Early Hints: Used to return some response headers before final HTTP message

3. 417 Expectation Failed: The server cannot meet the requirements of the Expect request-header field

4. Misdirected Request: The request was directed at a server that is not able to produce a response

5. 404 Not Found: The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.

2.

1. GET: The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

2. HEAD: Same as GET, but transfers the status line and header section only.

3. POST: A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms

4. PUT: Replaces all current representations of the target resource with the uploaded content.

5. DELETE: Removes all current representations of the target resource given by a URI.

6. CONNECT: Establishes a tunnel to the server identified by a given URI.

7. OPTIONS: Describes the communication options for the target resource.

8. TRACE: Performs a message loop-back test along the path to the target resource.

3. `wget -S --spider example.com` used

Shows the last modified date without downloading any file

4. I used `telnet towel.blinkenlights.nl sh` and it shows star wars story in ASCII characters

5. DNS resource record is a description of a Domain

Running `nslookup ucsc.edu` results in:

Server: 127.0.1.1

Domain: 127.0.1.1#53

Address: 128.114.109.5

6. `nslookup -type=ns .` queries a record of given domain →. this shows 13 root server details

7. They can be identified using a port address. Every application has a different address.

8. Windowing is done to ensure how many packets are sent at a time. Windowing is used to control flow of packets between two networks.

9. MTU: is the maximum transport unit. If the packet size is bigger than MTU, then packets are broken down, and reassembled on the receiver's side.

10.

The image shows a Wireshark packet capture interface. The filter is set to 'tcp'. The packet list shows three packets: a TCP segment (No. 215), an ICMP Echo Reply (No. 216), and another TCP segment (No. 217). The selected packet is No. 217, a TCP segment from 127.0.0.1:46896 to 127.0.0.1:6633, Seq: 145, Ack: 145, Len: 0. The packet details pane shows the following information:

- Frame 200: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
- Transmission Control Protocol, Src Port: 46896 (46896), Dst Port: 6633 (6633), Seq: 145, Ack: 145, Len: 0
 - Source port: 46896 (46896)
 - Destination port: 6633 (6633)
 - [Stream index: 2]
 - Sequence number: 145 (relative sequence number)
 - Acknowledgment number: 145 (relative ack number)
 - Header length: 32 bytes
 - Flags: 0x010 (ACK)
 - Window size value: 86
 - [Calculated window size: 86]
 - [Window size scaling factor: -1 (unknown)]
 - Checksum: 0xfe28 [validation disabled]
 - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - SEQ/ACK analysis
 - [This is an ACK to the segment in frame: 199]
 - [The RTT to ACK the segment was: 0.000000000 seconds]

The packet bytes pane shows the raw data of the packet, with a hex dump and ASCII representation.

Transmission Control Protocol (...) Packets: 270 · Displayed: 254 (94.1%) Profile: Default

Re

https://www.tutorialspoint.com/http/http_methods.htm
https://en.wikipedia.org/wiki/List_of_HTTP_status_codes