Chapitre 1 : les Bases et technologies de l'internet

Introduction

Internet fait référence à un réseau de réseaux. Dans ce réseau, chaque ordinateur est reconnu par une adresse unique appelée adresse IP. Un DNS (Domain Name Server) d'ordinateur spécial est utilisé pour donner un nom à l'adresse IP afin que l'utilisateur puisse localiser un ordinateur par un nom.

Les technologies Internet sont un domaine technique qui couvre les compétences nécessaires pour développer des applications sur Internet ou des systèmes basés sur Internet, exploitant les technologies du commerce électronique, du cloud, du mobile et du Web.

Modèle de référence :

Un modèle de référence offre un moyen de normalisation dans le domaine de l'informatique. Cela est dû à l'utilisation des périphériques réseau qui peuvent avoir une architecture hétérogène. Afin de fournir une communication entre des appareils hétérogènes, on a besoin d'un modèle standardisé, c'est-à-dire d'un modèle de référence, qui nous fournit la manière dont ces appareils peuvent communiquer quelle que soit leur architecture. On a deux modèles de référence qui sont le modèle OSI (Open Systems Interconnection) et le modèle TCP/IP.

Système de noms de domaine

Lorsque le DNS n'existait pas, il fallait télécharger un fichier hôte contenant les noms d'hôte et leur adresses IP correspondante. Mais avec l'augmentation du nombre d'hôtes Internet, la taille du fichier hôte a également augmenté. Cela a entraîné une augmentation du trafic lors du téléchargement de ce fichier. Pour résoudre ce problème, le système DNS a été introduit. Le système de noms de domaine aide à résoudre le nom d'hôte en une adresse. Il utilise un schéma de dénomination hiérarchique et une base de données distribuée d'adresses IP et de noms associés.

L'adresse IP est une adresse logique unique attribuée à une machine sur le réseau. Une adresse IP présente les propriétés suivantes :

L'adresse IP est l'adresse unique attribuée à chaque hôte présent sur Internet.

L'adresse IP a une longueur de 32 bits (4 octets). Chacun des 4 octets est représenté par un nombre de 0 à 255, séparés par des points. Par exemple 192.168.4.100

L'adresse IP se compose de deux composants : le composant réseau et le composant hôte.

Il y a des services en ligne qui permettent d'afficher l'adresse IP de l'utilisateur connecté, par exemple le service offert par le site 'expressvpn.com' comme montré par la figure 1.

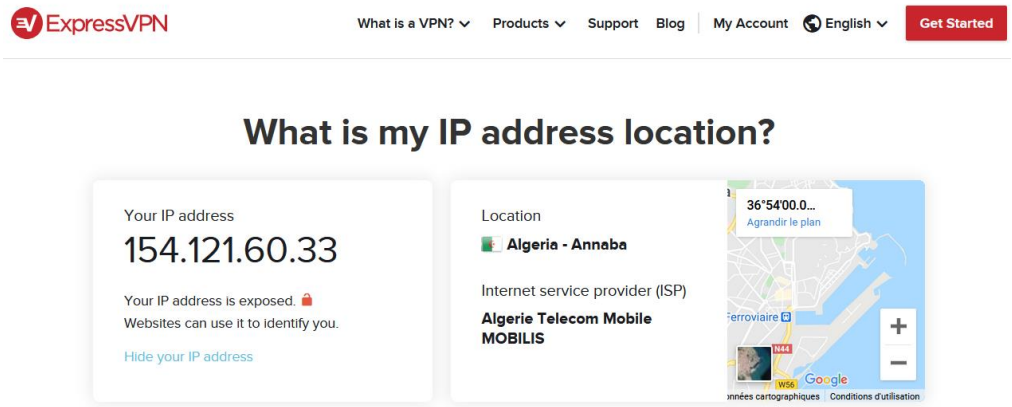


Figure 1. service de localisation et d'extraction d'adresse IP.

Localisateur de ressources uniformes (URL)

Uniform Resource Locator (URL) fait référence à une adresse Web qui identifie de manière unique un document sur Internet. Ce document peut être une page Web, une image, un son, une vidéo ou tout autre élément présent sur le Web. Par exemple : https://facsci.univ-annaba.dz/wp-content/uploads/2013/07/gadm.pdf

L'URL ci-dessus est une URL vers le fichier pdf qui porte le nom 'gadm.pdf' qui est stocké sur le serveur Web univ-annaba.

L'URL Absolut comme celle donnée plus haut, à l'inverse de l'URL relative, est composée de trois parties : le protocole, le nom de serveur, et la ressource.

Parmi les protocoles qu'on peut avoir :

HTTP : Hypertext Transfer Protocol, est un protocole de couche application pour la transmission de documents hypermédia, tels que HTML. Il a été conçu principalement pour la communication entre les navigateurs Web et les serveurs Web.

HTTPS : Hypertext Transfer Protocol Secure, est une combinaison du protocole http avec le protocole Secure Socket Layer (SSL) / Transport Layer Security (TLS). TLS est un protocole d'authentification et de sécurité largement implémenté dans les navigateurs et les serveurs Web.

FTP : File Transfer Protocol, est un protocole réseau standard utilisé pour le transfert de fichiers informatiques entre un client et un serveur sur un réseau informatique.

Le protocole http :

HTTP est un protocole qui permet de récupérer des ressources, telles que des documents HTML. C'est le fondement de tout échange de données sur le Web et c'est un protocole client-serveur, ce qui signifie que les demandes sont initiées par le destinataire, généralement le navigateur Web. Un document complet est reconstruit à partir des différents sous-documents récupérés, par exemple du texte, une description de mise en page, des images, des vidéos, des scripts, etc. comme montré par la figure 2.

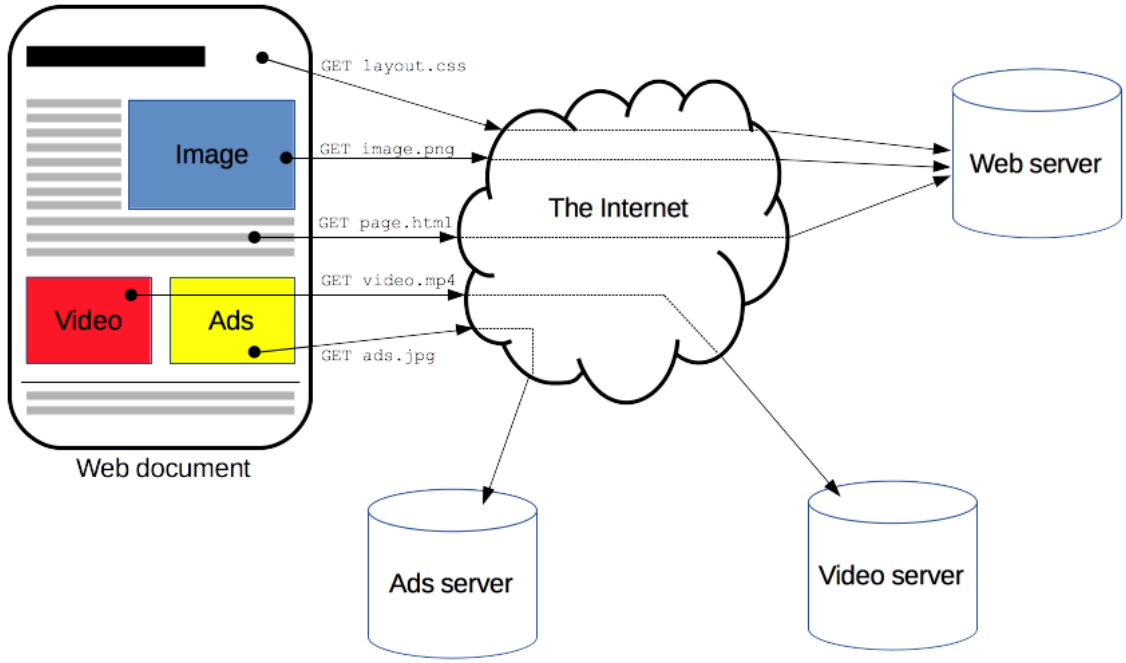


Figure 2. Récupération des données pour la construction d'une page Web via HTTP.

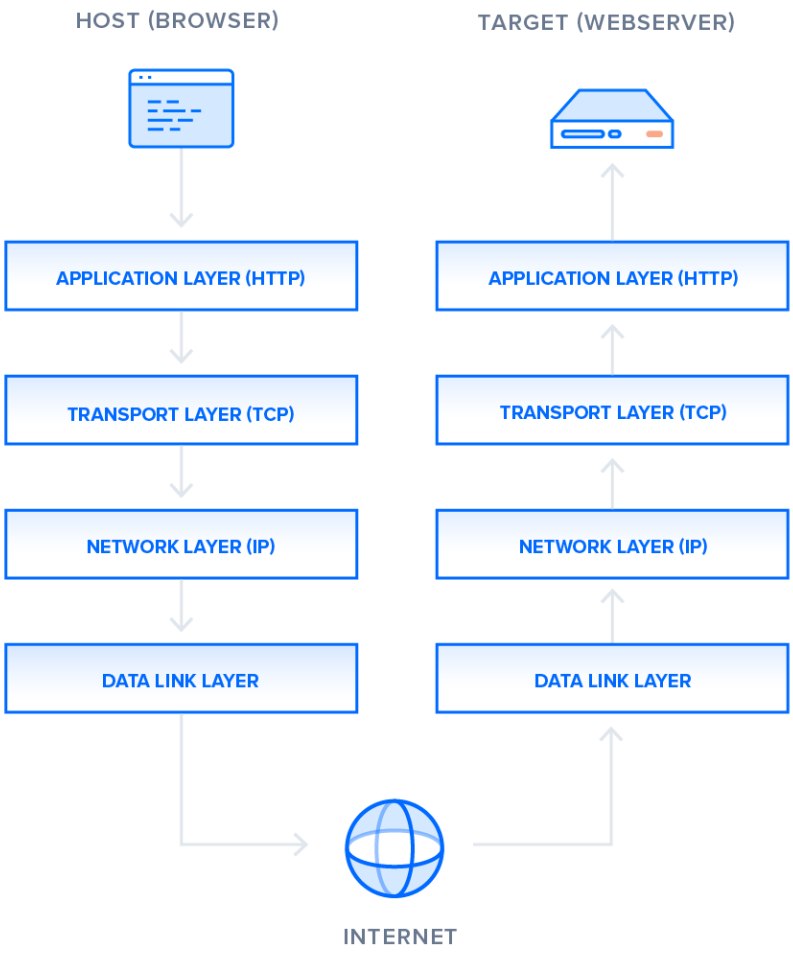
Les clients et les serveurs communiquent en échangeant des messages individuels (à l'inverse d'un flux de données). Les messages envoyés par le client, généralement un navigateur Web, sont appelés requêtes et les messages envoyés par le serveur en tant que réponse sont appelés réponses.

Conçu au début des années 1990, HTTP est un protocole de la couche application qui est envoyé via TCP ou via une connexion TCP cryptée SSL/TLS, bien que n'importe quel protocole de transport fiable puisse théoriquement être utilisé. HTTP est utilisé non seulement pour récupérer des documents hypertextes, mais également des images et des vidéos ou pour publier du contenu sur des serveurs comme avec les résultats de formulaires HTML.

Composants des systèmes basés sur http

HTTP est un protocole client-serveur, les requêtes sont envoyées par l'utilisateur. La plupart du temps, l'agent utilisateur est un navigateur Web, mais il peut s'agir d'une application écrite avec n'importe quel langage de programmation et qui explore le Web ou interroge des serveurs Web.

Chaque demande individuelle est envoyée à un serveur, qui la gère et fournit une réponse. Entre le client et le serveur, il existe de nombreuses entités, appelées collectivement proxies, qui effectuent différentes opérations et agissent comme des passerelles ou des filtres. Pratiquement, il y a plusieurs ordinateurs entre un navigateur et le serveur traitant la demande. Grâce à la conception en couches du Web, ceux-ci sont cachés dans les couches réseau et transport. HTTP est au top, au niveau de la couche application.



Versions du protocole http

D'un point de vue technique, l'une des caractéristiques les plus importantes qui distinguent HTTP / 1.1 et HTTP / 2 est la couche de trame binaire (binary framing layer), qui peut être considérée comme faisant partie de la couche application dans la pile de protocoles Internet. Contrairement à HTTP / 1.1, qui conserve toutes les demandes et réponses au format texte brut, HTTP / 2 utilise la couche de trame binaire pour encapsuler tous les messages au format binaire, tout en conservant la sémantique HTTP. Une API de niveau application créerait toujours des messages dans les formats HTTP conventionnels, mais la couche sous-jacente convertirait ensuite ces messages en binaire. Cela garantit que les applications Web créées avant HTTP / 2 peuvent continuer à fonctionner normalement lorsqu'elles interagissent avec le nouveau protocole. La conversion des messages en binaire permet à HTTP / 2 d'essayer de nouvelles approches de livraison de données non disponibles dans HTTP / 1.1.

Client (l'agent utilisateur)

L'agent utilisateur est tout outil agissant au nom de l'utilisateur. Ce rôle est principalement assuré par le navigateur Web. Pour présenter une page Web, le navigateur envoie une demande pour récupérer le document HTML qui représente la page Web demandée. Il analyse ensuite ce fichier, effectuant des requêtes

supplémentaires correspondant aux scripts d'exécution, aux informations de mise en page (CSS) à afficher et aux ressources contenues dans la page (généralement des images et des vidéos). Le navigateur Web mélange ensuite ces ressources pour présenter à l'utilisateur un document complet, qui est la page Web.
Les scripts exécutés par le navigateur peuvent récupérer plus de ressources dans les phases ultérieures et le navigateur met à jour la page Web en conséquence. Une page Web est un document hypertexte. Le navigateur traduit ces instructions en requêtes HTTP et interprète en outre les réponses HTTP pour présenter à l'utilisateur une réponse claire.

Le serveur Web

De l'autre côté du canal de communication, se trouve le serveur, qui fournit le document comme demandé par le client. Un serveur apparaît virtuellement comme une seule machine, c'est parce qu'il peut en fait être un ensemble de serveurs, partageant la charge (équilibrage de charge) ou un logiciel complexe interrogeant d'autres ordinateurs (comme le proxy, un serveur de base de données, etc.), générant totalement ou partiellement le document demandé.

Proxies

Entre le navigateur Web et le serveur, de nombreux ordinateurs et machines relaient les messages HTTP. Les proxies peuvent être transparents, de manière à transmettre les demandes qu'ils reçoivent sans les modifier, ou non transparents, en modifiant la requête avant de la transmettre au serveur.
Les proxies peuvent remplir de nombreuses fonctions comme:

- les serveurs proxy peuvent crypter les requêtes Web pour améliorer la sécurité. Ils peuvent également empêcher tout site malveillant **connu** d'infecter les ordinateurs des clients.
- Certains serveurs proxy peuvent masquer complètement l'adresse IP de client ou la modifier afin de protéger la vie privée des personnes.
- Contrôler l'usage d'Internet en appliquant des restrictions sur les sites que l'utilisateur peut visiter.

Propriétés du protocole HTTP

simple

HTTP est conçu pour être simple et lisible par l'homme, même avec la complexité supplémentaire introduite dans HTTP / 2.

HTTP est extensible

Introduits dans HTTP / 1.0, les en-têtes HTTP facilitent l'extension et l'expérimentation de ce protocole.

HTTP est sans état, mais pas sans session

HTTP est sans état: il n'y a pas de lien entre deux requêtes exécutées successivement sur la même connexion. Cela risque immédiatement d'être problématique pour les utilisateurs qui tentent d'interagir de manière cohérente avec certaines pages. Mais alors que le cœur de HTTP lui-même est sans état, les cookies HTTP permettent l'utilisation de sessions avec état. En utilisant l'extensibilité d'en-tête, les cookies HTTP sont ajoutés au flux de travail, permettant la création de session sur chaque requête HTTP pour partager le même contexte, ou le même état.

Exemple de requête http

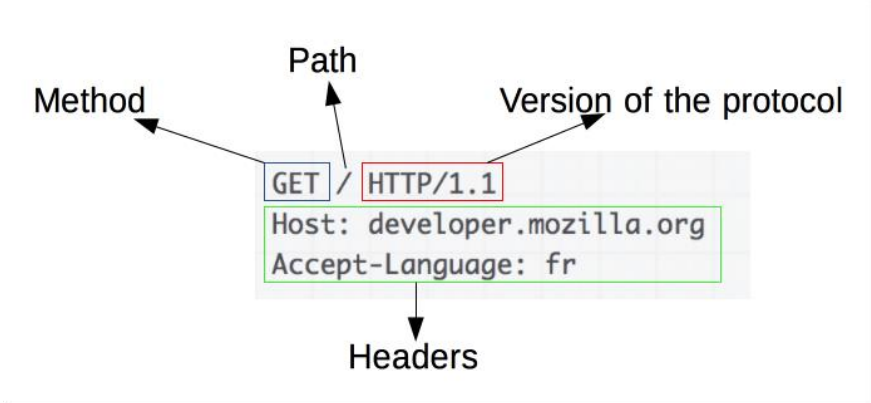


Figure 3. Exemple de requête http.

Les requêtes se composent des éléments suivants:

- Une méthode HTTP, généralement un verbe comme GET, POST ou un nom comme OPTIONS ou HEAD qui définit l'opération que le client souhaite effectuer. En règle générale, un client souhaite récupérer une ressource (en utilisant GET) ou publier la valeur d'un formulaire HTML (en utilisant POST).
- Le chemin de la ressource à récupérer, l'URL de la ressource dépouillée des éléments du contexte, par exemple sans le protocole (http: //), le domaine (ici, developer.mozilla.org), ou le port TCP (ici, 80).
- La version du protocole HTTP.
- Autres attributs facultatifs de l'en-têtes qui transmettent des informations supplémentaires pour les serveurs.
- Un corps dans le cas de certaines méthodes comme POST, il contient la ressource envoyée.

Methodes du protocole Http
GET La méthode GET est utilisée pour récupérer des informations d'un serveur.
HEAD Identique à GET, mais transfère uniquement la ligne d'état et la section d'en-tête.
POST Une requête POST est utilisée pour envoyer des données au serveur, par exemple des informations client, le téléchargement de fichiers, etc. à l'aide de formulaires HTML.
PUT Remplace toutes les représentations actuelles de la ressource cible par le contenu téléchargé.
DELETE Supprime toutes les représentations actuelles de la ressource cible.
CONNECT Établit une connexion vers le serveur identifié par une URL.

OPTIONS Décrit les options de communication pour la ressource cible.
TRACE Effectue un test de retour des messages http qui est utile pour le débogage lors de développement des applications

Exemple de réponse http

Les réponses se composent des éléments suivants:

- La version du protocole HTTP.
- Un code d'état, indiquant si la demande a réussi ou non, et pourquoi.
- Un message d'état, une brève description du code d'état.
- En-têtes HTTP, comme ceux des requêtes.
- Facultativement, un corps contenant la ressource récupérée.

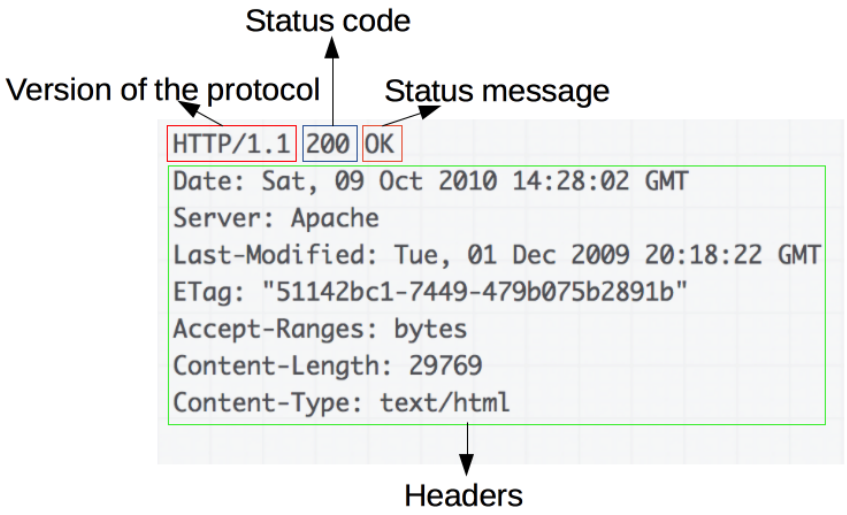


Figure 4. Exemple de réponse http

Travail à faire N°1:

Préparer un tableau qui contient avec explication tous les codes d'état des réponses http, (exemple 404 : ressource non trouvée).