

Université Assane SECK de Ziguinchor



Unité de Formation et de
Recherche des Sciences et
Technologies

Département d'Informatique

Le développement Web Back-end, qu'est-ce que c'est ?

Licence 2 en Ingénierie Informatique

Avril 2022

©Papa Alioune CISSE

Papa-alioune.cisse@univ-zig.sn

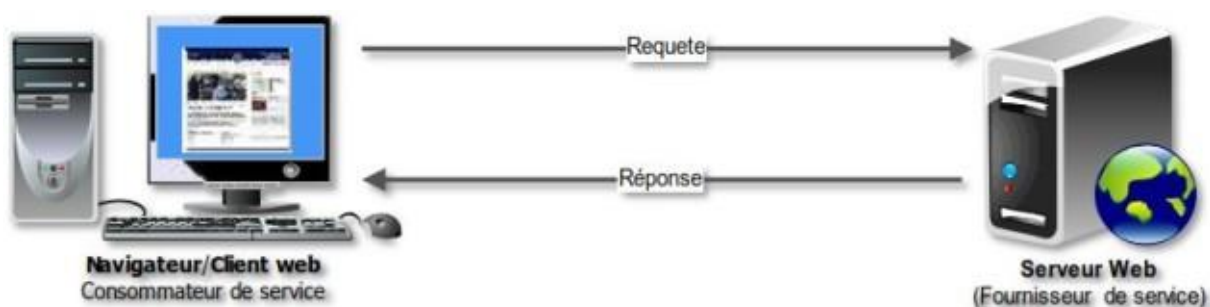
Résumé : À la différence du Développement Front-End, qui concerne l'aspect visuel et ergonomique d'un site web, le Développement Back-End se penche sur les aspects techniques et fonctionnels du développement d'un site web dynamique. Il s'agit du développement de l'ensemble des fonctionnalités "invisibles" d'un site web (le serveur, le code applicatif, la base de données, etc.) qui sont indispensables au bon fonctionnement d'un site. Ainsi, on peut dire que les Développeurs Back End travaillent sur la partie immergée de l'iceberg, alors que les Développeurs Front End se chargent essentiellement de la partie visible. Ce chapitre introduit, sous forme de définitions, un ensemble de notions relatives au développement back-end.

1 - DÉFINITIONS ET EXEMPLES

1.1 - Modèle Client/serveur

Sur le web, tous les objets qui implantent le service www jouent soit le rôle de client, soit le rôle de serveur.

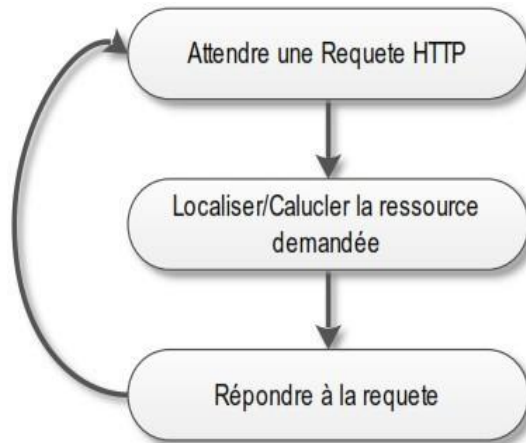
Dans ce modèle, les serveurs sont les fournisseurs de service (Web) et les clients sont les consommateurs de ce service. Pour consommer le service Web, le client envoie une requête au serveur pour accéder aux ressources du Web. Le serveur Web quant à lui héberge les ressources auxquels les clients peuvent accéder (Figure suivante) à travers des requêtes.



1.2 - Serveur web

C'est un logiciel qui gère des ressources qui sont accessibles via le protocole HTTP. Ces ressources peuvent être des fichiers de n'importe quel type (Page Web, Image, Vidéo, Logiciel, ...), stockés sur le système de fichier de la machine qui héberge le serveur ou bien des « *informations calculées* ». Par abus de langage, la machine qui héberge le serveur Web est aussi appelé Serveur Web.

Le fonctionnement d'un serveur Web peut être décrit en trois étapes:



Un serveur Web n'interprète pas les ressources qu'il gère (il n'affiche pas de page ou de vidéos), il se contente de fournir aux clients (les navigateurs), le service d'accès à ces ressources.

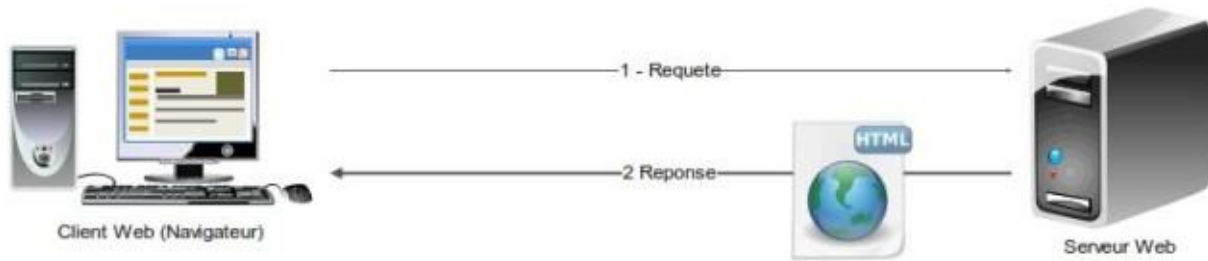
Le marché des serveurs WEB est dominé par quatre logiciels :

- Apache2. Le serveur Web le plus déployé d'Internet. Environ 60% des sites Web tourne sous Apache2. Il est sous licence Apache, c'est-à-dire utilisable quasiment sans contraintes de propriété.
- Microsoft IIS. Le serveur Web de Microsoft. Il est surtout utilisé pour les applications basées sur les technologies Microsoft tel qu'ASP.NET.
- NGINX. Ce serveur prend de plus en plus de place et prétend améliorer les performances par rapport à Apache2.
- Google WebServer. Il s'agit d'un serveur Web basé sur Apache2 et qui a été modifié par les ingénieurs de Google pour leurs besoins. Ces modifications n'ont jamais été divulguées.

1.3 - Pages statiques

En fonction de la manière dont les pages Web sont créés, mises à jour et leur interactivité, il y a principalement deux catégories de pages Web : les pages Web statiques et les pages Web dynamiques.

Les pages web statiques ont un contenu et une mise en forme fixés (codé en dur en HTML) au moment de leur création. L'apparence et le contenu de ce type de pages reste le même pour tous les clients et à tout moment. Pour modifier la page, le développeur Web doit éditer le code source de la page qui va avoir la même apparence et contenu pour tous les clients. Elles ont l'avantage d'être simples, sécurisées (aucun échange de données avec l'utilisateur) et très facilement visibles des moteurs de recherche (URL fixe).

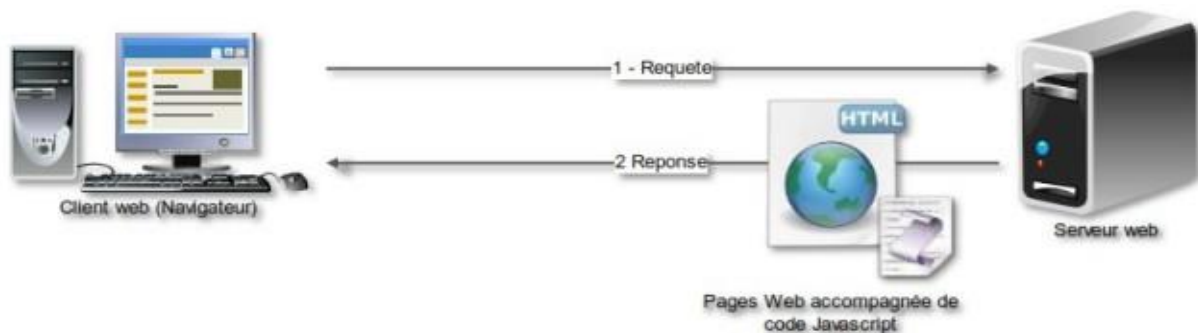


1.4 - Pages dynamiques

Les pages Web dynamiques quant à elles adaptent leur contenu ou apparence selon les interactions avec les utilisateurs, les évolutions des données fournies par l'application. Il y a deux sous-catégories de pages Web dynamiques selon là où leur contenu dynamique est généré : page dynamique coté serveur et page dynamique coté client.

Dans le cas des pages web dynamique coté client, du code binaire ou interprété est exécuté sur le client Web (Navigateur). Aucune interaction avec le serveur n'est nécessaire pour exécuter la page.

Les pages Web dynamiques, avec du code interprété, sont réalisées avec le langage JavaScript. Tous les navigateurs modernes savent interpréter le langage JavaScript qui permet de manipuler le contenu de la page web pour avoir de l'animation, des effets.



Les **pages Web dynamiques coté serveur** peuvent être réalisées en utilisant les scripts CGI ou les scripts coté serveur.

Les scripts CGI (Common Gateway Interface), comme leur nom l'indique sont des interfaces pour permettre d'exécuter un programme écrit dans les langages classiques (C/C++, Python, Perl, Pascal, Bash, ...) pour produire du contenu pour le Web.

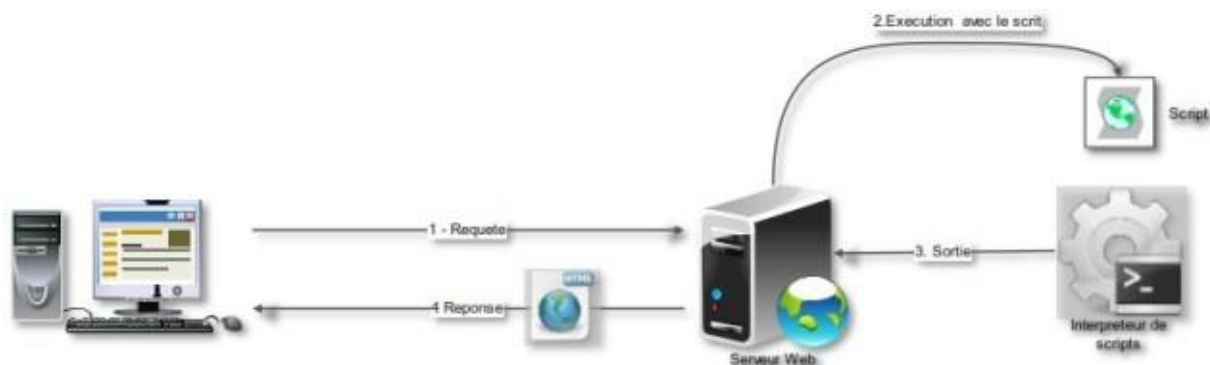


Lorsque le navigateur demande une ressource CGI, le serveur Web exécute le programme correspondant, récupère sa sortie (ce que le programme aurait affiché sur le terminal) puis retourne le contenu généré au client.

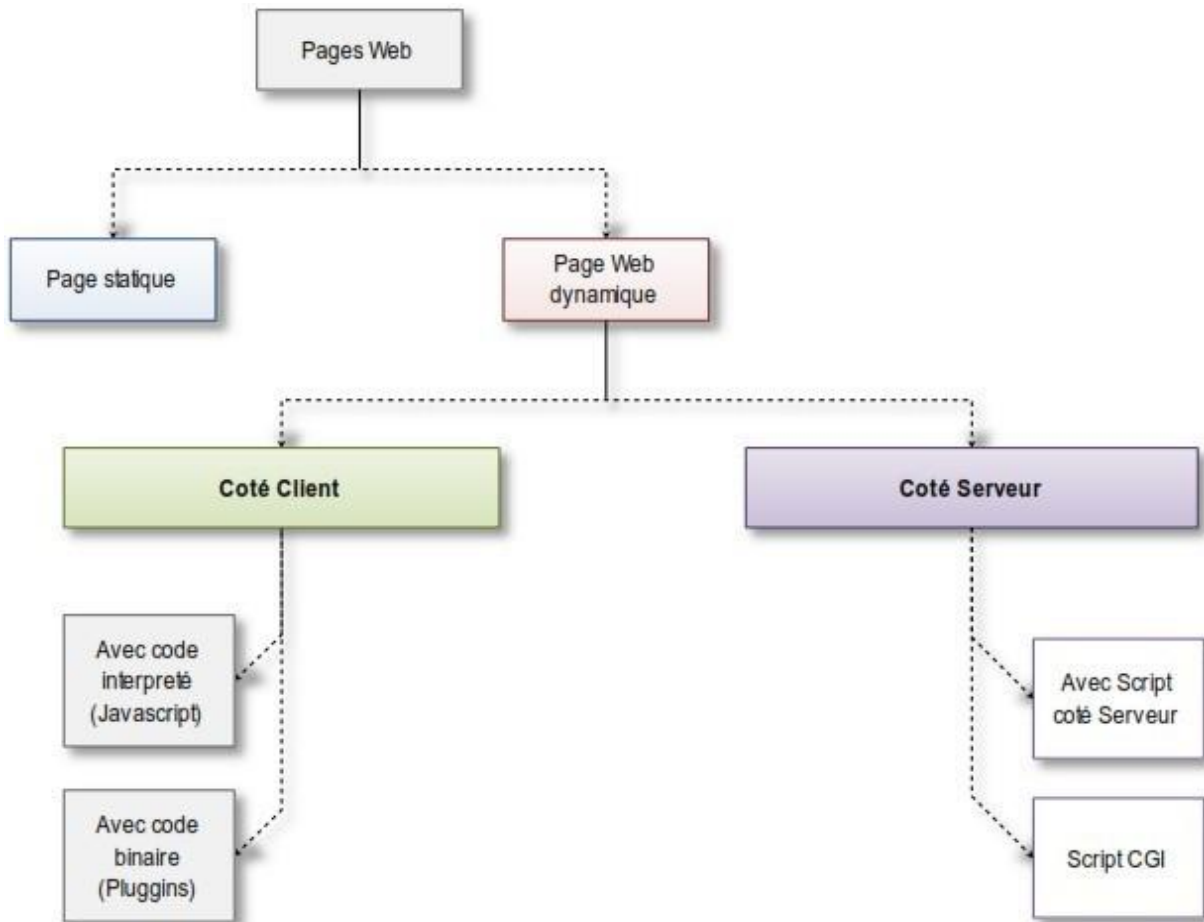
Dans ce cas de figure, le serveur Web et le programme exécuté sont différents.

Les scripts cotés serveurs, ont une approche contraire des CGI, plutôt que d'avoir un programme qui affiche du code HTML, nous avons une page HTML qui contient les instructions d'un programme dans un langage comme PHP, ASP, JSP, ... Il y a ensuite un programme s'exécutant comme le ferait un script CGI qui est chargé d'interpréter et d'exécuter les instructions qui sont contenu dans les pages Web. Cette page web contenant des instructions est appelé script coté serveur (car exécuté sur le serveur).

Le serveur web à chaque fois qu'il reçoit une requête vers une ressource qui est un script va exécuter le programme correspondant en lui passant comme argument le script ainsi que les données transmises par l'utilisateur:



1.5 - Catégories de pages web



1.6 - Développement Back-end

Le développement back-end peut englober la mise en place, la configuration et le maintien de plusieurs composantes essentielles d'un projet web :

- Le serveur, où sera hébergé le site internet qui doit tourner 24h/24. Il est souvent géré par un administrateur système.
- La base de données qui contient toutes les informations et données du site.
- L'application, c'est à dire le site internet dans sa globalité et la logique qui permet aux différentes pages, au serveur et à la base de données d'interagir.

Le "code **Back-End**", c'est la partie du code qui est exécutée par le serveur, il s'agit du travail qu'il réalise sur les pages Web des sites dynamiques avant de les envoyer au client.

Le travail du serveur sera codé dans un langage propre à lui : le "**langage serveur**".

C'est avec ce langage que le serveur pourra décider et générer des pages Web à renvoyer aux clients.

Il est important de comprendre que le résultat final renvoyé aux clients est toujours une page composée uniquement d'HTML, de CSS et de JavaScript.

2 - BREF PARCOURS DE QUELQUES OUTILS DE DÉVELOPPEMENT BACK-END

2.1 - Langages Back-end



Il existe de nombreux "langages serveurs" ou langages de programmation Back-end (ou langages Back-end tout court) :

- SQL, qui permet de faire des traitements sur les bases de données.
- PHP, Java, Python, Ruby, JS (NodeJs), C#, etc. qui sont les langages principaux pour le back-end.

Chacun des langages back-end a ses avantages et spécificités. En maîtriser un maximum est évidemment un plus mais c'est la connaissance approfondie de l'un d'entre eux qui est importante. Dans la majorité des cas, un développeur back-end a donc un langage de prédilection sur lequel il travaille exclusivement (ou presque).

2.2 - Frameworks de développement Back-end

Appelé aussi *infrastructure logicielle*, *infrastructure de développement*, *environnement de développement*, *socle d'applications*, *cadre d'applications*, etc.), un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme, un canevas.

Il est souvent fourni sous la forme d'une bibliothèque logicielle et accompagné du plan de l'architecture cible du Framework.

En complément des langages Back-end, il est essentiel de connaître un ou plusieurs Frameworks propres à chacun de ces langages pour diverses raisons : rendre le code plus clair, plus rapide à écrire, plus facilement modifiable et maintenable en équipe.

Il s'agit notamment de :

- Pour PHP : Symfony, Laravel, CakePHP, etc.
- Pour Java : Spring, Hibernate, Maven, Struts, etc.
- Pour Ruby : Ruby on Rails.
- Pour Python : Django, Pyramid, Flask, etc.
- Pour JavaScript : Node.js.
- Pour C# : ASP .NET (lire dotNet).

2.3 - CMS

CMS est l'acronyme de Content Management System, ce qui signifie "système de gestion de contenu". Un CMS est donc un outil qui sert à créer et mettre à jour un site internet sans avoir besoin de toucher au code source. Grâce à ces logiciels de gestion de contenu web, il est possible d'administrer un site en quelques clics. Ces outils permettent, via une utilisation simplifiée, de modifier une page web, des articles de blog ou d'autres données présentes sur votre support web (images, vidéos, etc.).

Voici une liste non exhaustive de CMS :

- Pour PHP : Joomla, WordPress, Drupal, SPIP, etc.
- Pour Java : Enonic XP, Magnolia, dotCMS, etc.
- Pour Python : Django CMS, Wagtail, Mezzanine, etc.

3 - LE DÉVELOPPEUR BACK-END

3.1 - Missions

Parmi les missions d'un développeur back-end, on peut citer :

- Analyse des besoins,
- Préconisation et conception de solutions techniques,
- Développement et évolution des fonctionnalités techniques du site web,
- Evolution dans un contexte agile,
- Validation avec son équipe des fonctionnalités développées.

- Documentation technique
- Former le client lorsque le site internet lui est livré,
- Contrôles qualités (montée en charge, tests unitaires) et veille à la performance du site web
- Maintenance d'applications :
 - Apporter un support technique tout au long de la vie du site internet,
 - Effectuer de la veille et des corrections suite à des problèmes remontés par le client.

3.2 - Compétences

Le développeur back-end doit disposer de compétences dans les domaines suivants :

- Outils de modélisation (analyse et conception) : MERISE, UML, etc.
- Langages de programmation « dynamiques » (au moins un langage) : PHP, NodeJS, Python, Java, Ruby, etc.
- API, web services.
- Frameworks de développement (au moins un Framework de prédilection).
- Frameworks de test automatisés (différents en fonction des langages de programmation utilisés).
- Administration de bases de données (SQL etc.).
- Sécurité, gestion des accès.
- Versioning de logiciel, par exemple avec Git, SVN, etc.
- Etc.