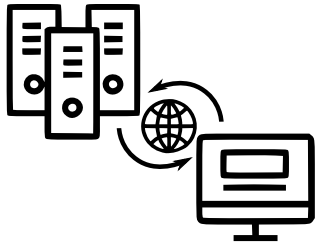
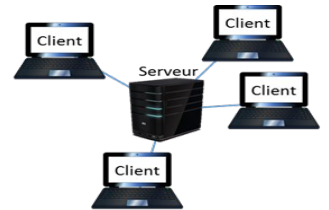


# Le modèle client serveur



Principe d'échange de pages Web sur Internet  
fonctionnement sur le modèle client-serveur :

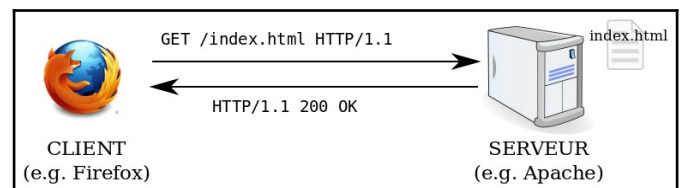
- **Client** : c'est le visiteur du site Web. Il demande la page Web au serveur. En pratique, vous êtes des clients lorsque vous surfez sur le Web. Plus précisément, le navigateur Web (Firefox, Chrome, Safari, IE, Edge, ...) est le client car c'est lui qui demande la page.
- **Serveur(s)** : ce sont les ordinateurs qui délivrent les sites Web aux internautes, c'est-à-dire aux clients.



Le client fait une requête au serveur, qui lui répond en lui envoyant le code **HTML** de la page Web.

## 1 Le navigateur comme client HTTP

- Quand on ouvre une **URL** commençant par **http://**, le navigateur va agir comme un client **HTTP**. Il va donc envoyer une requête **HTTP**.
- Le serveur **HTTP** (appelé aussi serveur **WEB**) renvoie une réponse **HTTP** avec le code **HTML** de la page demandée
- Le navigateur commence à interpréter ce code et à l'afficher.
- Dès qu'une ressource (**CSS**, image, ...) apparaît dans la page, une autre requête **HTTP** est envoyée au serveur
- Le serveur renvoie alors la ressource qui peut être affichée par le navigateur



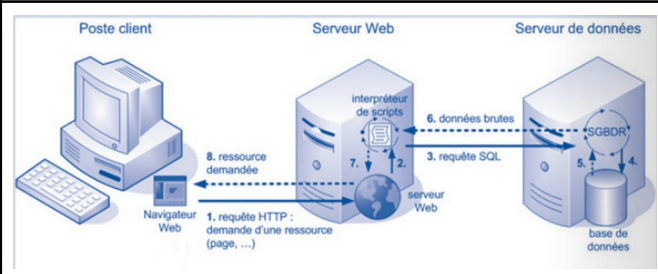
## Architecture à 2-tiers

On appelle architecture à 2-tiers, l'architecture client/serveur au sein de laquelle le client réclame une source et le serveur la lui donne directement sans même solliciter une autre application.

## Architecture 3-tiers, N-tiers

Le serveur web peut échanger avec des serveurs qui servent d'interface entre la partie web et une application spécifique.

Les sites web dynamiques sont souvent 3-tiers



## Sites Internet statiques vs dynamiques

Les sites **statiques** :

- Ce sont des sites réalisés uniquement à l'aide de **HTML/CSS**.
- Ils fonctionnent très bien mais leur contenu ne change pas.
- Les sites statiques sont donc bien adaptés pour réaliser des sites "vitrine".

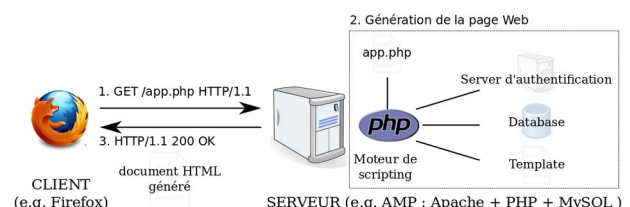
Les sites **dynamiques** :

- Ils utilisent d'autres langages tels que **PHP** pour générer du **HTML** et **CSS**.
- La plupart des sites Web que vous visitez sont dynamiques.
- Fonctionnalités typiques de sites dynamiques :  
un espace membres, un forum, un compteur de visiteurs, des actualités, une newsletter.

**Mécanisme de génération des pages dynamiques :**



### 1- Les langages du World Wide Web



- Le client demande au serveur à voir une page Web (requête **HTTP**) ;
- Le serveur crée la page spécialement pour le client (en utilisant par exemple le langage PHP)
- Le serveur répond au client en lui envoyant la page qu'il vient de générer (réponse HTTP).

# 1 Protocole HTTP

Un protocole de communication est un ensemble de règles qui permettent à des ordinateurs de communiquer ensemble. Le protocole **HTTP (HyperText Transfer Protocol)** va permettre au client d'effectuer des requêtes à destination d'un serveur web. En retour, le serveur web va envoyer une réponse.

## Requête HTTP (du client vers le serveur)

Ligne de requête	GET /monrepertoire/monFichier.html HTTP/1.1	Cette requête <b>HTTP</b> contient les informations suivantes : <ul style="list-style-type: none"> <li>● <b>GET</b> est la méthode employée</li> <li>● <b>/monrepertoire /monFichier.html</b> correspond au chemin de la ressource demandée</li> <li>● <b>HTTP/1.1</b> : la version utilisée du protocole <b>HTTP</b> (1.1 ici)</li> <li>● <b>Mozilla/5.0</b> : le navigateur web employé est Firefox de la société Mozilla</li> <li>● <b>text/html</b> : le client s'attend à recevoir du <b>HTML</b></li> </ul>
Entête de requête	User-Agent : Mozilla/5.0 Accept : text/html	
Saut de ligne		
Corps de requête		

## Réponse du serveur à une requête HTTP (du serveur vers le client)

Une fois que le serveur a reçu la requête du client, le serveur va adresser une réponse au client.

Ligne de requête	HTTP/1.1 200 OK	Voici quelques explications sur la réponse envoyée par le serveur : <ul style="list-style-type: none"> <li>● <b>HTTP/1.1 200 OK</b> : Le code <b>200</b> signifie que la ressource est bien présente sur le serveur. Il existe d'autres code de retour, le plus courant en cas d'erreur étant le code "<b>404 : not found</b>" lorsque la ressource demandée par le client n'existe pas sur le serveur.</li> <li>● <b>Server: Apache/2.0.54 (Debian GNU/Linux) SVN/1.1.4</b> : On a ici une information sur la machine serveur : le type de serveur est un serveur Apache fonctionnant sur le système d'exploitation Linux.</li> </ul>
Entête de requête	Date: Tue, 09 Jul 2019 19:43:31 GMT Server: Apache/2.0.54 (Debian GNU/Linux) SVN/1.1.4 Content-Type: text/html; Content-Length: 3521	
Saut de ligne		
Corps de requête	<!doctype html> <html lang="fr"> <head> ...	

Le code de statut HTTP	Content-Type : type MIME
Un sous-ensemble des codes et textes des réponses: <ul style="list-style-type: none"> <li>• 200 OK</li> <li>• 301 MOVED PERMANENTLY</li> <li>• 308 PERMANENT REDIRECT</li> <li>• 401 UNAUTHORIZED</li> <li>• 403 FORBIDDEN</li> </ul>	Un sous ensemble de type de contenu : <ul style="list-style-type: none"> <li>• text/plain (texte brut)</li> <li>• text/html</li> <li>• text/javascript</li> <li>• text/css</li> <li>• image/png</li> </ul>



- 404 NOT FOUND
- 500 INTERNAL SERVER ERROR

- image/jpeg
- video/mpeg

## La version sécurisée de HTTP : HTTPS :

Pour garantir le chiffrement des données entre le client et le serveur, il existe une version sécurisée du protocole **HTTP**: le protocole **HTTPS**.

- le client (le navigateur Web) contacte un serveur et demande une connexion sécurisée en proposant une liste de méthodes de chiffrement
- Le serveur répond en choisissant dans cette liste une méthode de chiffrement et produit un certificat garantissant qu'il est bien le serveur en question et pas un serveur pirate déguisé.
- Les données échangées ensuite entre le client et le serveur sont ensuite chiffrées grâce à un algorithme de cryptographie.

## Méthode GET ou POST

Parmi les requêtes **HTTP** disponibles (**GET**, **HEAD**, **POST**, **OPTIONS**, **CONNECT**, **TRACE**, **PUT**, **PATCH**, **DELETE**)

- **GET** : C'est la méthode pour demander une ressource.
- **POST** : Cette méthode est utilisée pour soumettre des données en vue d'un traitement côté serveur. C'est la méthode employée lorsque l'on envoie au serveur les données issues d'un formulaire.

Formulaire : méthode GET		Formulaire : méthode POST	
<div>  Non sécurisé   claudia.com/contact.php?nom=toulouse31&amp;satisf=1         </div> <b>URL</b>		<div>  Non sécurisé   claudia.com/contact.php         </div> <b>URL</b>	
<b>Requête HTTP</b>		<b>Requête HTTP</b>	
<b>Ligne de requête</b>	GET /contact.php?nom=toulouse31&satisf=1 HTTP/1.1	<b>Ligne de requête</b>	POST /contact.php HTTP/1.1
<b>Entête de requête</b>	Host: www.claudia.com Accept : text/html, text/CSS, image/jpg	<b>Entête de requête</b>	Host: www.claudia.com Accept : text/html, text/CSS, image/jpg
<b>Saut de ligne</b>		<b>Saut de ligne</b>	
<b>Corps de requête</b>		<b>Corps de requête</b>	nom=toulouse31&satisf=1

```
<h2> Évaluez votre visite </h2>
<form action="contact.php" method="get">
  <p> Pseudo: <input type="text" name="nom"/> </p>
  <p> Votre niveau de satisfaction : </br>
  <input type="radio" name="satisf" value="0"/> Pas satisfait </br>
  <input type="radio" name="satisf" value="1"/> Moyennement satisfait </br>
  <input type="radio" name="satisf" value="2"/> Très satisfait </br>
  <br/>
  <input type="submit" value="Envoyer">
</form>
```

```
<h2> Évaluez votre visite </h2>
<form action="contact.php" method="post">
  <p> Pseudo: <input type="text" name="nom"/> </p>
  <p> Votre niveau de satisfaction : </br>
  <input type="radio" name="satisf" value="0"/> Pas satisfait </br>
  <input type="radio" name="satisf" value="1"/> Moyennement satisfait </br>
  <input type="radio" name="satisf" value="2"/> Très satisfait </br>
  <br/>
  <input type="submit" value="Envoyer">
</form>
```