

# Introduction au HTTP, le protocole de transfert hypertexte

Le protocole de transfert hypertexte ou HTTP (HyperText Transfer Protocol) est, avec le langage HTML et les adresses web l'une des trois fondations qui permettent au World Wide Web tel qu'on le connaît aujourd'hui d'exister.

HTTP (HyperText Transfer Protocol ou Protocole de Transfert Hypertexte en français) est un protocole de communication permettant de récupérer et d'envoyer des ressources hypermédia. Au départ, HTTP a été créé pour permettre de transférer des documents HTML uniquement puis ses usages se sont rapidement étendus pour permettre de transférer d'autres types de ressources comme des images, des vidéos ou d'autres types de fichiers.

Un protocole est tout simplement un ensemble de règles. Dans un contexte réseau, un protocole est un ensemble de règles qui régissent les échanges de données.

## Pourquoi apprendre comment fonctionne HTTP ?

Comprendre comment fonctionne HTTP n'est pas obligatoire au sens strict du terme pour développer des pages web. Cependant, cela vous permettra d'avoir une bien meilleure compréhension du fonctionnement du web en général et de « ce qu'il se passe en arrière-plan » lors d'un échange web et cela vous aidera également à mieux comprendre certaines erreurs qui peuvent survenir sur un site et comment les résoudre.

De la même manière, il est intéressant de comprendre le modèle réseau complet qui permet aux informations d'aller d'un point d'origine à un point d'arrivée dans lequel s'insère HTTP. Cela vous permettra d'avoir une bonne vision d'ensemble du fonctionnement de l'échange de données entre deux machines.

## HTTP, un protocole de transfert client – serveur

Le protocole HTTP est basé sur un modèle client-serveur. Dans les modèles de communication client-serveur, le client est toujours celui qui demande à accéder (demande de récupérer) une ressource tandis que le serveur est toujours celui qui va répondre à la demande (qui va « servir » la ressource).

Dans la majorité des cas, le client va être un navigateur. Par exemple, lorsqu'un visiteur cherche à accéder à l'une de nos pages web, il va taper une URL dans son navigateur. Le navigateur va alors demander au serveur sur lequel est stockée la page de la lui renvoyer. La demande du client est appelée une requête HTTP.

Un serveur est un ordinateur très puissant donc l'unique rôle est de stocker des ressources et de les servir aux clients lorsque ceux-ci les demandent. Pour se faire, un serveur doit être toujours allumé et connecté à d'autres serveurs de manière à pouvoir être accessible.

Notez ici qu'une ressource peut être divisée en plusieurs parties et que chaque partie peut être stockée sur une machine différente. Au contraire, une même machine peut stocker des ressources appartenant à différents sites. On utilisera généralement le terme « le serveur » de manière indiscriminée pour désigner une machine à part entière, une partie d'une machine ou plusieurs machines à partir du moment où leur rôle est de stocker et de servir des ressources.

Les requêtes et les réponses HTTP (qu'on appelle également messages HTTP) sont au cœur du protocole de transfert hypertexte. Nous étudierons en détail comment elle sont créées et de quoi elles sont composées dans la suite de ce cours.

## HTTP, un protocole sans état et évolutif

HTTP est l'une des trois inventions majeures qui ont permis la création du web. HTTP est donc présent depuis la création du web et a su évoluer jusqu'à aujourd'hui pour répondre à l'évolution et à la diversification des usages du web et pour faire en sorte qu'ils soient possibles.

Au début du web, les pages n'étaient composés que de HTML et ne faisaient pas appel à des ressources externes. Le client n'avait donc qu'à effectuer une requête au serveur afin que celui-ci lui renvoie le document HTML.

Aujourd'hui, les pages web sont composées de différentes ressources : une structure HTML, des ressources CSS, JavaScript, des images, vidéos, etc.

Les usages se sont également diversifiés : au départ du web, la seule opération possible en tant qu'utilisateur était la récupération et la visualisation d'une ressource. Aujourd'hui, on peut également envoyer ou modifier des informations (via des formulaires HTML notamment).

Tout cela est possible car HTTP a su évoluer et ajouter de nouveaux composants / fonctionnalités pour fournir davantage de possibilité d'échanges. Toutefois, comme les usages et les capacités du web étaient au départ très limités, les règles composant HTTP n'ont pas été formulées avec une quelconque idée de performance ou d'optimisation et cela représente le défi principal de HTTP.

Par exemple, deux des principes de base du HTTP sont qu'une requête ne peut récupérer qu'une seule ressource à la fois / n'effectuer qu'une action à la fois et que HTTP est un protocole sans état.

Lorsqu'on dit que HTTP est un protocole sans état, cela signifie que HTTP n'a pas besoin que le serveur conserve des informations sur un client entre deux requêtes. Autrement dit, chaque nouvelle requête peut agir de manière totalement indépendante et n'a pas de lien à priori avec les requêtes précédentes ou suivantes.

Les pages web actuelles sont composées de nombreuses ressources et vont donc demander au client d'effectuer plusieurs requêtes. Généralement, le client va demander qu'on lui renvoie le document HTML demandé, puis va commencer à le lire et va ensuite effectuer de nouvelles requêtes au serveur à chaque fois qu'il va tomber sur une autre ressource incluse dans le document.

Chaque nouveau jeu de requête-réponse implique un aller-retour des données, ce qui prend du temps et entraîne donc de la latence (un temps d'attente) pour l'utilisateur. Des techniques comme les connexions parallèles, persistantes ou en pipelines ont été progressivement formulées et implémentées pour résoudre ces problèmes de performance. Nous étudierons évidemment ces techniques en détail dans la suite du cours.

Pour le moment, contentez-vous de retenir que l'un des enjeux majeurs du HTTP depuis quelques années est un enjeu lié à l'optimisation des performances pour « rendre le web plus rapide ».

# HTTP et les modèles TCP/IP et OSI

HTTP est un protocole qui définit la forme de la communication entre deux machines, c'est-à-dire la syntaxe des messages envoyés afin que les deux machines puissent effectivement communiquer.

Cependant, HTTP n'indique pas comment les messages doivent transiter pour partir d'une machine A et arriver à une machine B.

En fait, HTTP est un protocole qui ne sert à définir qu'une partie de la communication entre deux machines et qui s'insère dans un modèle réseau beaucoup plus large.

Aujourd'hui, il existe deux modèles réseaux de référence : les modèles modèle OSI (Open Systems Interconnection) et TCP/IP. Ces deux modèles permettent de fournir un cadre global pour la communication entre deux machines.

La chose importante à retenir de ces deux modèles pour le moment est qu'ils définissent tous les deux différentes « couches » de communication. Grosso-modo, ces deux modèles découpent la communication entre deux machines en différentes couches qui vont chacune définir les règles d'un point précis de la communication.

Chaque couche va contenir ses propres protocoles et les protocoles d'un modèle réseau ne vont pouvoir appartenir qu'à une couche précise. L'idée principale derrière ce genre de modèle est de pouvoir faire évoluer les différents protocoles de manière indépendante (sans avoir à modifier l'ensemble du processus de communication) et de nous permettre d'implémenter tel ou tel protocole de telle couche.

Le modèle OSI par exemple est un modèle théorique composé de 7 couches différentes : application, présentation, session, transport, réseau, liaison et physique. Les protocoles du modèle OSI vont pouvoir appartenir à l'une ou l'autre de ces couches.

L'inconvénient du modèle OSI est qu'il reste difficile à appliquer de manière pratique pour le web car certains protocoles vont « logiquement » couvrir plusieurs couches.

Pour cela, un autre modèle a été formulé : le modèle TCP/IP, encore appelé « modèle Internet ». Ce modèle, composé uniquement de 4 couches nommées couches application, transport, internet et accès réseau traduit moins fidèlement la réalité des échanges entre deux machines mais est beaucoup plus pratique à mettre en place. C'est le modèle de référence utilisé pour les échanges sur le web.

Nous allons étudier chacun de ces deux modèles en détail dans la suite du cours. Pour le moment, vous pouvez retenir qu'HTTP est un protocole de la couche application dans chacun de ces deux modèles.

# HTTP et HTTPS

HTTP est un protocole de transfert de la couche application des modèles OSI et TCP/IP. Ce protocole permet in fine l'envoi et la récupération ressources hypermedias.

Il y a quelques années, la plupart des utilisateurs du web ne faisaient pas confiance aux sites et n'osaient pas acheter sur le web car ils devaient transmettre des informations sensibles comme des numéros de carte bleu.

Netscape a alors inventé un protocole de sécurisation des échanges de données appelées SSL pour Secure Sockets Layer.

L'IETF (groupe chargé de la standardisation de certains protocoles et notamment chargé de l'évolution de HTTP) a ensuite continué son développement en le renommant TLS (Transport Layer Security).

TLS est aujourd'hui la norme dans le chiffrement / cryptage des échanges de données sur le net. HTTPS est simplement une variante de HTTP qui implémente ce protocole de sécurisation.

Pour information, un serveur HTTP utilise le port 80 par défaut. Un serveur supportant HTTPS utilisera le port 443 par défaut. Un « port » permet de distinguer différents interlocuteurs (différents programmes ou applications).

---

Révision #1

Créé 2020-04-25 19:59:31 UTC par garfieldtux

Mis à jour 2020-04-25 20:02:00 UTC par garfieldtux