

LE PRINCIPE D'UN CMS AVEC JOOMLA!

VERSION 1.0 – OCTOBRE 2013

© 2013 LÉOST Informatique – Auteurs : [Philippe LÉOST](#), [Guénoé LÉOST](#) et [Quentin LÉOST](#)
Support disponible en téléchargement sur <http://glenan.fr/>

Vous êtes libres :



de reproduire, distribuer et communiquer cette création au public

Selon les conditions suivantes :



Paternité. Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).



Pas d'Utilisation Commerciale. Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.



Pas de Modification. Vous n'avez pas le droit de modifier, de transformer ou d'adapter cette création.

License : Creative Commons <http://creativecommons.org/licenses/by-nc-nd/2.0/fr/>

Ce support évolue et vous pouvez télécharger la dernière version du support de cours : <http://www.glenan.fr/>

Table des matières

Les CMS	2
Principe	2
Fonctionnement d'un CMS	2
Schéma d'architecture à 3 niveaux.....	3

LES CMS

PRINCIPE

Dans les modules de formation précédents de <http://glenan.net> ou <http://glenan.fr> vous avez appris à construire un site ex-nihilo, soit de façon classique dit « site vitrine » soit de façon dynamique avec un langage de programmation orienté web et une base de données, en incluant son langage de requête.

Pour une première approche de la création de sites, il est aussi possible d'utiliser des kits plus ou moins pré-paramétrés qui permettent de réaliser rapidement un site.

Il existe différentes familles de kits suivant l'usage présumé du site, même si certains peuvent appartenir à plusieurs familles. Les principales familles sont :

- Les blogs** : version moderne des journaux de bord, journaux intimes... où le principe de classement est le temps qui est le plus souvent géré de façon antéchronologique (les articles plus récents en premier). Les blogs autorisent souvent les internautes à commenter les billets d'information déposés.
- Les wikis** : créés initialement pour les besoins des développeurs ; ils sont utilisés actuellement par une communauté pour maintenir un référentiel. Chaque article est donc modifiable par tout internaute autorisé. La force du wiki réside dans l'archivage des modifications apportées, permettant de revenir à une version antérieure. Une version Wiki de « Glenan.net » est prévue.
- Les CMS** : Content Management Systems, traduit en SGC ou PGC (Systèmes ou Progiciel de Gestion de Contenu). Ils permettent une gestion du flux informationnel (workflow) et donc un travail d'équipe, tout en séparant le contenu du contenant. La plupart du temps, ils utilisent une base de données.

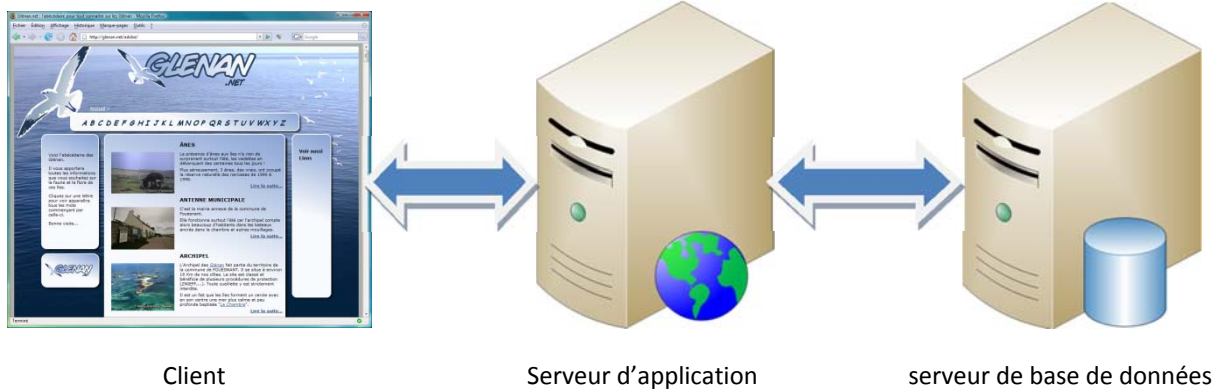
FONCTIONNEMENT D'UN CMS

Les CMS font partie d'un ensemble d'outils informatiques plus large appelé « Framework » qui sont des espaces de travail modulaires. Une approche d'Eclipse est indiquée dans le support : les outils du webmaster.

Il existe des centaines de CMS, dont les plus connus sont : Spip, Typo3, Plone, Plume, eZ Publish, Joomla, Drupal, etc.

D'autres outils, comme WordPress issu du monde Blog, intègrent des fonctionnalités CMS et rentrent dans cette catégorie.

Quasiment tous les CMS utilisent le même principe dit d'architecture 3-tiers, car il existe 3 niveaux :



L'internaute envoie une requête au serveur d'application (en cliquant sur un lien ou sur un bouton, par exemple). Le serveur d'application traite la requête et interroge la base de données. Les réponses de la base de données sont mises en forme et traitées par le serveur d'application qui retourne les informations au client pour qu'il puisse voir les résultats.

SCHEMA D'ARCHITECTURE A 3 NIVEAUX

L'architecture à 3 niveaux, appelée également architecture 3-tiers (de l'anglais three-tier) est une extension de l'architecture client/serveur (2-tiers) et elle est englobée dans les architectures multi-tiers ou n-tiers.

Le principe de fonctionnement s'effectue sur 3 niveaux :

- au niveau du client (appelé aussi couche de représentation),
- au niveau du serveur d'application (appelé parfois couche métier),
- au niveau du serveur de base de données.

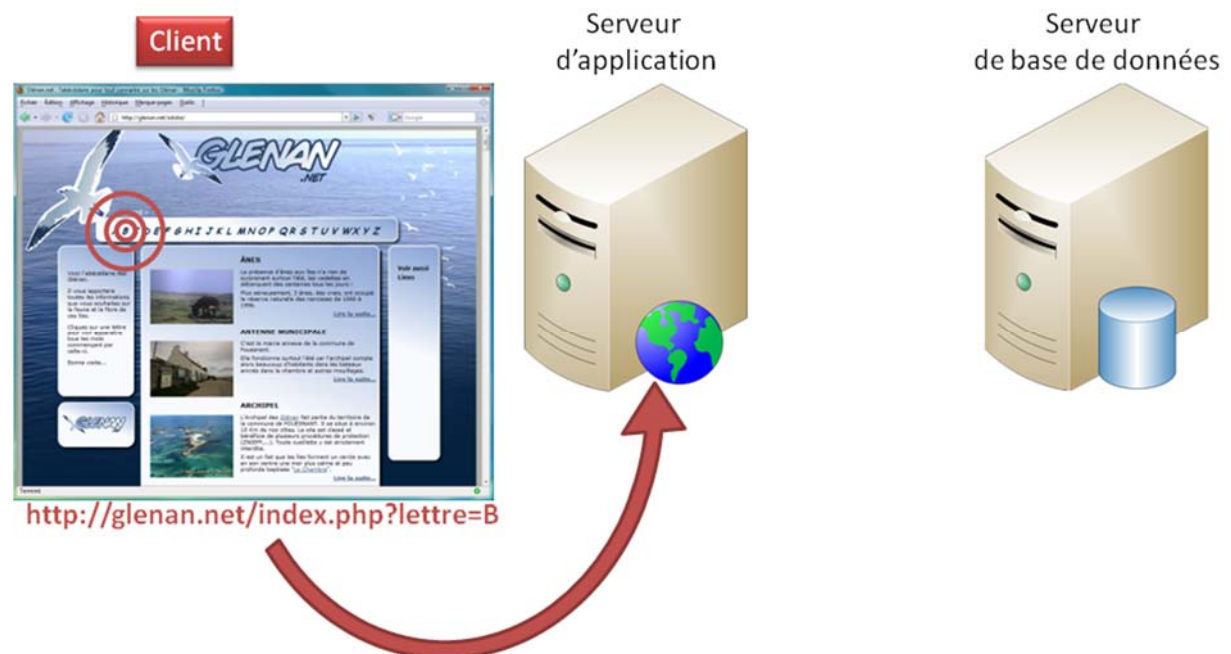
L'explication ci-dessous n'est évidemment qu'une approximation permettant de comprendre le fonctionnement d'une application qui utilise, dans le cas présent, un langage de type PHP et une base de données MySQL. Pour plus d'informations et appréhender la syntaxe PHP/MySQL, se référer au support de cours de Guérolé LÉOST : « Glenan.net : version PHP/MySQL », dont est extrait ce chapitre.

Le client affiche une page contenant du code HTML, avec éventuellement du JavaScript, des feuilles de styles, de l'AJAX... dans le but d'une représentation plus pratique pour l'internaute. Les liens proposés dans le site permettent d'envoyer un ordre sur le serveur d'application.

Par exemple, en déplaçant la souris sur la lettre B, en plus d'afficher un menu déroulant des différents articles de « Glenan.net » commençant par B, une URL apparaît dans la barre d'état en bas de page. Le lien pointe vers la même page index.php avec en plus un paramètre « lettre=B » qui est indiqué en tant que paramètre grâce au « ? » (point d'interrogation) dans l'URL.

Des paramètres multiples seront traités ultérieurement dans le support « Glenan.net : version PHP/MySQL ». C'est ce qui se passe également pour de nombreux CMS.

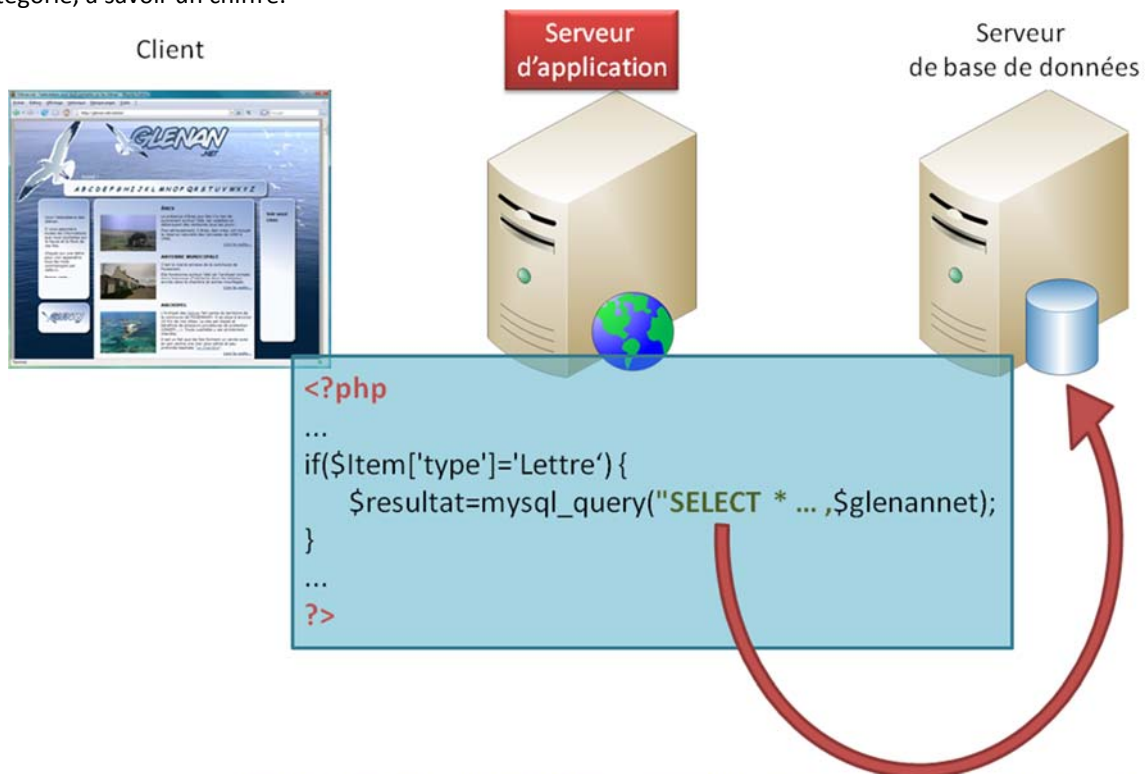
Attention néanmoins à l'utilisation de module de type SEO (Search Engine Optimization) et SEF (Search Engine Friendly) pour lesquels l'URL est réécrite.



En cliquant sur le lien, la requête est transmise au serveur d'application. La page index.php va traiter les paramètres qui lui sont transmis, à savoir « lettre=B ».

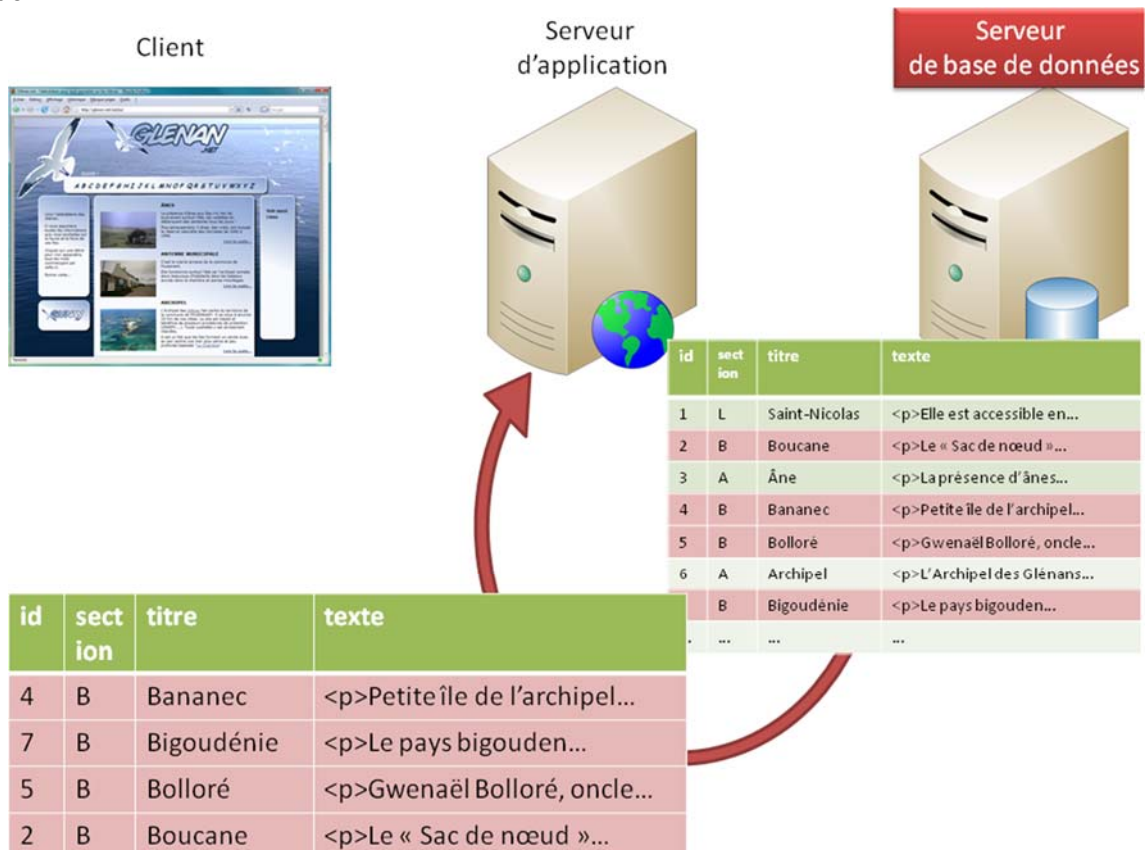
Un test conditionnel (voir schéma page suivante) va permettre de connaître quel type de paramètres a été transmis, (« lettre » dans notre cas) et de lancer une requête sur la base de données « \$glenannet ». Cet argument est optionnel, car dans notre exemple il n'y a qu'une base de données concernée par nos requêtes.

La requête permet de récupérer l'intégralité des champs (*) de la table « article » pour lesquels la catégorie correspond à la valeur « b ». Dans la réalité, la valeur correspondra à celle d'un identifiant de champ de la table catégorie, à savoir un chiffre.

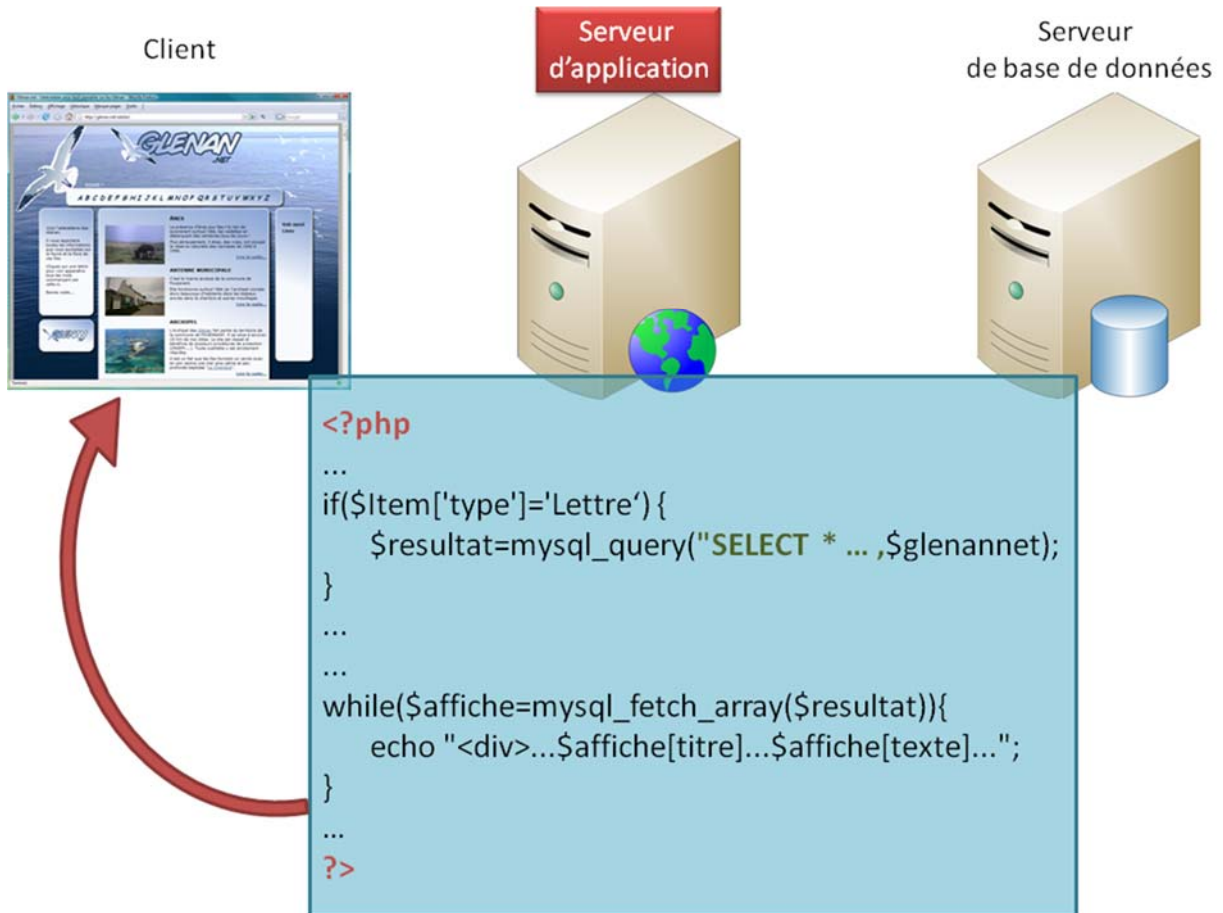


SELECT * FROM article WHERE section=' \$Item[value]' ORDER BY titre DESC

La requête SQL est transmise au serveur de base de données. Ce dernier sélectionne les enregistrements (les lignes du tableau) correspondant aux critères de la requête et les retourne triés par « titre » au serveur d'application.



Le serveur d'application récupère les informations dans une variable (\$resultat dans notre cas) et traite ensuite les enregistrements récoltés tant qu'il en existe (while).



Pour chaque enregistrement le serveur d'application génère du code HTML intégrant une balise <div> (pour plus d'informations voir le cours « Glenan.net : création du modèle ») et intègre les champs des enregistrements « titre », « texte »...

D'autres informations comme le « read more – lire la suite » peuvent être intégrées par le serveur d'application. Le code HTML généré est transmis au « client » qui l'interprète.

