Name: Hemank Bajaj
Department: Computer Science and Engineering
Year: 1st Year
Entry Number: 2020CS10349
Hostel: Girnar
Contact No.: 6284776699
Email: cs120039@cse.iitd.ac.in

1. HTTP and basic working of the web.

What is HTTP?

Hyper Text Transfer Protocol is a set of protocols that controls the flow of information over the internet. It is the foundation of data exchange(HTML documents, images, etc) between client and the server. A complete document is fetched by combining the different entities fetched(text, layout, images, scripts, etc.)

How does HTTP exchange messages between servers and clients?

HTTP is a client-server protocol i.e. client sends 'HTTP requests' and in response to these requests the servers send data such as scripts, HTML documents, images,etc. Now the browser recombines these components to show the complete webpage to the user.

What are the different HTTP methods that can be specified in a request?

Following list shows the various HTTP methods and their uses:

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.
POST: A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms
OPTIONS: Describes the communication options for the target resource.
PUT: Replaces all current representations of the target resource with the uploaded content.
CONNECT: Establishes a tunnel to the server identified by a given URI.
HEAD: Same as GET, but transfers the status line and header section only.
DELETE: Removes all current representations of the target resource given by a URI.

What is the User-Agent header for the request?

The **User-Agent** request header is a characteristic string that lets servers and network peers identify the application, operating system, vendor, and/or version of the requesting user agent. In our case , i.e. the user agent shown in the chrome->developer tools->networks tab->header for any HTTP request , it is the browser that requests the HTTP Request.

What does Mozilla mean in this context? What about Gecko?

HTTP response detects the browser through which request is made and serve accordingly. The web should be accessible to all. In order to make this possible almost all the browser make request with user agent as ‘Mozilla/5.0’.
Gecko is a web browser engine developed by mozilla to read and render webpages. Gecko in user agent shows that the browser is based on Gecko engine.

2. Research and document how web cookies work

What are Cookies?

Cookies are text files with small pieces of data like username and password which are used to identify our computer on a network. Web Cookies(HTTP cookies) are a special type of cookies built for web browsers to track, personalize and save information about user’s session. Session is the time spent by the user on a website.

How does a website set a cookie and why would it do that?

Cookies are usually chosen and first sent by the web server, and stored on the client computer by the web browser. The browser then sends them back to the server with every request, introducing states (memory of previous events) into otherwise stateless HTTP transactions.
Cookies are set using the Set-Cookie HTTP header, sent in an HTTP response from the web server. This header instructs the web browser to store the cookie and send it back in future requests to the server (the browser will ignore this header if it does not support cookies or has disabled cookies).

As HTTP is a stateless mode of communication, there should exist a medium to store these states across page loads and browser session. For example, if I set my github in dark mode, I should be able to see it in the dark mode itself across the entire session I am using their website.
Also, Web Pages use cookies in order to track our preferences and interests and thus aim to improve our web browsing experience.

What are the different attributes that can be applied while setting a cookie? What do they achieve?

Cookie Attributes are various parameters that can be set for a cookie. Following points explain the various types of cookies and their use cases:

domain

Includes the Domain attribute in the cookie to specify the domain for which the cookie is sent.

path

Includes the Path attribute in the cookie to specify the path for which this cookie is sent.

secure

Includes the Secure attribute in the cookie to set whether the cookie is transmitted only over an encrypted connection. Since it allows , cookies to be transmitted only through encrypted channels, it helps in maintaining the security of sensitive user data.

httponly

Includes the HttpOnly attribute in the cookie to set whether the cookie is exposed only through HTTP and HTTPS channels. There can be various client-side scripts that may aim at stealing cookie data. If we set this attribute we prevent client side scripts to access the cookies and thus protect data.

max-age

Includes the Max-Age attribute in the cookie to specify the duration of the cookie.

expires

Includes the Expires attribute in the cookie to set the cookie expiration date. It signifies how long the browser should use the persistent cookie and when the cookie should be deleted. It plays a very important role in the safety of cookie data because if the cookie is permanent then an attacker may get user data and might be able to authenticate into the system.

custom

Includes custom attributes in the cookie to specify additional attributes.

How do cookies assist in advertising companies (such as Google) being able to track users across different sites?

Cookies help advertising agencies to show advertisements of products or services that a user shows interest in. Advertisements of those brands are also shown which users are already familiar to. Cookies can also be used to track the search preferences of users and thus show ads accordingly. By showing personalized ads cookies help in attracting more customers.

What are the privacy concerns with cookies?

Though cookies might seem to be an advantageous method to deploy in the stateless communication of HTTP but that is not the complete picture. Cookies do keep our privacy at stake. Ther are cookies which are solely used for tracking purposes. They track the activity of the user, get their personal data and distribute it to various websites who use this data for their own benefits.

Also, there can be a case that the server side does not set appropriate attributes to cookies which can lead to attackers stealing personal data of the users.

How would FLoC system solve these issues?

Under the FLoC system the web browser studies the recent browsing pattern of an individual and groups people with similar interests under a cohort. Now these cohorts are shown similar ads based on their browsing patterns. The main aim of cohorts is to hide data of individuals in a large group of individuals. FLoC system is quite appealing because it is about 95% as effective as the third party cookies method. Moreover, it helps in maintaining user privacy as well.

What are the criticisms against this system?

1. As of now, only Google seems to be incorporating the FLoC system in their chrome browser. Therefore, on other browsers advertising will still be making use of the FLoC system.
2. Another issue is that of Fingerprinting(process of collecting as much information from the browser to create a unique, stable identifier for that browser). Now since FLoC already creates cohorts based on browsing history. Fingerprinting is in fact made easier.
3. The technology will share new personal data with trackers who can already identify users. For FLoC to be useful to advertisers, a user's cohort will necessarily reveal information about their behaviour.
4. It is criticized by advertising agencies by the argument that under this system a user won't be able to see personalized ads but will see the ads that have been suited for cohort that he falls under. This is likely to be a setback for advertising. Moreover, the ad agencies can also not measure their performances.
5. A further problem with FLoC becoming a standard to replace other means of targeting users is that Google will have the ability to tweak the FLoC algorithms to their benefit.

3. Read about Cross-Origin Resource Sharing (CORS)

When is the CORS mechanism required?

CORS mechanism is required when we need to incorporate data in our webpage by requesting a server other than the server we requested for. Under this mechanism, our browser allows sharing of resources. The resource-providing server needs to tell the browser "This origin where the request is coming from can access my resource". The browser remembers that and allows cross-origin resource sharing. API Services are an example of CORS.

What are the headers set in CORS and how do their values affect things?

Suppose a server needs to share a resource from server B, it will first make a preflight request to server B. After this the srever B verifies this call and sends a response with various HTTP headers. The most common ones are access-control-allow-origin , access-control-allow-headers and access-control-allow-method.

The origin header tells which domains(server A) are allowed to share resources. By default its value is * i.e. any server can access it. If we set its value to a domain, then only the requests from that domain are answered. The method header tells to which type of HTTP request it responds to (e.g GET,POST,OPTIONS). The allow-headers header states the allowed headers in the main HTTP request. If the header in the main HTTP request do not match, the request is not executed.

What are CORS preflight requests? When are these requests sent? How does a CORS preflight request look like?

CORS preflight request is a CORS request that checks to see if the CORS protocol is understood and a server is aware using specific methods and headers.

Preflight request are sent from server A to a server B before the main request is made for resource sharing. Preflight request basically, seeks a response in the form of header sets in order to see whether the main request can be made or not. If anything included in the header is false then the actual request is not made.

A preflight request has the following structure:

It clearly states the ORIGIN of the request, the method of the request , headers included in the request.

For e.g.

```
OPTIONS /resource/foo

Access-Control-Request-Method: DELETE

Access-Control-Request-Headers: origin, x-requested-with

Origin: https://foo.bar.org
```

BIBLIOGRAPHY

1. https://www.cloudflare.com/en-gb/learning/ddos/glossary/hypertext-transfer-protocol-http
2. https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
3. https://www.tutorialspoint.com/http/http_methods.htm
4. https://developer.mozilla.org/en-US/docs/Web/HTTP/Browser_detection_using_the_user_agent
5. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent
6. https://webaim.org/blog/user-agent-string-history/
7. https://www.youtube.com/watch?v=eesqK59rhGA
8. https://oko.uk/blog/web-cookies
9. https://www.kaspersky.com/resource-center/definitions/cookies
10. https://en.wikipedia.org/wiki/HTTP_cookie#:~:text=cookies%20in%20total.-,Setting%20a%20cookie,cookies%20or%20has%20disabled%20cookies).

11. https://www.freecodecamp.org/news/everything-you-need-to-know-about-cookies-for-web-development/
12. https://digitalkites.com/blog/cookie-based-advertising/#:~:text=Display%20ad%20through%20email%20retargeting,ads%20on%20third%20party%20sites.&text=It%20allows%20the%20brand%20to%20understand%20how%20users%20engage%20with%20their%20website.
13. https://www.mondaq.com/turkey/privacy-protection/784926/are-cookies-a-threat-for-our-privacy#:~:text=The%20main%20types%20of%20cookies,the%20website%20you%20are%20visiting.
14. https://www.allaboutcookies.org/privacy-concerns/?__cf_chl_captcha_tk__=a7b76ca03dc513ac7a5c0fa205895c0e44fb84c4-1619284090-0-AZ1nGVGkOinw6iAC_umfhdLUw3WbjE83uLlRL1R5dkfLuwXgAJX6Fm5lhm-AgGbMPrAJhAvim9zxg5wYNYk9yQuE-UNjJ6VLPQxCrQlnf2zk1RnKq4mFCQ9G18MUSaDhhYBu0HN_5xP-rGv-y_EYLTvuX9NSwQ6x0LBer8su6re76forFzcyP53eKUTgEVWsN8sBD7dwnhJMFaDnT6VyHRaX0VwSb4osF2i4sbuSyOrc4eDixjKb-8LouV-vVxxO9HngFEjTJQcfDnPkh16_OZ8Abj3V5Y4Njgu0fjKimffBWy5xDRKo-litksTPjB7G-O_STVcTiDnnsD5I8DYRiOMgPqqFIJ9ECFm876E1fDXdmGeDH8B0TI8BrXrsK-D5J2QbKndhvbKKCU2cnzghGA91_9j1gTk2cd-sZaI9emR_6yK6dDMaU4BW9jj_6RG98FgKUHRfTZQR543czrJ4VGlg4H72BaikperpjhiCsH2dmPV1Z9HOVYA1Lw1vIc5HXnFf1FQbolGznjW5kAyHPim6MMTTj0pKwMyB9hb4eDOIjMqTo0x1_sHIvRmXSe6xgEebWYtIRKQz-5hFGJZZSfXrR6XXm973Byg8xILn7A_cavkz_tKJfNpwMOixEDx0lDKUVXvkx3Uc88ACBcVzZkeWKoaN6ktd6Lba_Dp7vdYY
15. https://www.dbswebsite.com/blog/website-cookies-and-data-privacy/
16. https://blog.google/products/ads-commerce/2021-01-privacy-sandbox/
17. https://web.dev/floc/
18. https://whatsnewinpublishing.com/6-problems-with-googles-floc-and-1-silver-lining/
19. https://www.eff.org/deeplinks/2021/03/googles-floc-terrible-idea
20. https://developer.mozilla.org/en-US/docs/Glossary/Preflight_request
21. https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
22. https://www.youtube.com/watch?v=tcLW5d0KAYE
23.