Research

KEVIN SHAH

# HTTP and basic working of web

1. HTTP (Hyper Text Transfer Protocol) is a protocol for transferring data over a network.

It exchanges data via requests and responses. A client (commonly a browser) sends a request over the network which is received by a host (commonly a web server). The server returns a response to the client according to the request.

A request consists of

* HTTP version
* URL
* method
* headers
* body (optional)

A response consists of

* Status code
* Headers
* Body (optional)

1. There are many HTTP methods. Some commonly used methods are:

* GET: It is used to request data (read only)

It is commonly used to request web pages. It is also used to requesting data such as user info.

* POST: It is used to modify, create, or send data

It is commonly used for registration and posting data such as images or text.

* PUT: It is used to modify existing data

It can be used to update user data.

1. On sending a request to [www.google.com](http://www.google.com), here are the request and response data.

Request URL: https://www.google.com/

Request Method: GET

Status Code: 200

Remote Address: 142.250.192.132:443

Referrer Policy: strict-origin-when-cross-origin

The user agent is:

user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36

1. The user-agent string is used by the host to send a specific copy of data to the client according to the capabilities of the client. As the client might not have newer technologies implemented, the host might send an older or backwards compatible version of the data.

Mozilla, short for Mosaic killer, was introduced as a competitor to the NCSA Mosaic. It supported frames (a container that displays content independently) and so servers could send latest copies of web pages to Mozilla by checking the user-agent string. Future browsers such as Internet Explorer declared itself Mozilla compatible via its user-agent string to receive the most up to date pages and so till date we have Mozilla in the user-agent string.

Gecko was the rendering engine used by Mozilla and so other browsers also impersonated gecko in their user-agent string to receive up to date pages.

# HTTP Cookies

1. Cookies are data that are used to uniquely identify a user. When a user logs in to a web page or performs some similar action, a cookie is sent on the first login which is stored by the browser. The cookie is valid for a short amount of time and one does not have to log in again till the cookie is deleted or has expired.

The host (web server) sends a set-cookie header along with the response which is then saved by the browser. The browser sends it with each subsequent request to uniquely identify itself.

Storing a cookie allows the user to use the web page without having to login after each request. It is unique to the user and browser and so it is useless if stolen and is quite secure.

1. A cookie begins with a key-value pair followed by some optional attributes. Some of them are:

* Expires: to determine max lifespan of cookie (expects date)
* Max-Age: to determine max lifespan (expects number of seconds)

Setting them lower reduces the chance of an unauthorized use. For example: banks may expire a cookie in a few minutes whereas Wikipedia may have no expiry dates.

* Secure: cookie is only valid if an https request is made (has security benefits)
* SameSite: allows a cookie from another site.

This can be used to prevent cross-site request forgery (CSRF) attacks.

It can also be used for advertising and accepting logins from social media accounts such as google and facebook.

1. As cookies can be used to uniquely identify a user, companies such as google and facebook who have a lot of data on a particular user, can provide advertising services to websites to provide user specific ads to generate revenue.
2. There is a big concern of privacy breach as companies get access to user behaviors via cookies. FLoC is an attempt by google to increase anonymity by grouping people into cohorts of similar interests. Ad companies will only get to know your cohort and as there will be a lot of people in a particular cohort, ad companies will have less information on a particular user.

Since FloC will use unsupervised learning to group people into cohorts, past experiences suggest grouping will be done on sensitive characteristics such as gender, race, age, financial status etc.

# CORS

1. Cross-Origin Resource Sharing (CORS) allows data to be received from other origins such as other servers, sites etc. CORS mechanism is required when data from several domains is to be sent to clients. For example, third party ad services, embedded youtube videos, different frontend and backend servers etc.
2. Some headers set in CORS are:

* Access-Control-Allow-Origin: Origins from which to accept data
* Access-Control-Allow-Methods: (ex: POST, GET) Allowed request methods
* Access-Control-Allow-Headers: Allowed header in request

1. CORS preflight requests are first made to the servers of different origins. If CORS is allowed on those servers, the servers send back the headers mentioned above. These requests are sent during the first request and then after Access-Control-Max-Age has been exceeded.

A CORS preflight request begins with OPTIONS to validate the request. It contains the above headers and other information such as origin, user-agent, host, accept-language etc.