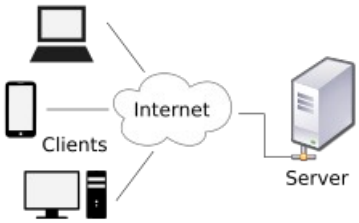
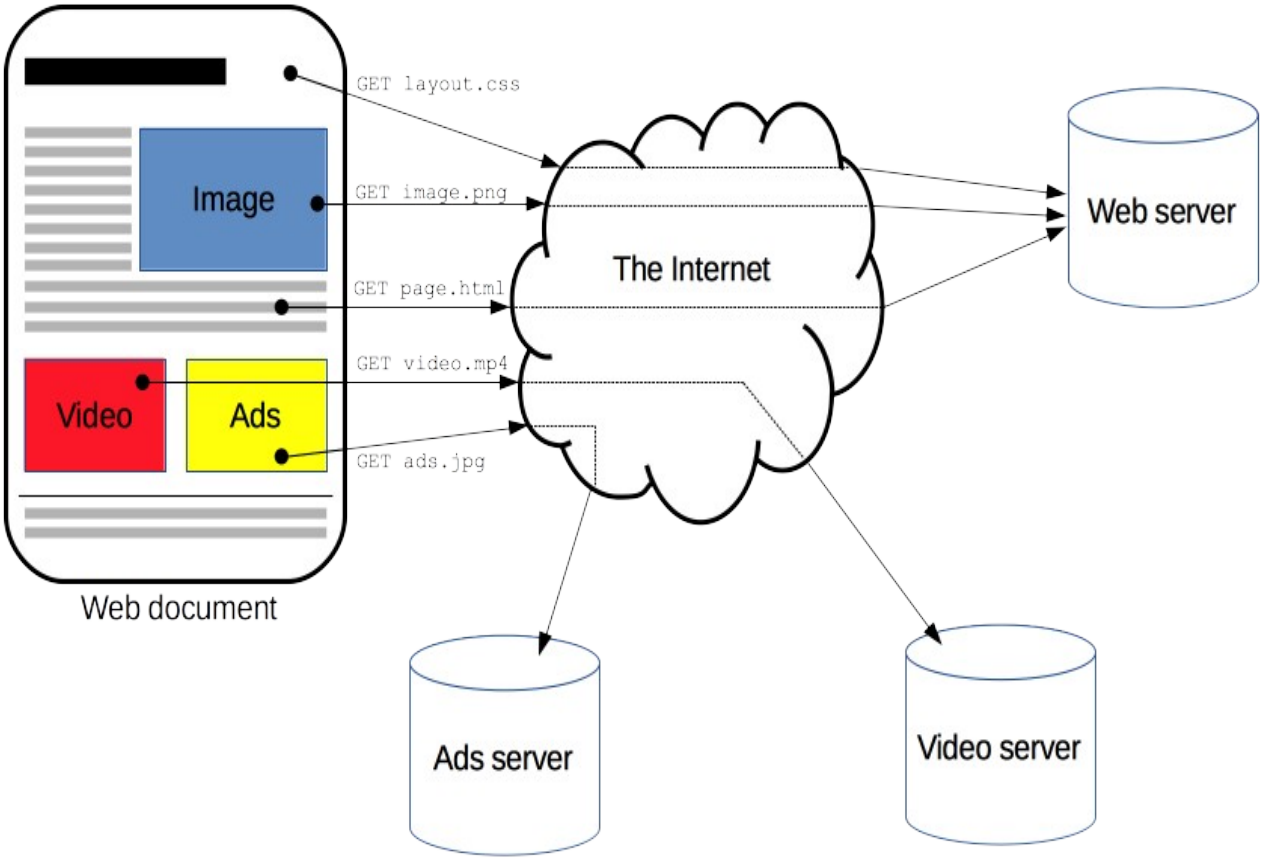
Notes

Client server model



Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.[1] Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but it requests content or service from a server. Clients, therefore, initiate communication sessions with servers, which await incoming requests. Examples of computer applications that use the client-server model are email, network printing, and the World Wide Web.

HTTP



Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML.

- It was designed for communication between web browsers and web servers, but it can also be used for other purposes.
- HTTP follows a classical client-server model, with a client opening a connection to make a request, then waiting until it receives a response.
- HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests.

HTTP allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.

Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called requests and the messages sent by the server as an answer are called responses.

HTTP as an application layer protocol, on top of TCP (transport layer) and IP (network layer) and below the presentation layer.Designed in the early 1990s, HTTP is an extensible protocol which has evolved over time. It is an application layer protocol that is sent over TCP, or over a TLS-encrypted TCP connection, though any reliable transport protocol could theoretically be used. Due to its extensibility, it is used to not only fetch hypertext documents, but also images and videos or to post content to servers, like with HTML form results. HTTP can also be used to fetch parts of documents to update Web pages on demand.

HTTP Headers

HTTP headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (:), then by its value. Whitespace before the value is ignored.

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

HTTP Request Methods

HTTP defines a set of request methods to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as HTTP verbs. Each of them implements a different semantic, but some common features are shared by a group of them: e.g. a request method can be safe, idempotent, or cacheable.

GET

- The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

POST

- The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.

PUT

- The PUT method replaces all current representations of the target resource with the request payload.

DELETE The DELETE method deletes the specified resource.

Request Methods

HTTP Response Codes

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

- Informational responses (100–199),
- Successful responses (200–299),
- Redirects (300–399),
- Client errors (400–499),
- Server errors (500–599).

```
200 - OK
201 - Created
202 - Accepted
404 - Not Found
405 - Method Not allowed
408 - Request Timed out
500 - Internal Server Error
503 - Service unavailable
```

curl

curl is used in command lines or scripts to transfer data. It is also used in cars, television sets, routers, printers, audio equipment, mobile phones, tablets, settop boxes, media players and is the internet transfer backbone for thousands of software applications affecting billions of humans daily.

Basic curl commands:

- Curl head request

```
curl -I https://www.google.com
```

- Curl head request with verbose

```
curl -v -I https://www.google.com
```

- Curl with explicit http method

```
curl -X GET https://www.google.com
```

- Curl has no timeout by default

```
curl --connect-timeout 10 -I -k https://www.google.com
```

- Curl get with extra headers

```
curl --verbose --header "Host: www.mytest.com:8182" www.google.com
```

- Curl get response with headers

```
curl -k -v https://www.google.com
```

- Curl post request

```
curl -d "name=username&password=123456" <URL>
```

- Curl post send json

```
curl <URL> -H "Content-Type: application/json" -d "{ \"woof\": \"bark\"}"
```

- H can also be replaced --header
- Curl with explicit http method

```
curl -X POST -H "Content-Type: application/json" -d "{ \"woof\": \"bark\"}" <URL>
```

Pre-requisites for the class

Download Postman and gitbash

Postman

Gitbash