

AIDE

De

Marc Bolard – We Design

# Dossier CLIENT

## config.php

Ce fichier contient les informations de configurations principales et spécifiques au site web :

$thisSite->LIST\_LANG: tableau des langues disponibles pour le site public

$thisSite->TYPE\_VIDEO\_DEFAUT: type de video par défaut (vimeo ou youtube). Cela facilite l’administration des vidéos.

$thisSite->cookiesAccept : Indique si l’on doit gérer l’acception des cookies comme l’oblige la réglementation Européenne.

$thisSite->googleAnalytics: Code Google Analytics ajouté à toutes les pages du site

$thisSite->mobile\_detect : Chargement de la bibliothèque de détection des mobiles (voir <http://mobiledetect.net> pour plus d’informations)

$thisSite->printCSS : Chargement de la feuille de style pour l'impression

D’autres sont différentes suivant le type de serveur $thisSite->SERVER :

$thisSite->DOMAINE : URL absolue du domaine (exemple : <http://www.monsite.com>). Ne pas mettre de slash à la fin. Défini dans config.php suivant le type de serveur

$thisSite->RACINE : URL absolue du site (par exemple : <http://www.monsite.com/demo>). Mettre un slash à la fin. Défini dans config.php suivant le type de serveur. Ne pas indiquer la langue.

$thisSite->SERVEUR\_BDD: nom du serveur de base de données

$thisSite->NOM\_BDD: nom de la base de données

$thisSite->LOGIN\_BDD: login de la base de données

$thisSite->MDP\_BDD: mot de passe de la base de données

$thisSite->MAIL\_HOST: Host SMTP

$thisSite->MAIL\_Username: login SMTP

$thisSite->MAIL\_Password: mot de passe SMTP

$thisSite->SMTPAut: indique si SMTP demande une authentification (si non, pas besoin d’indiquer le login et le mot de passe)

$thisSite->MAIL\_PORT: Port SMTP

$thisSite->MAIL\_SENDMODE: mode d’envoi : SMTP ou MAIL

$thisSite->MAIL\_SENDER= email expéditeur

## MODULES

Les modules commençant par « \_ » sont des modules « standards » dont on a besoin sur la plupart des sites

Le dossier d’un module peut contenir tout ce que l’on veut (PHP, JS, CSS, images, etc..)

Mais il faut savoir :

* Automatiquement est exécuté le script PHP portant le même nom que le dossier le contenant (« \_ » en moins) ou que le paramètre « Sous module »
* Automatiquement est chargé le script JS portant le même nom que le dossier ou que le paramètre « Sous module »
* Automatiquement est chargé la feuille de style portant le même nom que le dossier ou que le paramètre « Sous module »
* Automatiquement est chargé la feuille de style ayant le même nom que la skin en cours (en général defaut.css)

Appel d’un module

$obj\_module = new module("nom\_du\_module",”sous\_module”,”debug”);

La notion de sous module permet de gérer différents éléments dans un même module.  
Par exemple, le module « \_menus » gère tout les menus du site et c’est le sous module qui indique quel élément chargé et exécuté.

Propriétés facultatives :

$obj\_module->identifiant= « à renseigner si plusieurs instances du même module dans la page »

$obj\_module->params= permet de passer des paramètres aux scripts du module.  
Pour les utiliser dans les scripts du module utiliser la variable $params\_module

$obj\_module->js= »soit un fichier JS, soit un tableau de fichiers JS à charger dans l’entete de la page »

$obj\_module->css= »soit un fichier CSS, soit un tableau de fichiers CSS à charger dans l’entete de la page »

$obj\_module->head=”code html insérer dans la balise HEAD de la page”;

$obj\_module->doc\_ready=”javascript ajouté à $(document).ready en fin de page”;

$obj\_module->win\_load=”javascript ajouté à $(window).on("load"en fin de page”;

$obj\_module->javascript=”javascript ajouté en fin de script”;

$obj\_module->footer=”code html insérer tout à la fin de la page”;

$obj\_module->return= si True, retourne le code HTML généré par le module.

Chargement du module

$obj\_module->load();

// Utilisation

Dans le fichier TPL, insérer le code suivant :

{$MODULE\_xxxyyyy}

Xxxx étant le nom du module ou du sous module (si exsite)

Yyyy étant l’identifiant indiqué en paramètre

## OUTILS

Les outils commençant par « \_ » sont des outils « standards » dont on a besoin sur la plupart des sites

Le dossier d’un outil peut contenir tout ce que l’on veut (PHP, JS, CSS, images, etc..)

Mais il faut savoir :

* Automatiquement est exécuté le script PHP portant le même nom que le dossier le contenant (« \_ » en moins) ou que le paramètre « Sous outil »
* Automatiquement est chargé le script JS portant le même nom que le dossier ou que le paramètre « Sous outil »
* Automatiquement est chargé la feuille de style portant le même nom que le dossier ou que le paramètre « Sous outil »
* Automatiquement est chargé la feuille de style ayant le même nom que la skin en cours (en général defaut.css)

Appel d’un module

$obj\_outil = new module("nom\_du\_module",”sous\_module”,”debug”);

La notion de sous module permet de gérer différents éléments dans un même module.  
Par exemple, le module « \_menus » gère tout les menus du site et c’est le sous module qui indique quel élément chargé et exécuté.

Propriétés facultatives :

$obj\_ outil ->js= »soit un fichier JS, soit un tableau de fichiers JS à charger dans l’entete de la page »

$obj\_ outil ->css= »soit un fichier CSS, soit un tableau de fichiers CSS à charger dans l’entete de la page »

$obj\_ outil ->header=”code html insérer dans la balise HEAD de la page”;

$obj\_ outil ->doc\_ready=”javascript ajouté à $(document).ready en fin de page”;

$obj\_ outil ->win\_load=”javascript ajouté à $(window).on("load"en fin de page”;

$obj\_ outil ->javascript=”javascript ajouté en fin de script”;

$obj\_ outil ->footer=”code html insérer tout à la fin de la page”;

Chargement du module

$obj\_ outil ->load();

## \_PAGES\_STD

Contient des pages (PHP et TPL) que l’on retrouve fréquement dans un site et qui évolue peu:

404 : Page 404

Download : Utiliser lorsque l’on souhaite proposer une fichier à télécharger

Index\_langue : Utiliser pour créer une page d’index de choix de la langue

Pub : Utiliser pour gérer l’affichage des publicités

Redirection\_pub : Utiliser pour la redirection et les statistiques en cas de click sur une Pub.

Site\_en\_construction : Page si le site est indiqué en construction

Site\_en\_demonstration : Page si le site est indiqué en démonstration

Sitemap : Script utilisé pour générer sitemap.xml

# Dossier COMMUN

## CSS

Contient un reset CSS. Pour plus d’infos: <http://knacss.com/>

## FCT

Contient des fonctions PHP de base :

Fichier fonctions\_chaine.php

**startsWith** : indique si une chaine commence par une autre chaine

**full\_sanitize\_string** : nettoie une chaine des caractères (accent, majuscule, spéciaux, etc..)

**sanitize\_string\_for\_url** :nettoie une chaine pour en faire une URL valide

**delQuotes** :supprime uniquement les simple et double quotes dans une chaine

**accent\_majuscule** : transforme une chaine en minuscule accentué en majuscule accentué

**suppression\_accent\_majuscule** : transforme une chaine en minuscule accentué en majuscule NON accentué

**suppression\_accent** : supprime tous les accents

**translate\_chaine** : converti une chaine ISO en UTF 8 et inversement

**br2nl** : remplace <br> par nl (inverse de la fonction PHP nl2br)

**remplace\_text** : Remplace une chaine de caractere dans un texte

**tronc\_texte** : tronque un texte à nbc caractères

**complete\_lien** : complète les liens si lien email

**format\_num** : mise en forme d'un nombre

**crypt\_string et decrypt\_string**  : encryptage et décryptage d’une chaine

**add\_variables** : Remplacement des variables dans un texte

**add\_glossaire** : Remplacement des termes du glossaire dans un texte

Fichier fonctions\_data.php

**getTexte**: Retourne un texte Divers à partir de son ID

**getImage**: Retourne une image Divers à partir de son ID

**getFile**: Retourne un fichier Divers à partir de son ID

**getEmail**: Retourne les données de la table Emails à partir de son code

**sendEmail**: Envoi un email en utilisant PHPMailer

**add\_plan\_du\_site**: Ajoute un élément au fichier sitemap.xml

Fichier fonctions\_date.php

**test\_date**: test si une date au format JJMMAA est valide

**add\_jours**: ajoute des jours à une date

**ecart\_dates**: Retourne l’écart en jours entre 2 dates

**get\_jour**: Retourne le jour d’une date au format AAMMJJ ou AAAAMMJJ

**get\_mois**: Retourne le mois d’une date au format AAMMJJ ou AAAAMMJJ

**get\_annee**: Retourne l’année d’une date au format AAMMJJ ou AAAAMMJJ

**format\_date**: Formate une date AAMMJJ en JJ/MM/AA ou AAAAMMJJ et JJ/MM/AAAA

**deformat\_date**: Fait l’inverse de la fonction format\_date

**format\_datejour**: Formate une date AAMMJJ en JJ MOIS AAAA

**format\_datemois**: Formate une date AAMMJJ en MOIS AAAA

**format\_heure**: Formate une heure HHMMS en HH :MM :SS

**numero\_semaine**: Retourne le n° de semaine d’une date

Fichier fonctions\_initialisation.php

**addStructure**: Ajoute du code (HTML ou JS) dans le template de la page en cours

Type possible : PAGE\_doc\_ready, PAGE\_win\_load, PAGE\_javascript, PAGE\_footer, PAGE\_head, PAGE\_css\_client, PAGE\_js\_client

**prepareMenuLinkClassLI**: Retourne la ou les classes à ajouter aux balises <LI> d’un menu (tenant compte du type de page)

**prepareMenuLinkClassA**: Retourne la ou les classes à ajouter aux balises <A> d’un menu (tenant compte du type de page)

**prepareMenuLink**: Retourne le lien à insérer dans une balise <A> d’un menu (tenant compte du type de page)

**get\_path\_pages**: retourne le chemin d’un script PHP d’une page (soit dans le dossier \_pages soit dans \_pages\_std du dossier \_client)

**get\_path\_modules**: Idem get\_path\_pages mais pour les modules

**get\_path\_outils**: Idem get\_path\_pages mais pour les modules

**load\_intitules\_by\_lang, load\_data\_lang\_plus, pre\_load\_data\_lang\_plus**: Chargement des intitulés par langue.

Fichier fonctions\_systeme.php

**echoa**: Remplace la fonction ECHO avantageusement (gère les array et objet)

**curl\_get**: Nécessaire à VIMEO. cURL doit etre activité sur le serveur PHP

**config\_404**: Prépare le chemin et la page 404 à afficher

**generer\_token**: Génère, sauvegarde et retourne un token (pour sécuriser un formulaire)

**verifier\_token**: Vérifie un token

**is\_IE, is\_Chrome, is\_Safari**: Vérifie si le navigateur est Internet Explorer, Chrome ou Safari

**is\_IE6, is\_IE7, is\_IE8**: Vérifie la version d’Internet Explorer

**get\_os**: Retourne l’OS (Linux, Mac ou Windows)

**ctr\_global**: Conversion et contrôle données passer en \_GET ou \_POST

**show\_var\_session**: Affichage de toutes les variables de Session en cours

**get\_extension**: Retourne l’extension d’un fichier

**remove\_extension**: Retourne le nom d’un fichier sans son extension

**get\_nom\_fichier**: Retourne le nom d’un fichier sans le chemin

**format\_poids\_fichier**: Formate un poids de fichier en Ko, Mo, Go, …

**resize\_me**: Retourne des dimensions en respectant des contraintes

**read\_tree**: Retourne l'arborescence d'un dossier

**del\_tree**: Supprime une arborescence de dossier (sous dossiers et fichiers)

**get\_vignette**: Retourne le nom du fichier vignette à partir d'un fichier image

**get\_image**: Retourne le nom du fichier image

**get\_permissions**: Retourne les droits d'un fichier ou d'un dossier

**prop\_fichier**: Retourne les propriétés d’un fichier ou un dossier

**var\_to\_js**: Conversion d'un tableau PHP en JS

Fichier fonctions\_verif.php

**check\_mail**: vérifie si un email est valide

**check\_URL**: vérifie si une URL est valide

**is\_unsigned\_float**: vérifie si c'est bien un float non signé

**is\_unsigned\_integer**: vérifie si bien un entier non signé

**is\_signed\_float**: vérifie si c'est bien un float signé

**is\_signed\_integer**: vérifie si bien un entier signé

**is\_alpha**: vérifie si c'est bien que des lettres sans acccent

**is\_alphanum**: vérisie si que des chiffres ou des lettres sans accent

**is\_valide\_code**: vérifie si que des chiffres, des lettres et "\_" et "-"

## INC

Contient des classes d’objets :

Class.articles.php

Permet de manipuler les données de la table ARTICLE.

**Appel**

$obj\_article = new article("code de l’article");

**Propriétés facultatives**

$obj\_article->id=identifiant d’un article;

$obj\_article->where=" titre!=’’ "

La clause WHERE contient automatiquement : ACTIF=1 AND art=’code de l’article’

$obj\_article->lg="en";

Si cette propriété n’est pas défini on utilise la langue courrante ($thisSite->current\_lang)

Si elle est égale à « all » on ne l’ajoute pas à la clause WHERE

$obj\_article->orderby="ID ASC"; (valeur par défaut)

$obj\_article->limit="1,5";

$obj\_article->medias="image,video,file,link";

Ajoute le ou les médias indiqué

**Exécution**

$result=$obj\_article->query();

Accepte « 1 » en paramètre pour afficher la requête.

$result est un tableau (array) avec un n° incrémental comme clé.

**Exemple utilisation du résultat**

$row = current($result);

Ou

foreach($result as $row){

echo($row["titre"]);

}

Class.bdd.php

Permet de générer des requetes SQL sécurisés

**Pour un SELECT**

$mySelect = new mySelect(\_\_FILE\_\_);

$mySelect->tables="matable” ou array(“matable1”,”matable2”)

$mySelect->fields="liste champs ou \*";

$mySelect->where="champ1=:bind1 AND champ2>=:bind2 AND champ3=:bind3 AND champ4=’test’";

$mySelect->whereValue["bind1 "]="chaine";

$mySelect->whereValue["bind2 "]=array(140819,PDO::PARAM\_INT);

$mySelect->whereValue["bind3 "]=array("chaine ",PDO::PARAM\_STR);

$mySelect->orderby="champ DESC";

$mySelect->groupby="champ";

$mySelect->limit="1,5";

$result=$mySelect->query();

**Pour un INSERT**

$myInsert = new myInsert(\_\_FILE\_\_);

$myInsert->table=" matable";

$myInsert->field["champ1"]="chaine";

$myInsert->field["champ2"]=integer;

$result=$myInsert->execute();

**Pour un UPDATE**

$myUpdate = new myUpdate(\_\_FILE\_\_);

$myUpdate->table=" matable";

$myUpdate ->field["champ1"]="chaine";

$myUpdate ->field["champ2"]=integer;

$myUpdate->field["champ3"]="LITERAL: champ3+1";

$myUpdate ->where="champ1=:bind1 AND champ2>=:bind2 AND champ3=:bind3 AND champ4=’test’";

$myUpdate ->whereValue["bind1 "]="chaine";

$myUpdate ->whereValue["bind2 "]=array(140819,PDO::PARAM\_INT);

$myUpdate ->whereValue["bind3 "]=array("chaine ",PDO::PARAM\_STR);

$result=$myUpdate->execute();

**Pour un DELETE**

$myDelete = new myDelete(\_\_FILE\_\_);

$myDelete->table=" matable";

$myDelete ->where="champ1=:bind1 AND champ2>=:bind2 AND champ3=:bind3 AND champ4=’test’";

$myDelete ->whereValue["bind1 "]="chaine";

$myDelete ->whereValue["bind2 "]=array(140819,PDO::PARAM\_INT);

$myDelete ->whereValue["bind3 "]=array("chaine ",PDO::PARAM\_STR);

$result=$myDelete->execute();

Class.medias.php

Permet de récupérer les données d’un medias (image, file, video, link)

**Appel**

$thisMedia = new media(“code du media”);

**Propriétés facultatives**

$thisMedia->idParent=identifiant du parent ;

$thisMedia->idMedia=identifiant du media ;

$thisMedia ->where=" titre!=’’ "

La clause WHERE contient automatiquement : ACTIF=1 AND field\_media=’code du media’

$thisMedia ->orderby="chrono ASC"; (valeur par défaut)

$thisMedia ->onlySelection=1;

Ne retourne que les medias cochés

$thisMedia ->sizeThumbVideo=0 ou 1 ou 2; (0 par défaut)

Utiliser uniquement pour les videos et permet de choisir la taille de l’image illustrant la vidéo (renvoyer par YouTube ou Vimeo)

**Exécution**

$dataMedias=$thisMedia->get();

Le résultat est différent par type de média :

Pour une image :

* Id
* Image
* vigX (X étant le n° de la vignette de 0 à 9)
* legende
* fichier\_destination
* lien\_destination
* cible\_destination

Pour un fichier :

* Id
* fichier
* legende

Pour une vidéo:

* Id
* Lien\_video
* Type\_video
* Legende
* Thumb (lien vers l’image YouTube ou Vimeo)
* Player (lien vers une URL YouTube ou Vimeo intégrable

Pour un lien :

* Id
* lien
* cible
* legende

Class.modules.php et Class.outils.php

Voir les explications dans le chapitre « Dossier CLIENT »

Class.site.php

Permet de définir toutes les variables générales utilisés dans le site.

En mode « local » ($thisSite->SERVER=’local’), cet objet est rechargé à chaque changement de page.   
Par contre en mode production, il est stocké dans une variable de Session et il faut donc fermer le navigateur pour forcer le rechargement.

Ces variables sont modifiables par n’importe quel script.

Voir le fichier pour les découvrir toutes.

Il charge également les informations sur le site, les pages et les menus :

**Informations sur le site**

$thisSite->siteTitle : Titre du site

$thisSite->suffixeTitle : Suffixe ajouté à tout les MetaTag Title des pages

$thisSite->statut: C: En construction, D: En démonstration

$thisSite->skin: chemin vers le dossier de la skin en cours (en général : client/skins/defaut/)

$thisSite->skinImages : chemin vers les images standards utilisées par la skin

$thisSite->skinImagesClient: chemin vers les images spécifiques utilisées par la skin

$thisSite->socialImage : Image utilisé pour le partage sur les réseaux sociaux

$thisSite->LOADER\_PAGE : indique si l’on affiche un loader à chaque chargement de page

$thisSite->SERVER : indique le type de serveur (local ou prod)

$thisSite->racineWithLang : URL absolu du site avec langue (pour les sites multilangues, sinon identique à $thisSite->RACINE)

$thisSite->menus : tableau de menus avec l’arborescence en valeur (idPage)

$thisSite->pages : tableau de toutes les pages de la page table PAGES.

La clé est l’IdPage et la valeur un tableau de toutes les données de la table PAGES, plus :

$thisSite ->pages[$idPage]["page\_url\_menu"][$menu] : tableau par Menu des URL a utiliser dans les menus (avec arbo et type de page)

$thisSite ->pages[$idPage]["page\_url\_arbo"][$menu] : tableau par Menu des URL dans l'arborescence (sans le type de page)

$thisSite ->pages[$idPage]["classLI"] : classes à ajouter aux balises <LI> d’un menu

$thisSite ->pages[$idPage]["classA"] : classes à ajouter aux balises <A> d’un menu

$thisSite->tags: tableau des tags de la table TAGS.   
Ils se présente ainsi : array(‘code parent’=>array(idTag=ValeurTag))

$thisSite->chronoPages = tableau de toutes les pages visités depuis le début de la session

**Informations sur la page**

$thisSite->current\_lang : Langue de la page en cours

$thisSite->idPage : Identifiant de la page en cours dans la table PAGES

$thisSite->page : données de la page en cours

(c’est un raccourci de $thisSite ->pages[$thisSite->idPage ])

$thisSite->current\_rub: Identifiant de la page au niveau Rubrique

$thisSite->current\_srub: Identifiant de la page au niveau Sous Rubrique

$thisSite->current\_ssrub : Identifiant de la page au niveau Sous Sous Rubrique

$thisSite->metaTags["title"]: MetaTag title de la page en cours

$thisSite->metaTags["keywords"]: MetaTag keywords de la page en cours

$thisSite->metaTags["description"]: MetaTag description de la page en cours

$thisSite->metaTags["robots"]: MetaTag robots de la page en cours

$thisSite->socialTags["titre"]: Titre de la page en cours pour les réseaux sociaux

$thisSite->socialTags["image"]: Image de la page en cours pour les réseaux sociaux

$thisSite->socialTags["texte"]: Description de la page en cours pour les réseaux sociaux

$thisSite->socialTags["lien"]: URL de la page en cours pour les réseaux sociaux

Commun.js

Contient des fonctions Javascript de base

**getQueryString**: Retourne les paramètres passés dans l’URL (après le ? de l’URL)

**inView**: Indique si un élément HTML est dans la partie visible de la page.

Exemple d’utilisation :

$(window).scroll(function(){

if($("#googlsseMap").inView()) {

loadGoogleMap();

}

});

**loadScript**: Ajout du code JS à la volée

**goTargetLink**: Appel un lien à la place de la page en cours ou dans une nouvelle fenêtre

**ucwords**: Met en majuscule la première lettre d'une chaine

**checkemail**: Vérification email

**checkchiffres**: Vérification que des chiffres

**randomBetween**: retourne une valeur aléatoire comprise entre deux valeurs données

# Dossier GESTION

## \_client

Contient tous les scripts spécifiques aux clients.

A la racine, il y a 4 scripts qui permettent de personnaliser les éditeurs Wysiwyg :

ckstyles.js : styles disponibles dans CkEditor

cktemplates.js : templates disponibles dans CkEditor

tmstyles.tpl : styles disponibles dans TinyMce

tmtemplates.tpl : templates disponibles dans TinyMce

## inc

Contient des objets, bibliothèques de fonctions et script à inclure.

controle\_login.php : Contrôle qu’on est bien identifié pour accéder à l’administration.  
Contient également l’identification automatique sur « Marco » lorsqu’on est en local.

## Pages

Contient les scripts des rubriques standards de l’administration

Ils y a des sous dossiers par catégorie de scripts.

Dans le sous dossier « exemples », il y a des exemples complets de rubriques  avec toutes les propriétés et fonctionnalités.