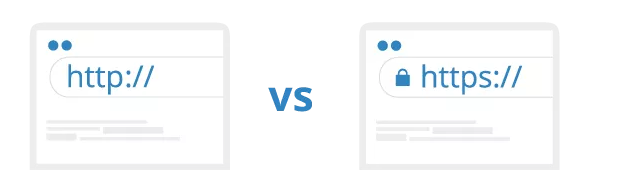
**What is HTTPS?**

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data web browser and a website. HTTPS is encrypted in order to increase security of data transfer, which makes it important when users are transferring sensitive data.

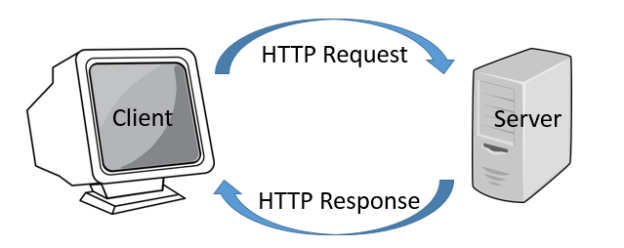


**What is HTTP?**

HTTP is a protocol which 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, meaning requests are initiated by the recipient(client), usually a Web browser and response messages are sent by the server usually a web server.

**How does HTTP work?**

HTTP communicates a web-browser and a web server by using requests and responses.



**HTTP request**: this is the way a web browser asks the web server the information it needs to load a website. Typical HTTP request contains:

1. HTTP version
2. a URL
3. HTTP method
4. HTTP request headers
5. Optional HTTP body

**HTTP method:**

This is the action that the HTTP request expects from the web server. The most common HTTP methods are GET and POST, where GET is used to get information from the web server and POST is used to submit information from the client to the web server.

**HTTP request header:**

HTTP headers contain information about the request and are stored in key-value pairs. They hold information such as what browser the client is using, what data is being requested, what the URL of the host is. Etc



**HTTP response**: this is what the client receives from the web server as an answer to a request. This gives the web browser the information it needs to load the web page. Typical HTTP responses contain:

1. HTTP status code
2. HTTP response headers
3. Optional HTTP body

**HTTP status code**:

HTTP status code is a 3-digit code that indicated the resulting status of the request. There are 5 types of HTTP status codes:

1. 1xx Informational
2. 2xx Success
3. 3xx Redirection
4. 4xx Client Error
5. 5xx Server Error

2XX status code means the request was completed successfully. 4xx and 5xx status code means that the request was unsuccessful due to an error and the webpage cannot be displayed. 4xx means is a client-side error usually a typo when writing the URL. 5xx means there was a server-side error. 1xx and 3xx status code indicate informational or redirection. 1xx status code usually is just a wait response by the web server until it prepares the final response. 3xx status code means the client is being redirected to anther URI.

**HTTP response body**:

HTTP response comes with headers that convey important information such as the language and format of the data being sent in the response body.

**HTTP response body**:

HTTP responses with 2xx status code to GET requests have a body that contains the requested information which usually is the HTML data that a web browser converts to a web page.

**How does HTTPS work?**

HTTPS uses an encryption protocol to encrypt communications. The protocol is called Transfer Layer Security (TLS), also known as Secure Sockets Layer (SSL). This protocol works by using what is called an asymmetric public key infrastructure. This system uses two different keys to transfer data between two parties securely:

The private key- this is the key the owner of the website holds and is kept private. This key lives in a web server and is used to decrypt information that is encrypted by the public key.

The public key – this key is available to anyone that access the web server in a secure way. Information that is encrypted by the public key can only be decrypted by the private key.

**Why use HTTPS?**

HTTPS prevents websites from having information easily seen by anyone spying on the network. Communication over HTTP occurs in plain text, which makes it very easy to steal information with someone with the right tools and weak to on-path attackers.

**Questions we still don’t know:**

How does TSL encryption work?

How do the different xx values of HTTP status code within the 1xx,2xx,3xx,4xx,5xx status codes differentiate?

Aside from encryption what differences do HTTP and HTTPS have?

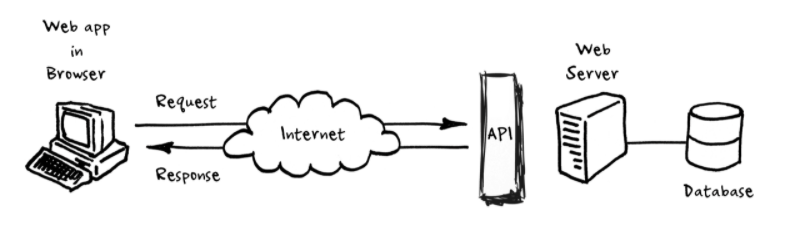
What are URIs and what do they hold?

**What is REST API/RESTful API?**

A RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the creating, reading, updating, and deleting (CRUD) of operations concerning resources.

**What is an API?**

An API or Application Programming Interface is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.



**What is the REST architecture principles?**

REST stands for Representational State Transfer, a term coined by Roy Fielding in 2000 in his PHD dissertation. It is an architecture style for designing loosely coupled applications over HTTP, which is often used in the development of web services.

REST defines 5 architectural constraints:

1. Uniform interface
2. Client-server
3. Stateless
4. Cacheable
5. Layered system

**Uniform interface**: The idea behind this is that you stick to finite set of operations of the application protocol the service uses. For most cases it is HTTP which means the services are restricted to using the GET, PUT, POST, and DELETE methods.

**Client-server**: This just means that the client side and the server side of the application MUST exist separately without dependency on each other.

**Stateless**: This simply means that the web server doesn’t store anything about past HTTP requests or responses. It treats each request as new.

**Cacheable**: This means the data in a response from the web server be labeled as cacheable or non-cacheable. If cacheable the client cache can use the response data for corresponding requests.

**Layered system**: This simply just means the client does not know if it is connected directly to the end server or an intermediary along the way(there might be layers between the client and end server).

HTTP follows the architectural constraints of REST which makes it the best and most efficient protocol to use for web service REST APIs

**Questions I don’t know:**

How does REST APIs differ from other types of APIs?

Where are cached HTTP responses stored?

Is there anyway to make REST API somehow stateful for the end-user or stateful sessions?

Could a REST API use a protocol that isn’t HTTP?