Serveur Web

- matériel (ordinateur) et/ou logiciel (application informatique) qui permet de fournir du contenu Web qui peut être accessible via Internet.
- logiciel appelé serveur HTTP, HTTP daemon ou encore HTTPd.

Fonction principale

- stocker, traiter et livrer des pages Web pour les clients.

Communication entre le client et le serveur

- utilise le protocole HTTP (hypertext transfer protocol).

Pages délivrées

- le plus souvent des documents HTML, qui peuvent inclure des *images*, des *feuilles de style* et des *scripts*, en plus du contenu *texte*.

Fonctionnement, Interaction avec le serveur

- Généralement un navigateur web initie la communication en faisant une demande pour une ressource spécifique en utilisant le protocole HTTP.
- Le serveur répond en servant le contenu de cette ressource ou un message d'erreur.

La ressource est soit un fichier réel situé sur l'unité de stockage du serveur, soit un document créé dynamiquement en réponse à la demande.

Le protocole HTTP permet également au serveur de recevoir des informations en provenance du client.

Cette fonction est utilisée pour transmettre des variables, soumettre des formulaires Web et télécharger des fichiers sur le serveur.

Scripts côté serveur

Beaucoup de serveurs web prennent en charge des scripts côté serveur.
Ils peuvent utiliser ASPX, .NET, PHP, Java ou d'autres langages de script .

Le comportement du serveur Web est programmé dans des fichiers script, tandis que le logiciel serveur demeure inchangé.

Cette fonctionnalité est utilisée pour créer des documents HTML dynamiques ("on-the-fly») au lieu de retourner simplement le contenu de documents statiques.

Elle est principalement utilisé pour les pages nécessitant la récupération et/ou la modification d'informations dans des bases de données.

Utilisation des serveurs Web

Les serveurs Web ne sont pas uniquement utilisés pour le Web.

Ils peuvent également être embarqués dans des dispositifs tels que les imprimantes , les routeurs , les webcams et servir uniquement sur un réseau local.

Le serveur Web peut alors être utilisé comme une partie d'un système de surveillance et / ou d'administration de l'appareil en question.

Cela signifie généralement qu'aucun logiciel supplémentaire ne doit être installé sur l'ordinateur client, puisque seul un navigateur Web (inclus avec la plupart des systèmes d'exploitation) est nécessaire.

URL ou adresse Web

Le sigle URL, provient de l'anglais Uniform Resource Locator.

C'est une chaîne de caractères normalisée permettant de localiser une ressource du Word Wide Web.

Une URL peut être formée de cinq composants :

- un schéma. Exemples : http://, ftp://, mailto:
- le domaine du service, (host name ou IP) complété de manière optionnelle par des données d'authentification et un port. Exemples : www.exemple.com, utilisateur:motdepasse@exemple.fr:8080
- un chemin (path). Exemples : blog/historique, images/logo.png
- une requête (query ≠ request) formé de couples variable=valeur. Exemple : ?page=1&field=name&sort=asc
- un fragment ou signet. Exemple : #top

URL relative

Dans un document possédant une URL, il est possible de définir des URL sans schéma ni domaine.

Dans ce cas, le schéma, le domaine et le début du chemin sont déduits de l'URL du document englobant.

Utilisation des URL

Pour afficher une page Web, un navigateur émet une requête HTTP contenant l'URL de cette page.

Il établit une connexion avec le serveur Web référencé par le domaine de l'URL.

Ce serveur est alors capable d'associer le chemin de l'URL à une des ressources qu'il propose.

Ressource associée à une URL

Cette association est réalisée par le serveur Web.

Répertoire [ROOT]

Tous les fichiers servis par un serveur Web sont contenus dans son répertoire [ROOT].

Exemples :

Pour wampserver, ROOT est le répertoire wamp/www

Pour easyPHP, ROOT est le répertoire EasyPHP/www

Pour Xampp, ROOT est le répertoire xampp/htdocs

Pour IIS, ROOT est le répertoire c:\inetpub

Chez certains hébergeurs, ROOT est le répertoire public_html ou encore web, ...

Demande d'un fichier existant

C'est le cas lorsque la partie chemin de l'URL, ajoutée au chemin du répertoire [ROOT] correspond à un fichier existant dans le système de fichiers du serveur.

Exemples :

URL = http://www.monsite.fr/dfin/wim2/accueil.html

et

[ROOT]/dfin/wim2/accueil.html existe sur le disque dur du serveur

URL = http://www.monsite.fr/dfin/wim2/evaluation.php

et

[ROOT]/dfin/wim2/evaluation.php existe sur le disque dur du serveur

Fichier statique

- Si le fichier existant est un fichier statique, son contenu est simplement renvoyé au navigateur client, précédé de quelques headers.

Script

- Si le fichier existant est un script, celui ci est exécuté.
- Il peut récupérer des informations contenues dans la partie requête de l'URL
- Pendant son exécution, il écrit du texte dans sa sortie standard.
- Le texte produit par le script est finalement renvoyée au navigateur client, précédé de quelques headers.

Remarque : Dans un script PHP, les caractères situées en dehors des balises <?php ?> sont automatiquement recopiés dans la sortie standard.

Demande d'une ressource ne correspondant pas à un fichier

- l'URL ne correspond pas à un fichier,
- mais il existe des règles de réécriture d'URL permettant de l'associer à un script,
- alors ce script est exécuté.

Exemples :

Dans un site Drupal, une URL du type http://monsite.com/cours1/lesson1 est réécrite en monsite.com/index.php?q=cours1/lesson1.
C'est donc le script index.php qui est exécuté. Il reçoit une variable de requête de nom q et de valeur 'cours1/lesson1'

Dans un site utilisant une architecture MVC, une URL du type http://monsite.com/c1/a1 sera traitée par le contrôleur c1 qui utilisera l'action a1.

Réécriture d'URL dans un serveur Apache

La réécriture est une technique couramment utilisée non seulement dans l'architecture de certaines application Web, mais également pour améliorer le référencement naturel des pages web sur les moteurs de recherche.

Dans les serveurs Apache, la réécriture d'URL est réalisée dans les fichiers de configuration .htaccess.
Le module d'URL Rewriting doit être activé.

Exemple :

```
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_URI} !=/favicon.ico
RewriteRule ^ index.php [L]
```

Explication de l'exemple

RewriteEngine on

La directive RewriteEngine active ou désactive l'exécution du moteur de réécriture (valeur on ou off). Ici, il est activé.

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

La directive RewriteCond définit une condition d'application d'une règle.

Syntaxe : RewriteCond [chaîne à tester] [expression de comparaison]

%{ NOM_DE_VARIABLE } permet d'obtenir la valeur d'une variable

REQUEST_FILENAME est la variable qui contient le chemin complet local au système de fichiers du fichier ou du script correspondant à la requête.

! veut dire « ne doit pas correspondre au modèle qui suit »

-d correspond à « est un répertoire qui existe »

-f correspond à « est un fichier régulier qui existe »

RewriteCond %{REQUEST_URI} !=/favicon.ico

REQUEST_URI est la partie chemin de l'URI (l'URL est une URI)

RewriteRule ^ index.php [L]

RewriteRule remplace les chemins qui vérifient le pattern par target

Syntaxe : RewriteRule pattern target [Flag1,Flag2,Flag3]

Le flag [L] indique qu'il s'agit de la dernière (last) règle.