Q1: How does https Works behind the scene?

Ans: HTTPS uses an encryption protocol to encrypt communications. The protocol is called Transport Layer Security (TLS), although formerly it was known as Secure Sockets Layer (SSL). This protocol secures communications by using what’s known as an asymmetric public key infrastructure. This type of security system uses two different keys to encrypt communications between two parties:

- The private key - this key is controlled by the owner of a website and it’s kept, as the reader may have speculated, private. This key lives on a web server and is used to decrypt information encrypted by the public key.
- The public key - this key is available to everyone who wants to interact with the server in a way that’s secure. Information that’s encrypted by the public key can only be decrypted by the private key.

How does TLS work?

For a website or application to use TLS, it must have a TLS certificate installed on its origin server (the certificate is also known as an "SSL certificate" because of the naming confusion described above). A TLS certificate is issued by a certificate authority to the person or business that owns a domain. The certificate contains important information about who owns the domain, along with the server's public key, both of which are important for validating the server's identity.

A TLS connection is initiated using a sequence known as the TLS handshake. When a user navigates to a website that uses TLS, the TLS handshake begins between the user's device (also known as the *client* device) and the web server.

During the TLS handshake, the user's device and the web server:

- Specify which version of TLS (TLS 1.0, 1.2, 1.3, etc.) they will use
- Decide on which cipher suites (see below) they will use
- Authenticate the identity of the server using the server's TLS certificate

- Generate session keys for encrypting messages between them after the handshake is complete

The TLS handshake establishes a cipher suite for each communication session. The cipher suite is a set of algorithms that specifies details such as which shared encryption keys, or session keys, will be used for that particular session. TLS is able to set the matching session keys over an unencrypted channel thanks to a technology known as public key cryptography.

The handshake also handles authentication, which usually consists of the server proving its identity to the client. This is done using public keys. Public keys are encryption keys that use one-way encryption, meaning that anyone with the public key can unscramble the data encrypted with the server's private key to ensure its authenticity, but only the original sender can encrypt data with the private key. The server's public key is part of its TLS certificate.

Q2: What are the different https methods available and what they do?

Ans:

1. GET

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

2. HEAD

The HEAD method asks for a response identical to a GET request, but without the response body.It request the headers rather than the whole body. In case of large file download and user just want to know the filesize. Then without downloading the file we can make use of content-length headers by using head method.

3. POST

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

4. PUT

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

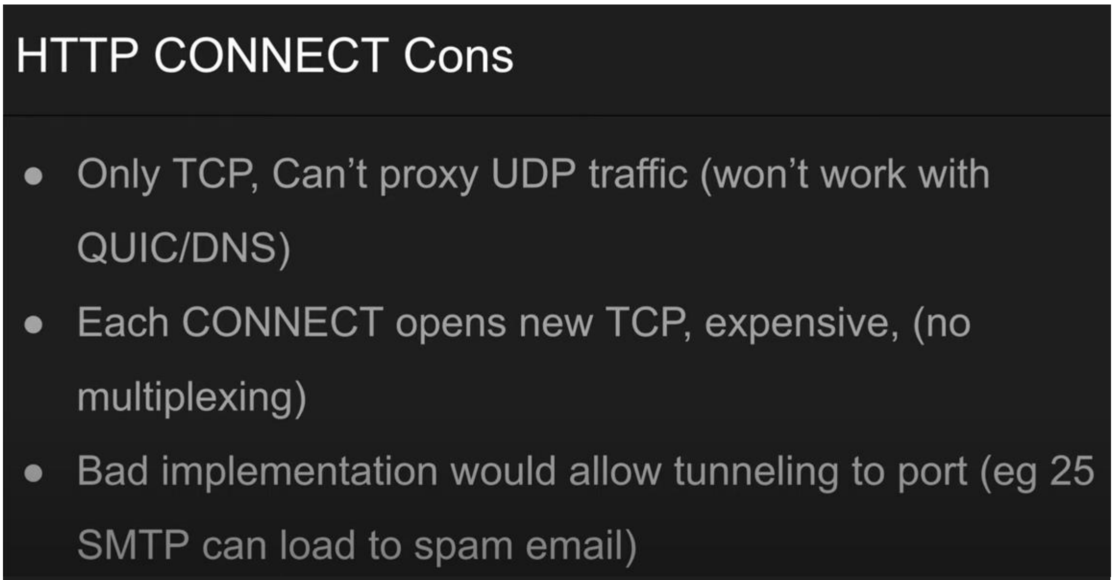
5. DELETE

The DELETE method deletes the specified resource.

6. CONNECT

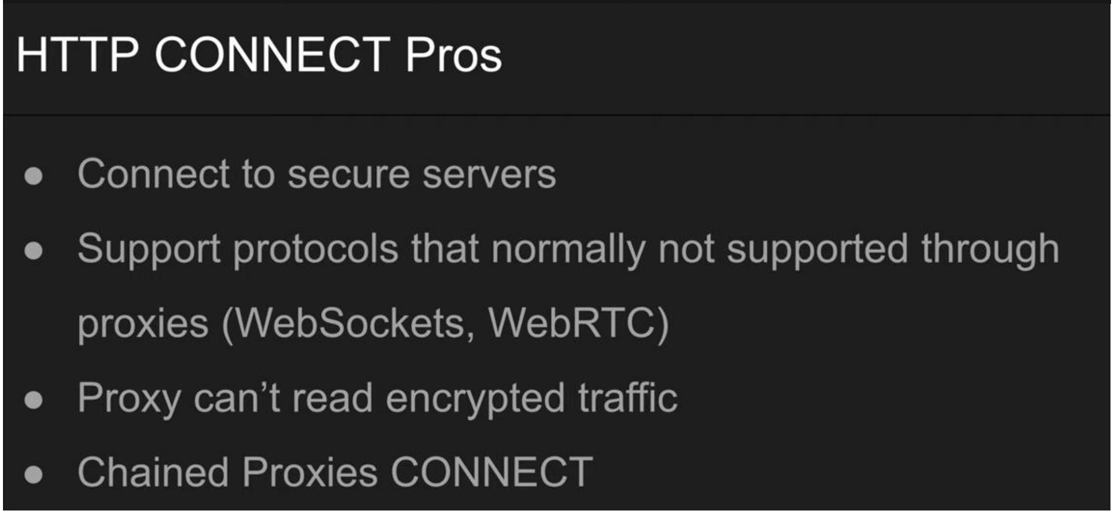
The CONNECT method establishes a tunnel to the server identified by the target resource.

he HTTP CONNECT method starts two-way communications with the requested resource. It can be used to open a tunnel.
For example, the CONNECT method can be used to access websites that use SSL (HTTPS). The client asks an HTTP Proxy server to tunnel the TCP connection to the desired destination. The server then proceeds to make the connection on behalf of the client. Once the connection has been established by the server, the Proxy server continues to proxy the TCP stream to and from the client.



HTTP CONNECT Cons

- Only TCP, Can’t proxy UDP traffic (won’t work with QUIC/DNS)
- Each CONNECT opens new TCP, expensive, (no multiplexing)
- Bad implementation would allow tunneling to port (eg 25 SMTP can load to spam email)



HTTP CONNECT Pros

- Connect to secure servers
- Support protocols that normally not supported through proxies (WebSockets, WebRTC)
- Proxy can’t read encrypted traffic
- Chained Proxies CONNECT

7. OPTIONS

The OPTIONS method describes the communication options for the target resource.

find out which request methods a server supports, one can use the curl command-line program to issue an OPTIONS request:

8. TRACE

The TRACE method performs a message loop-back test along the path to the target resource.

The HTTP TRACE method performs a message loop-back test along the path to the target resource, providing a useful debugging mechanism.

The final recipient of the request should reflect the message received, excluding some fields described below, back to the client as the message body of a 200 (OK) response with a Content-Type of message/http. The final recipient is either the origin server or the first server to receive a Max-Forwards value of 0 in the request.

9. PATCH

The PATCH method applies partial modifications to a resource.

Q3: Understand and explain the use of various http response codes.

Ans: 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)
- Redirection messages (300–399)
- Client error responses (400–499)
- Server error responses (500–599)

Q4: What are the different web communication protocols and their use cases?

Ans:

Let's discuss each of them briefly:

1. **Transmission Control Protocol (TCP):** TCP is a popular communication protocol which is used for communicating over a network. It divides any message into series of packets that are sent from source to destination and there it gets reassembled at the destination.
2. **Internet Protocol (IP):** IP is designed explicitly as addressing protocol. It is mostly used with TCP. The IP addresses in packets help in routing them through different nodes in a network until it reaches the destination system. TCP/IP is the most popular protocol connecting the networks.
3. **User Datagram Protocol (UDP):** UDP is a substitute communication protocol to Transmission Control Protocol implemented primarily for creating loss-tolerating and low-latency linking between different applications.
4. **Post office Protocol (POP):** POP3 is designed for receiving incoming E-mails.
5. **Simple mail transport Protocol (SMTP):** SMTP is designed to send and distribute outgoing E-Mail.

6. **File Transfer Protocol (FTP):** FTP allows users to transfer files from one machine to another. Types of files may include program files, multimedia files, text files, and documents, etc.
7. **Hyper Text Transfer Protocol (HTTP):** HTTP is designed for transferring a hypertext among two or more systems. HTML tags are used for creating links. These links may be in any form like text or images. HTTP is designed on Client-server principles which allow a client system for establishing a connection with the server machine for making a request. The server acknowledges the request initiated by the client and responds accordingly.
8. **Hyper Text Transfer Protocol Secure (HTTPS):** HTTPS is abbreviated as Hyper Text Transfer Protocol Secure is a standard protocol to secure the communication among two computers one using the browser and other fetching data from web server. HTTP is used for transferring data between the client browser (request) and the web server (response) in the hypertext format, same in case of HTTPS except that the transferring of data is done in an encrypted format. So it can be said that https thwart hackers from interpretation or modification of data throughout the transfer of packets.

Q5: Pros and cons of Single page and multi page applications.

Ans:
Pros of a Single-page Application:

- **Performance**. All resources are loaded during one session, and then, when interacting with the page, only the necessary data is changed. This approach significantly increases web app performance.

- **Improved user experience**. Such apps provide users with a more understandable linear experience. Moreover, the use of AJAX and JavaScript frameworks, as well as the fact that there is only one web page, allows building a more flexible and responsive interface.

- **Data caching**. After the first request to the server, all the necessary local data is stored in the cache, and that provides users with the possibility to work in an offline mode (for example, GoogleDocs offline mode).

- **Development speed**. All things equal, you will have to develop and test fewer app elements and will be able to reuse part of the code.

- **Ease of debugging**. An SPA is most often developed based on popular frameworks (React, Vue.js, AngularJS) that offer their own debugging tools based on Google Chrome, for example, Vue.js devtools.

Besides, we should note they are just as convenient to use on mobile devices as native mobile apps.

Cons of a Single-page Application:

- **Problems with SEO**. Any web app runs in JavaScript, and the data is loaded without reloading the page and only at users' demand. This means there are no separate URLs optimized for search engines, and search engines do not see the content. Exclusively server-side rendering can solve the problem.

- **Downloading time**. If the platform is complex, large, and poorly optimized, the users' browsers will take more time to load the content.

- **JavaScript support is necessary**. Without this feature, you cannot fully use the complete functionality of a certain app. If users disable JS in their browser, they will not be able to use the app to its fullest.

Pros of a Multi-page Application:

- **SEO optimization is possible**. The app has multiple pages, and each of them can be optimized for a specific group of requests to get free organic traffic from Google.

- **Ease of scaling**. This architecture type allows creating as many new pages for each product or service as you like and implementing any changes in them.

- **Available ready-made solutions**. As a rule, MPA development requires a smaller technology stack, and besides, a wide range of ready-made solutions (CMS) are available.

- **Analytic capabilities**. Web analytics tools like Google Analytics can be easily integrated into this type of project and allow tracking each business page’s performance.

Cons of the Multi-page Application:

- **Possible performance issues**. In case of a large number of requests and the necessity to reload a large number of pages, performance and speed will inevitably take a knock. This is especially true for projects with high website traffic, a large number of pages, and multiple functions.

- **Front-end and back-end tight integration**. As a rule, these components of a web app are deeply integrated, and that is why it can take longer to develop and test them.

- **Maintenance and updates**. It can be a daunting task to provide technical support to websites with a lot of pages. This issue also applies to security matters.